VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum-590018

A PROJECT REPORT **(15ECP85)** ON

"Smart Billing System based on Bar Code Recognition and Image Classification using Webcam"

Submitted in Partial fulfillment of the Requirements for the VIII Semester of the Degree of

Bachelor of Engineering in Electronics & Communications Engineering

By

Nikhil Kumar (1CR16EC099)

Kumar Abhijeet (1CR16EC065)

Nitendra Singh Chauhan (1CR16EC205)

Under the Guidance of,

Mr. Harsha B.K

Assistant Professor, Dept. of ECE

DEPARTMENT OF ELECTRONICS & COMMUNICATIONS ENGINEERING

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

2019-2020

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI,BANGALORE-560037

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled **"Smart Billing System based on Bar Code Recognition and Image Classification using Webcam"** carried out by Mr. **Nikhil Kumar**, USN **1CR16EC099** ,Mr. **Kumar Abhijeet** , USN **1CR16EC065**, Mr. **Nitendra Singh Chauhan** USN **1CR16EC205** , bonafide students of CMR Institute of Technology, in partial fulfillment for the award of **Bachelor of Engineering** in Electronics & Communications Engineering of the Visvesvaraya Technological University, Belgaum during the year 2019-2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Signature of Guide	**Signature of HOD**
Mr. Harsha B.K	**Dr. Elumalai R**
Assistant Professor	**Professor & HOD**
Dept. of ECE, CMRIT	**Dept. of ECE, CMRIT**

DECLARATION

We, the students of 8th semester of Electronics & Communication Engінeerring, CMR Institute of Technology, Bangalore declare that the work entitled " **Smart Billing System based on Bar Code Recognition and Image Classification using Web-cam**" Project Work has been successfully completed under the guidance of Mr. Harsha B.K, Electronics & Communications Engineering Department, CMR Institute of technology, Bangalore. This dissertation work is submitted in partial fulfillment of the requirements for the award of Degree of Bachelor of Engineering in Electronics & Communications Engineering during the academic year 2019 - 2020. Further the matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

Place:Bengaluru

Date:20/05/2020

Team members:

NITENDRA SINGH CHAUHAN (1CR16EC205) ____________________

NIKHIL KUMAR (1CR16EC099) ____________________

KUMAR ABHIJEET (1CR16EC065 ____________________

ABSTRACT

This report presents an algorithm to develop a supermarket billing system, which recognizes both the barcode and object using image processing and deep learning tools. MATLAB software is used to test the algorithm.

There are various types of barcodes in use today. These include code 128, code 39, EAN 13, etc. Our algorithm decodes EAN 13 barcode that is used universally.

In first algorithm, the database of products is created in an excel sheet. Images of products are taken using USB webcam, which is to be installed manually in MATLAB image acquisition toolbox. Preprocessing is done that includes processes such as RBG space conversion, feature extraction; barcode detection and row scan line on these images to extract the barcode region and the barcode number is decoded using the decoding algorithm.

In second algorithm, the pre-trained CCN model GoogleNet is retrained for the specific sets of the images, in which barcode based billing is not possible. Once trained, real time image through webcam can be classified in the same window.

The barcode number and classified image label is assigned with random number is further compared with the database, and the corresponding information of the scanned product is displayed in a different excel sheet. A final bill of all scanned products is created after scanning of all products.

Proposed system provides a convenient method of extracting information from the barcode and product image at a lesser cost as compared to typical electronic barcode scanners. This system can be used by businesses requiring accurate results at low cost.

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing me a platform to pursue my studies and carry out my final year project

I have a great pleasure in expressing my deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore, for his constant encouragement.

I would like to thank **Dr. Elumalai R,** HOD, Department of Electronics & Communications Engineering, CMRIT, Bangalore, who has been a constant support and encouragement throughout the course of this project.

I consider it a privilege and honor to express my sincere gratitude to my guide **(Mr. Harsha B.K), Assistant Professor,** Department of Electronics & Communications Engineering, for the valuable guidance throughout the tenure of this review.

I also extend my thanks to all the faculty of Computer Science and Engineering who directly or indirectly encouraged me.

Finally, I would like to thank my parents and friends for all their moral support they have given me during the completion of this work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Individuals have constantly created innovation to bolster their requirements as from the start of humankind. The fundamental reason for the development of innovation is ought for more independency and this leads to improving tasks and making regular ones simpler and speedier. One significant task that individuals invest in the maximum measure of energy is in shopping. The shopping center is a spot where individuals get their everyday necessities running from sustenance items, garments, electrical machines, and so forth. In this innovative world, each grocery store and supermarkets utilize barcode-based billing with a specific end goal to help clients choose and store the items which they expect to buy. Customers usually purchase the products required and place them in their carts and thereafter wait at the counters for payments of bills. The items in the cart include first that have the barcode and second without barcode-like fruits and vegetables.

In recent years, extensive research has been carried out on vision-based automatic identification technology that recognizes image codes using a system to provide various services that can recognize the authenticity of any product. Using Multiplexing and De-multiplexing process encode and decode the information from a single barcode with special symbols and split the data back to their barcode pattern where this barcode pattern can be read by the system. Further, these can be includes in billing. In case if the barcode is not available we have recent advancements in artificial intelligence and machine learning have hugely contributed to the growth of Image Recognition and Object Detection in retail.

Barcodes are widely used in many grocery supermarkets like Big Bazar, Easy Day for billing, and statement generation. Check-out counters use laser bar-code readers in such stores but the space between the sensor and the object should be nearly zero when the reader is applied. It is essential to have an alternative method for billing methods that can be affordable to all kinds of shops as well as supermarkets. And include both types of items that have barcode and second which don’t have a barcode.

1.1 Problem Statement

In the supermarket, many hand-held scanners automatically scan the barcode on the merchandise. In small shops, the billing process remains simply without barcode scanners or any other complicated tool. Installing a computer system then attaching a barcode reader followed by accessing information at software is costly and requires additional skills in the maintenance of such a system. Also, barcodes are included only in those items that are registered with the government, as government issues barcode numbers. Then what about other categories of items which include handmade or packed local items, organic food items which include fruits and vegetables in such items barcode-based scanning is not possible. Barcode reader is a dedicated product as it can only be used for barcode scanning purposes. Thus, our final objective is developing a system that tries to solve these issues and remains not so expensive and ultimately easy to use that is GUI should user-centric.

1.2 Existing System

Nowadays most of the barcode scanners are using infrared methods to scan a barcode. A barcode reader (or barcode scanner) is an optical scanner that can read printed barcodes, decode the data contained in the barcode and send the data to a computer. Like a flatbed scanner, it consists of a light source, a lens and a light sensor translating for optical impulses into electrical signals. Additionally, nearly all barcode readers contain decoder circuitry that can analyze the barcode's image data provided by the sensor and sending the barcode's content to the scanner's output port. Also image classification based billing is available only in E-commerce platform.

1.3 Previous Work

Barcode reading has been studied for decades. And after continuous improvement, it is now represented by a well established industrial standard. Until recently, barcode reading was performed almost exclusively with dedicated hardware. Even though there is growth in the use of camera-based readers, most of the challenges posed by this new method are yet to be solved. Commercial scanners that are used in supermarkets shine a stripe of pulsed light on the barcode and record the intensity of its reflection. As it makes use of active illuminations, it makes barcodes virtually insensitive to changes of ambient illumination. Additionally, their design often requires the barcodes to be relatively close to the scanner to scan it successfully. In general, these dedicated devices produce a high-quality signal that allows for robust barcode reading. On the other hand, reading barcodes with cameras present new challenges. The major issue faced is with the poor quality of the image due to noise or low contrast. Orazio Gallo and Roberto Manduchi[1] proposed a new approach to barcode decoding that bypasses binarization. Their technique relies on deformable templates and exploits all of the gray-level information of each pixel. They have used UPC-A symbol of barcode and presented a new algorithm for barcode decoding (localization and reading) that can deal with blurred, noisy and low-resolution images.

Luping Fang and Chao Xie [2] present an algorithm, which can detect barcode region even in a complex background with the help of region-based image analysis. Being different from traditional region-based image analysis, the algorithm is processed with an integral image. It can deal with images that are blurred, contain obliquely positioned barcode regions and are shot under different illumination. Xianyong Fang[3] has proposed a fast and robust recognition method for noisy Code 39 barcode. The proposed method follows two steps: search and decoding. In the first step, all asterisks in the image are found with evenly defined scan lines and then those lines with the same directions are matched together to get a valid barcode region. In the second step, a local de-noise method is applied for eliminating noise in the barcode region. Further, a middle band filter is used for decoding the barcode. Their method also has certain shortcomings. For images whose resolution are quite different from 2400x3500, this approach cannot successfully recognize the barcode automatically as many threshold values need to be set again, e.g. the local de-noise threshold.

W. N. W. Shuhaimi projected in Real Time Barcode Reader for Laboratory Attendance (software) that the project was developed to create and develop database system for laboratory attendance based on real time barcode reader. System was able to read and display barcode and then save the data into a database system for recording purpose. Microsoft Visual Basic 6 is used as the main software to create Microsoft Access [4].

M. E. H A. Azizi developed a mobile barcode scanner to be used for laboratory attendance which it can save data in PIC16F877A Microcontroller and monitor using LCD module interface. Project consists of the combination of hardware such as barcode scanner, PIC16F877A microcontroller and LCD module to produce system that have input output signal. The barcode scanner is device that transmits data to microcontroller and the result show up instantly to the LCD by using serial communication. Project used Isis 6 Professional and the MPLab IDE as a compiler for the microcontroller [5].

1.4 Present Development

Apart from billing purpose barcode and image classification have multiple use we have listed these includes

i. **Barcode Enabled Event Management System for Logistics and Consumables Management**
Here Event management has a complex set of processes, which involve registration, secured venue access, real time stock taking of consumables and crowd-management.

ii. **Barcode and Image Recognition based Student Attendance System**
Here student attendance play significant role in order to justify academic outcome of a student and college as overall. There is a tool to systematically keep the students attendance record due to increasing number of college students .This help the teachers in college to avoid maintaining the registry book. The tool can either use barcode based tracking technique or image recognition techniques.

iii. **Camera-phone based barcode scanning**
Mobile phones are always with us. Advertisers already use the mobile phone as a tool to interact with consumers. However, current options are either constrained or difficult to implement.

Contrast, in 2D barcodes allow owners of camera-enabled wireless phones to conveniently interact with print and electronic media. For example, consumers could use 2D barcodes to gain instant access to specific information, such as product and service prices, recipes, or bus arrival times. For Publishers, code scanning also provides rich targeted information and the ability to drive and collect responses, which will help them gain insight into scanning users. Overall, 2D barcodes will enable a higher level of mobile interaction and communication. 2D barcode also allow us to pay online through various like UPI Pay, Paytm Pay and others.

1.5 Brief Outline of the Project

The works carried out at each project phase are outlined below:-

1.5.1 Learning & Analysis Phase

This phase includes:

i. Gathering knowledge about existing techniques.
ii. Well understanding of the project design review.
iii. Learning tools, technologies & programming Language for
iv. Coding purpose.

1.5.2 Design & Implementation

This phase Includes:

i. Designing the overall functional view
ii. Describing the language, platform used in the project implementation.
iii. Identification and design of the modules for implementing.
iv. Implementing the applications for accessing and controlling the different types of services.

1.5.3 Testing Phase

This phase includes:

i. Writing the test cases for testing the implemented modules.
ii. Executing the test cases manually, comparing and evaluating the actual result with the expected result.

1.6 Organization of Report

This report provides an overview of the project that is “Smart Billing System based on Bar Code Recognition and object classification using Webcam”. This project has seven chapters starting from introduction to conclusion and future work.

Thus, the following pages summarize background research, implementation, working and results relating to the project. **Chapter 2** includes information about barcode and different types of barcode. Followed by An overview on EAN-13 Encoding and decoding. It also includes different methods of detecting the barcode and other pre-trained models. **Chapter 3** includes features involves in project followed by system flow chart and algorithm behind. It also includes Conceptual Analysis for Barcode Localization and Image Pre-Processing and transfer training for a pre-trained model. Here, we also discuss about system methodology development and reasons for choosing this project. **Chapter 4** includes various software requirements including various tools that will provide various functions support to project. It also includes hardware requirement. **Chapter 5** includes implementation of different algorithms for the detection and recognition of barcode and image classification by transforming them into blocks of programming codes. Further, it also includes steps involve in the setup of webcam and database. **Chapter 6** includes steps for the execution of the project work and it also includes various discussions over accuracy of the project working. **Chapter 7** is conclusion and also discuss about future work of the project.

CHAPTER 2

LITERATURE SURVEY

Object recognition is a general term to describe a collection of related computer vision tasks that involve identifying objects in digital photographs or real time video. Image classification involves predicting the class of one object in an image. Object localization refers to identifying the location of one or more objects in an image and drawing abounding box around their extent. Object detection combines these two tasks and localizes and classifies one or more objects in an image. Object classification systems can be very helpful for an automated billing system with minimum human involvement specially if barcode of product not available. Many object recognition algorithms have been developed performance of which varies in terms of accuracy, computation time etc. A pre-trained practical computer vision system network for automated billing application must be simple with very good accuracy as a mismatch will lead to wrong billing having huge economic implication. Even for a particular item that have definite shape can be trained based on features detection and it can be used in billing system. We are working with both options first based on object classification and based on barcode reading.

A barcode is a machine-readable strip of data printed in parallel lines, used to represent a multitude of information. Traditionally, a barcode scanner is used by retailers to keep track of inventory and speed up data entry. Due to their heavy commercial and industrial sector usage, barcode scanning applications have been producer-centric, focusing on improving the efficiency, accuracy, and productivity of supply management. The uses for a barcode reader in the marketplace are numerous and there are endless ways to tie it to the consumer to make exciting and innovative applications.

Barcode is a visual representation of information in the form of bars and spaces on a surface. The bars and spaces are designed with different widths and consist of numbers, characters and symbols such as dot, colon and others. Different combinations of these alphanumeric characters are used to represent information. There are various types of barcodes in use today e.g. Code 128, Code 39, EAN etc. (Brain, 2000) [6]. The successful of barcode technology has been constantly improving in order to accommodate more information in the minimum possible space. Today barcodes are widely used on books and at retail stores in order to keep track of the products available and easy checkout of the products. The barcodes are normally read using scanners using laser beams or cameras [7].

Generally, barcodes are symbols shaped in the form of rectangles which consist of thin or thick parallel lines parallel to each other. Barcodes provide means for automatic rapid data input into the computer. Since the last decade, barcodes are being used in many areas such as market products and electronic devices. The lines on barcodes contain the reference number of the product. This information should be recorded in computers to store each product separately for counting company sales and purchase quantities. When reading barcodes on products using some laser scanning device, a signal is generated by the system and processed in the computer by software. Then this information is used to determine which product is selected. This process provides rapid and reliable sales opportunities to companies for selling their products. There are several types of barcode that being used within the industrial field nowadays. A barcode symbol defines the technical details of a particular type of barcode includes width of bars, character set, method of encoding and checksum specifications. Barcode types can be classified into four type of category and those categories are numeric-only barcodes, alpha-numeric barcodes, 2D barcode and industry standard for barcode and labels.

2.1 Different Types of Barcode System

We know that a barcode is a machine-readable representation of data. It contains information about the object it is associated with. The vast majority of goods sold today have a barcode on them. Although it uses many types of barcode systems in different countries, consumer goods usually use GTIN-13 or EAN-13. Barcode symbolic refers to the technical details of a particular barcode type, such as the width of the bars, character set, coding method, checksum specifications, and so on [Fig 2.1].

Variant types of barcodes are:

i. One-dimensional barcodes
ii. 2-dimensional barcodes.
iii. Industry standards for barcodes and labels.
iv. Rumer's barcodes.
v. Alphanumeric barcodes.

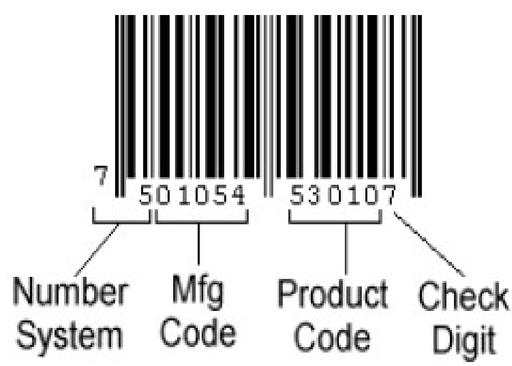


Fig 2.1 1D barcode

Fig 2.2 GTIN-13 barcode

Fig 2.3 Numeric- only barcode

Fig 2.4 2D QR code

GTIN-i3 (Global Trade Item Number). EAN-i3 is a European origin, EAN-i3 is considered an international barcode standard for consumer goods. EAN-13 is a 13-digit barcode with four elements. The first two digits indicate the nationality of the manufacturer. The following five digits are a manufacturer ID assigned by a country's numbering authority. The following five digits are a product ID assigned by the manufacturer. The last digit is a check digit that can be used to check the scanned digits as we can see on the [Fig.2.2]. The only way an EAN-13 barcode differs from a UPC-A barcode is to use a single digit to represent its numbering system [8].

2.2 EAN-13 OVERVIEW

An EAN-13 barcode was originally called European Article Number but now renamed as International Article Number, although the abbreviation EAN has been retained. It is a 13 digit (12 data and 1 check) bar coding standard. It is a superset of the original 12-digit Universal Product Code (UPC) [6] system developed in the United States. Fig. 2.5 shows the EAN-13 barcode. As EAN-13 is considered as a superset of UPC-A, any software or hardware that is capable of reading an EAN-13 symbol can also read a UPCA symbol.

The EAN-13 barcode is used more than any other barcode worldwide. The EAN-13 barcode encodes a GTIN-13 and is used to identify individual items at retail point of sale. It can also be used for trade units sold to consumers, e.g. a case of wine.

i. Encodes a GTIN-13
ii. Does not support attributes, e.g. use-by date and/or barcode identifiers
iii. Used for items we want to sell at retail POS
iv. Scanned by Omni-directional scanners
v. EAN stands for European Article Number
vi. Original barcode was an EAN/UPC symbol, back in 1973.

The following process is followed for binary encoding of data digits into EAN-13 barcode: Firstly, the digits are split into three groups such that the first digit, the first group of 6 and the last group of 6, form three distinct set of numbers. The first group of six is encoded using a scheme where each digit has two possible encodings, one that has even parity and one that has odd parity. The first digit is encoded by selecting a pattern of choices between these two encodings for the next six digits, as shown in Table 2.1 and Table 2.2 (As opposed to other digits, the first digit is not represented by a pattern of bars). Every digit in the last group of six digits is encoded using a single set of bar-patterns that are the same patterns that are used for UPC. If the first digit is zero, all digits in the first group of six are encoded using the patterns that are used for UPC; therefore, a UPC barcode is also considered as an EAN-13 barcode with just the first digit set to zero.

Benefits

i. Can be read right-side-up or upside-down, making the EAN-13 an efficient barcode for high-volume scanning in supermarkets.
ii. Can be used for both retail units (bottle of wine) and trade units (case of wine) sold in store.
iii. Recognized globally, thanks to GS1 standards.
iv. Encodes a GTIN-13, made up of company prefix, item reference and check digit at the end
v. Minimum magnification of 80-100% for retail POS and 150-200% for general distribution. Standard size (100%) is 38mm wide x 25mm high.
vi. Minimum bar heights apply within particular retailer standards.
vii. Quiet Zones are mandatory on left and right sides.
viii. Print the Human Readable Interpretation beneath the barcode and show

all digits encoded in the barcode.

Fig 2.5 EAN-13 Barcode

First digit	First group of 6 digits	Last group of 6 digits
0	LLLLLL	RRRRRR
1	LLGLGG	RRRRRR
2	LLGGLG	RRRRRR
3	LLGGGL	RRRRRR
4	LGLLGG	RRRRRR
5	LGGLLG	RRRRRR
6	LGGGLL	RRRRRR
7	LGLGLG	RRRRRR
8	LGLGGL	RRRRRR
9	LGGLGL	RRRRRR

Table 2.1 ENCODING OF EAN-13

Digit	L-code	G-code	R-code
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

Table 2.2 ENCODING OF L,G,R CODES

2.3 Different Application of Barcodes and its Processing

2.3.1 The Process of Reading Barcodes through Image Processing Using Digital Camera

In this approach the barcode reading is directly done from the camera images by edge detection algorithm to obtain the barcode borders and some threshold mechanism to process the image as binary shapes. In the edge detection approach, the edges of the image is detected and then white areas are given into arrays and threshold process obtained the number from white and black areas of the image.

2.3.2 Readers of Barcodes Using Mobile Phones Camera

In this approach the images of barcode can be captured using the mobile phones camera device. This approach is basically used 2D barcode detection which has four corners. The image which is captured by the embedded camera device of a mobile is in deformed shape. In this project an algorithm called as inverse perspective transformation is used to normalize the deformed shape of an image.

2.3.3 Skew Detection for the 2D Barcode Images

In this approach, two processes are used for detecting 2D barcode. This approach is basically based on the Hough transformation which is an expensive process; it quickly obtains the skew angles of the images and making it applicable foe real-time applications. In this approach, it first does the segmenting process which searches for the barcode region and the life fitting process, fitting the line for the barcode and obtains the skew angle. This approach, shows in the results that, it reduces the running time and it can detect the image even the high noise.

2.3.4 Recognition of Highly Distorted and Resolution Images

In this approach, the barcode recognition can be done even when the image quality is low, blurring, colour saturation where the code density is less than two pixels. The proposed algorithm gives accurate and fast result within the specified time constraint. This algorithm can be used in small portable devices like mobile phones.

2.3.5 Barcode Recognition Systems

This approach is described by Kuroki, tetsuosatou, takayukiyoneoka, Noriakikayaniori, Tadaakikitamura, ypichitagaki [9] which tells that by using the computer technology we can develop the barcode recognitions system which is one of the most important tools for recognizing of their product uniquely for the company. Generally retail shops use the handy barcode reader which is used for small amounts of products, while the big organizations use the barcode recognition systems using image processing for reading their large amount of diverse products.

2.3.6 Barcode Reader Health Applications on Android Mobiles

Navya Shridhar C S, Dr. Mohan K G [18] proposed a new application for reading barcodes by using the camera of a mobile phone. In this approach, the modern android mobiles have barcode scanner for health application which captures the image of the products barcode and sends to the products server which sends back the information about the product. The application does three steps:

1. Process the barcode.
2. Communicate with products server using web services.
3. Server module design.

2.4 Types of Methods for Reading of Barcodes

2.4.1 Pen-type readers

Pen-type readers consist of a light source and photodiode that are placed next to each other in the tip of a pen. To read a barcode, the person holding the pen must move the tip of it across the bars at a relatively uniform speed. The photodiode measures the intensity of the light reflected back from the light source as the tip crosses each bar and space in the printed code. The photodiode generates a waveform that is used to measure the widths of the bars and spaces in the barcode. Dark bars in the barcode absorb light and white spaces reflect light so that the voltage waveform generated by the photodiode is a representation of the bar and space pattern in the barcode. This waveform is decoded by the scanner in a manner similar to the way Morse code dots and dashes are decoded.

Fig 2.6 A handheld barcode scanner

2.4.2 Laser scanners

Laser scanners work the same way as pen-type readers except that they use a laser beam as the light source and typically employ either a reciprocating mirror or a rotating prism to scan the laser beam back and forth across the barcode. As with the pen-type reader, a photo-diode is used to measure the intensity of the light reflected back from the barcode. In both pen readers and laser scanners, the light emitted by the reader is rapidly varied in brightness with a data pattern and the photo-diode receive circuitry is designed to detect only signals with the same modulated pattern.

2.4.3 CCD readers (also known as LED scanners)

CCD readers use an array of hundreds of tiny light sensors lined up in a row in the head of the reader. Each sensor measures the intensity of the light immediately in front of it. Each individual light sensor in the CCD reader is extremely small and because there are hundreds of sensors lined up in a row, a voltage pattern identical to the pattern in a barcode is generated in the reader by sequentially measuring the voltages across each sensor in the row. The important difference between a CCD reader and a pen or laser scanner is that the CCD reader is measuring emitted ambient light from the barcode whereas pen or laser scanners are measuring reflected light of a specific frequency originating from the scanner itself. LED scanners can also be made using CMOS sensors, and are replacing earlier Laser-based readers.

2.4.4 Omni-directional barcode scanner

Omni-directional scanning uses "series of straight or curved scanning lines of varying directions in the form of a starburst, a Lissajous curve[7], or other multiangle arrangement are projected at the symbol and one or more of them will be able to cross all of the symbol's bars and spaces, no matter what the orientation. Almost all of them use a laser. Unlike the simpler single-line laser scanners, they produce a pattern of beams in varying orientations allowing them to read barcodes presented to it at different angles. Most of them use a single rotating polygonal mirror and an arrangement of several fixed mirrors to generate their complex scan patterns.

Omni-directional scanners are most familiar through the horizontal scanners in supermarkets, where packages are slid over a glass or sapphire window. There are a range of different Omni-directional units available which can be used for differing scanning applications, ranging from retail type applications with the barcodes read only a few centimetres away from the scanner to industrial conveyor scanning where the unit can be a couple of metres away or more from the code. Omni-directional scanners are also better at reading poorly printed, wrinkled, or even torn barcodes.

2.4.5 Cell phone cameras

While cell phone cameras without auto-focus are not ideal for reading some common barcode formats, there are 2D barcodes which are optimized for cell phones, as well as QR Codes (Quick Response) codes and Data Matrix codes which can be read quickly and accurately with or without auto-focus.

Cell phone cameras open up a number of applications for consumers. For example:

i. Movies: DVD/VHS movie catalos.
ii. Music: CD catalos – playing an MP3 when scanned.
iii. Book catalos and device.
iv. Groceries, nutrition information, making shopping lists when the last of an item is used, etc.
v. Personal Property inventory (for insurance and other purposes) code scanned into personal finance software when entering. Later, scanned receipt images can then be automatically associated with the appropriate entries. Later, the barcodes can be used to rapidly weed out project copies not required to be retained for tax or asset inventory purposes.
vi. If retailers put barcodes on receipts that allowed downloading an electronic copy or encoded the entire receipt in a 2D barcode, consumers could easily import data into personal finance, property inventory, and grocery management software. Receipts scanned on a scanner could be automatically identified and associated with the

appropriate entries in finance and property inventory software.

vii. Consumer tracking from the retailer perspective (for example, loyalty card programs that track consumers purchases at the point of sale by having them scan a QR code).

2.4.6 Cordless scanner (or Wireless scanner)

A cordless barcode scanner is operated by a battery fit inside it and is not connected to the electricity mains and transfer data to the connected device like PC.

2.5 Other Pre–trained CCN Network (AlexNet) for image classification

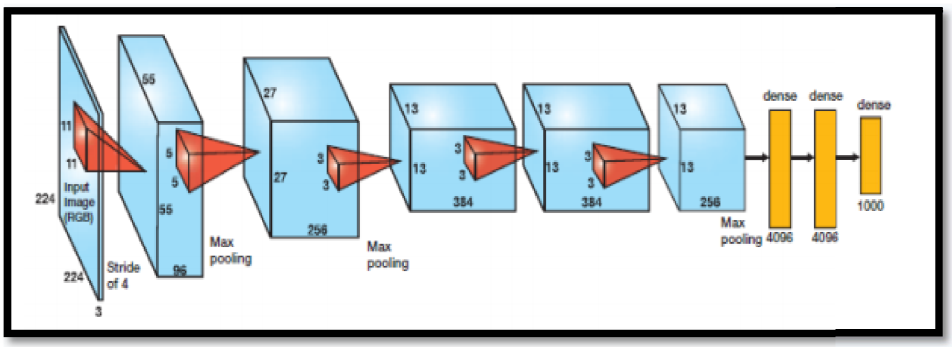


Fig. 2.7 High level architecture of AlexNet

The architecture used in the paper published by Alex Kriszhevsky in 2012 is popularly called AlexNet [10]. It primarily solved the problem of image classification where the input image could belong 1000 different classes and output would be vector of 1000 numbers. The i^{th} element of the output vector would be the probability that input image belongs to the ith class in Network. The input image would be in RGB format with the size of 256 x 256 and consists of 60 million parameters and 650000 neurons. In fig. 2.7 shows that the high level architecture of AlexNet. When occurred less than zero in matrix. AlexNet also solves the problem of over fitting by applying dropout layer after every fully connected layers. Dropout layer has a probability associated with it which is applied to every neuron of the response map independently. This is implemented based on ensembles, with the help of dropout layers different set of neurons are switched off and would represent different architecture which are trained in parallel with weight given to each subset and summation of weight being one. For n neurons attached to dropout the number of subset architecture formed would be 2^n. This tends to avoid.

AlexNet [11]. Network consumes less CPU and memory compared to AlexNet and it is a pre-trained convolutional neural network that is 22 layers deep. The primary intent of the GoogLeNet was the minimize the uniformly increasing networks size and proportionally increasing computational resources, because any uniform increase in the number of their filters results in a quadratic increase of computation. This kind of design unnecessarily wastes the computational resources and larger the network it also leads to over fitting. The solution proposed by GoogLeNet is to move on to sparsely connected network architectures which will replace fully connected network architectures, inside the convolutional layers.

2.6 Summary

This chapter includes information about barcode and different types of barcode. Followed by An overview on EAN-13 Encoding and decoding. It also includes different methods of detecting the barcode and other pre-trained models.

CHAPTER 3

OBJECTIVES AND METHODOLOGY

In this project, we like to have an algorithm for the barcode recognition system by using a web camera or digital camera for capturing the barcode image and then display the barcode information for the user and can be included in the bill. Also if the barcode is not available in items such as vegetables, fruits, and other organic items even other categories of items can also be detected and included in the billing process. For a specific object, we can train a model based on features detection in a large image dataset of the object. Nowadays most of the barcode scanners are using infrared to scan a barcode and barcode is not available for a wide range of essential items. These types of scanners may lead to cost issues as those scanners are expensive and unaffordable to the common user. These cost issues can be solved by using the camera-based system for reading barcodes and object recognition.
It involves two steps for automatic scanning of all the barcodes using webcam:

1. To extract the features of the barcode image
2. To decode the barcode.

It involves three steps for automatic object classification using a webcam.

1. To load pre-trained deep convolution neural network GoogLeNet.
2. If needed we can train a model for a specific item.
3. Load image and classify it based on classification probability.

Unfortunately, images taken by a web camera or digital camera [12] are often of low quality. In this system, the proposed algorithm detects the barcode region using a gradient method, and this region is then cropped. The image pre-processing module transforms the cropped barcode image into a pre-processed grayscale image, then reduces noise, and enhances the contrast between bars and spaces in this pre-processed image. Also for object recognition, we will select only class probabilities greater than 0.5. This is useful to classify objects that are visually distinguishable.

3.1 Project Features

The objective of the project is fulfill all the following features which include barcode detection and validation, object detection and classification and at last database billing is ensured. Such features may address some problems faced at supermarkets. Complete features can be accessible through MATLAB guide. GUI also includes graphics for better look it includes display panel and billing table. Barcode information along with other information needs to update in database file. If object have no barcode in that case any random number can be written in database in place of barcode number, the same needs to be included in realtime.m file against the name of the object. Our project first search for object detection based on image passed to pre-trained GoogleNet database. If not detected barcode based billing can be initiated. Remember to update database file after making any changes in the database.

3.1.1 Barcode Reading and Validation

A simulation platform is a powerful tool for the development and validation of various controllers. Although we have become accustomed to the Computer Vision System Toolbox to create a barcode recognition system that can recognize and interpret a GT1N-13 barcode. The advantages of EAN-13 are that it is not necessary to mark all items with the price and it is also fast and safe because there is no way to make a mistake.

The proposed method first converts the RGB values of the input barcode video to intensity values [Fig: 3.1]. These intensity values are given to the feature calculation block. Feature calculation converts pixel values to features. This is done by assigning values to each pixel. Scan lines can be used to scan barcodes and calculate pixel values from barcode intensity on a given line to a vector. Some lines of the input image are selected and it acts as scan lines. The number and position of the scan lines can be changed. In this project uses three scan lines. The feature calculation level sets black pixels to 1 and white pixels to -1. Barcode recognition block comes after that [Fig: 3.2]. This block consists mainly of three modules. They are bar detect, barcode recognition and barcode comparison block. The bar detection block detects black and white bars from the barcode feature signal. It also calculates the bar width specified as input for the next block, i.e. the barcode recognition block. The barcode recognition module calculates all possible barcode values. These values are compared to the codebook generated in the barcode comparison block. Then comes the barcode validation block, which checks whether the barcode is valid or not. The display block displays the valid barcodes and recognizes them. Before displaying the intensity, values are converted back to RGB. The output will be a barcode video that identifies all barcodes [13-14].

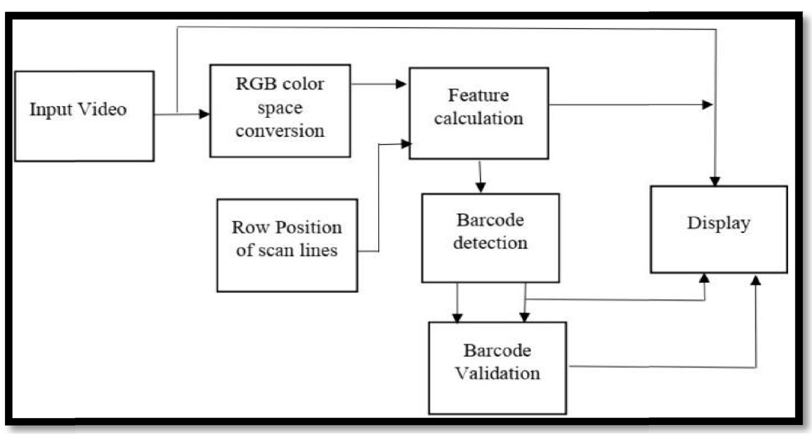


Fig 3.1 Block Diagram for Barcode Reading and Validation

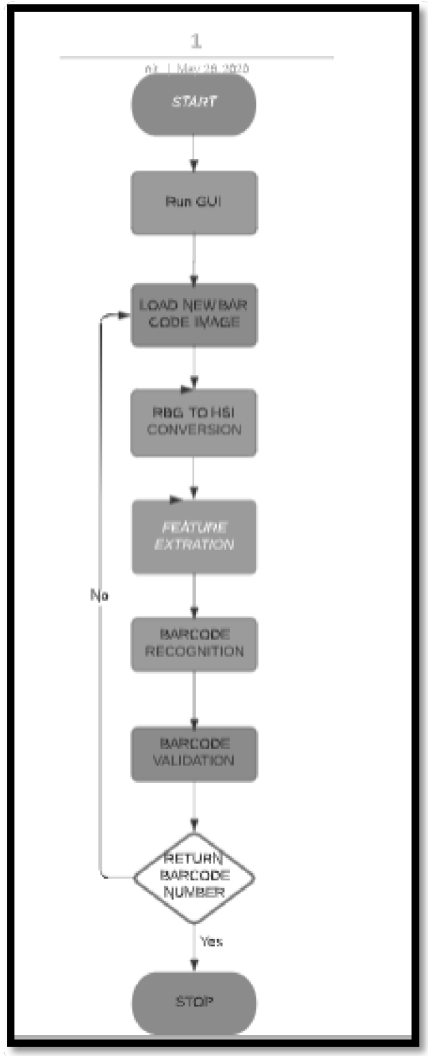


Fig 3.2 System Flowchart for Barcode Reading and Validation

3.1.2 Object Detection Using Pre-Trained Convolution Network Googlenet

GoogLeNet is a convolutional neural network that is 22 layers deep. We can load a pre-trained version of the network trained on either the ImageNet or Places365 data sets. The network trained on ImageNet classifies images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. These networks have learned different feature representations for a wide range of images. The pre-trained networks both have an image input size of 224-by-224.Thus ones the image is classified, the data like image label and score can be used for billing purpose.

To classify new images using GoogLeNet, use classify. This function returns two arguments first image label and second image prediction probability. As shown in [Fig. 3.3]

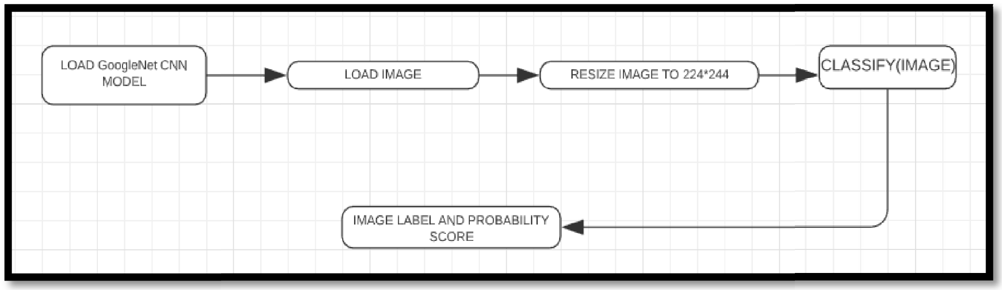


Fig 3.3 Block Diagram for Object Detection using pre-trained convolution network GoogleNet

3.1.3 Billing Process

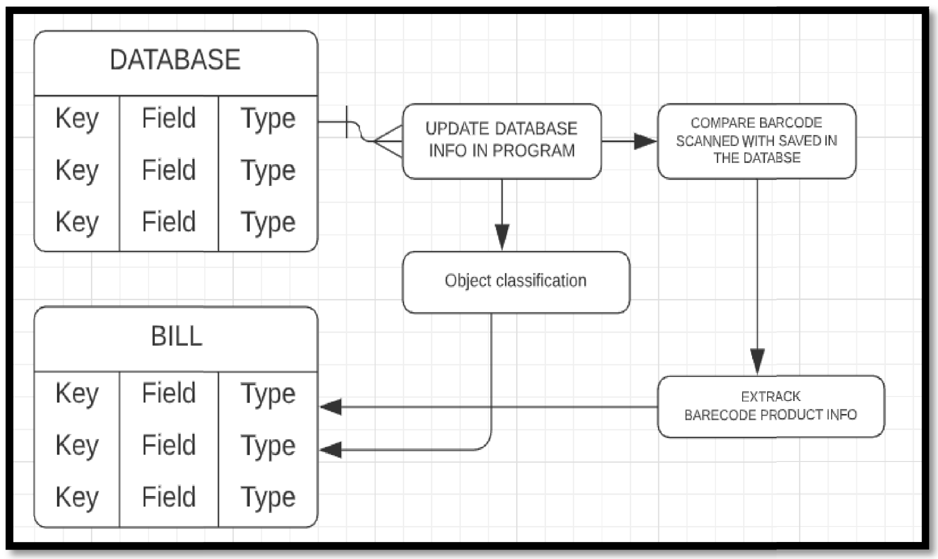


Fig 3.4 Block diagram for Billing Process

Initially, a database of products is created in an excel sheet using MATLAB R2020a. It is very easy to import and export data from an excel sheet to MATLAB. Barcode data can be entered into and read from an excel sheet in MATLAB conveniently as shown in [Fig. 3.4.] The database consists of fields such as barcode number, product name, manufacturer's name and product price. Other fields can also be added as per requirement. (Note that in an excel sheet, the 13 digit barcode number is not displayed fully in the column but is visible in the form of an exponential number. To read the whole number it is necessary to click on that particular cell.) After the whole number is decoded, it is compared with the database and when a match is found the corresponding information is displayed in another excel sheet. As more number of barcodes is scanned their information is added to the same sheet and a bill is created (by adding the price of each product). Similarly in case of object classification based on billing instead of barcode number some random number can be given. As soon as object detected and the corresponding random number under if condition is matched with random barcode number saved in database, the required product details will also be included in the billing.

3.2 System Working Algorithm

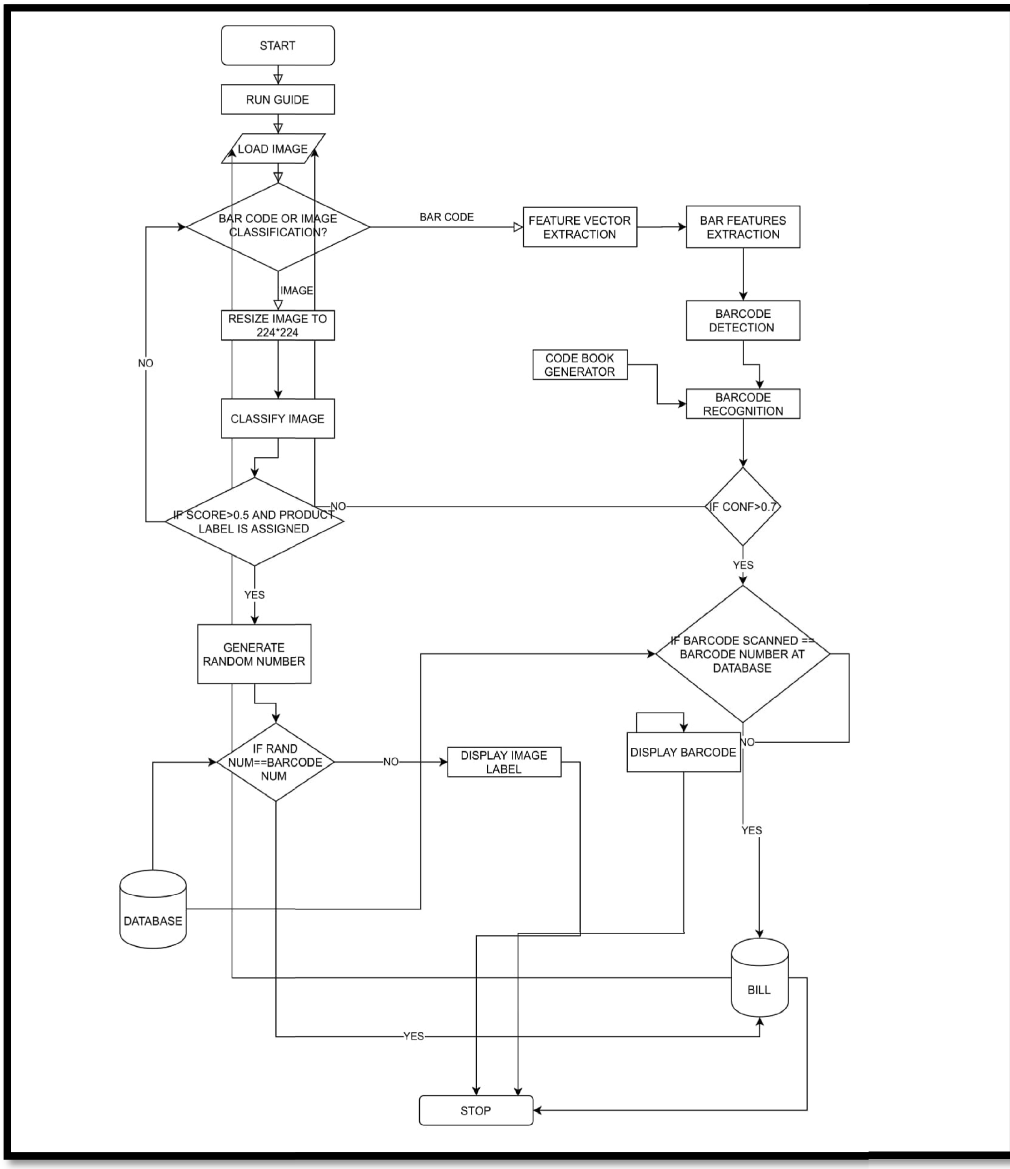


Fig 3.5 System Flow Chart

The working system flow chart is shown in [Fig. 3.5]. That includes all features stated above. The barcode detection example performs a search for selected rows of the input image, called scan lines. Before detection, each pixel of the scan line is preprocessed by converting it to a feature value. The feature value of a pixel is set to 1 if the pixel is considered black, -1 if it is considered white and a value between -1 and 1 otherwise. Once all pixels are transformed, the scan line sequences are analyzed. The example identifies the order and position of the protection patterns and symbols [15]. The symbols are sampled and compared to the codebook to determine the appropriate code.

To compensate for different barcode recognition orientations, the example analyzes from left to right and right to left and selects the better match. If the checksum is correct and a corresponding score is higher than a specified threshold compared to the codebook, the code is considered valid and displayed.

We can change the number and position of the scan lines by changing the value of the "Line Positions of Scan lines" parameter [16-17].

The barcode number is decoded using the array of bar widths. The first digit is always 8 as it is the first digit of the country code and is fixed. As from the Table 3.1, the first group of 6 digits is encoded as LGLGGL and the last group of 6 digits is encoded as RRRRR. Each digit is represented by 4 bars, such that black and white bars are alternate. For the first 6 digits the first bar is always a white bar and next black and so on and for last 6 digits it is exactly reverses. A bar can have four widths from 1 to 4. From the array of bar widths, the first and last three widths are always one indicating the starting and ending of the barcode. The remaining widths are grouped 4 at a time as a particular digit is represented using 4 bars. In Table 2.1, 0 represents white, and 1 represents black. For example, if a group of 4 widths (representing a number from the first 6 digits) is 3112, it is considered as 0001011 (3 is the width of a white bar hence represented by three zeroes). Here, 3 is the width of a white bar, 1 is of black and so on. Thus from the Table 2.1, this first group of 4 widths that is encoded under the L-code is searched for in Table 2.2 under the L-code and gets matched with the code of digit 9. Hence, the first digit is decoded as 9. Similarly, other digits are decoded using the table.

As soon as the barcode is detected and recognized, the barcode number returned from it is compared with the barcode information stored in the database, if matched the corresponding product details along with quantity is passed to the billing section.

If the barcode is not available, in such condition image classification is done using the GoogleNet pre-trained model. We can retrain the model also through transfer learning. In this case, the first image is considered and resized to 224*244. Because the GoogLeNet model pre-trained on ImageNet. That means the network was trained on images cropped to the size 224x224 pixels. Finally, image is classified using the function Classify that returns two arguments first image label and secondly probability score of classification. Based on that a random number can be assigned as a barcode which will be get matched with barcode information store in the database. If matched, that barcode information will be pushed to the bill along with the number of quantity.

3.3 Train Deep Learning Model to Classify New Images

Pre-trained image classification networks have been trained on over a million images and can classify images into 1000 object categories, such as keyboard, coffee mug, pencil, and many animals. The networks have learned rich feature representations for a wide range of images. The network takes an image as input, and then outputs a label for the object in the image together with the probabilities for each of the object categories.

Transfer learning is commonly used in deep learning applications. We can take a pre-trained network and use it as a starting point to learn a new task. Fine-tuning a network with transfer learning is usually much faster and easier than training a network from scratch with randomly initialized weights. We can quickly transfer learned features to a new task using a smaller number of training images.

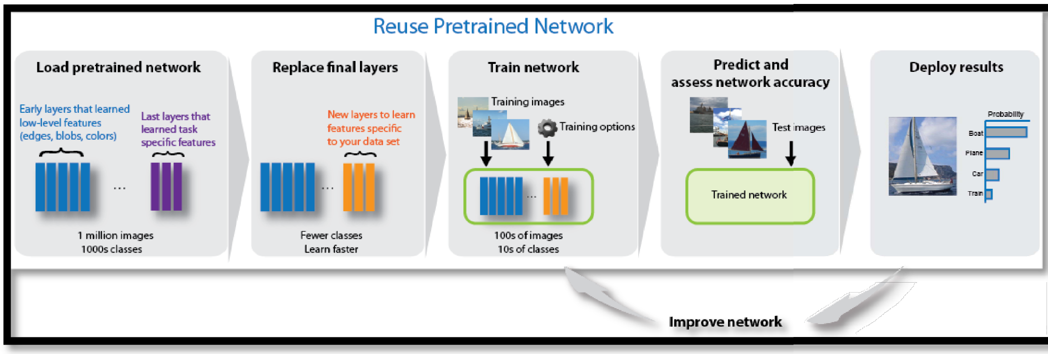


Fig. 3.6 Steps involve to Train Deep Learning Model to Classify New Images

3.3.1 Steps involve to Train Deep Learning Model to Classify New Images [18]

1. Load Data

Unzip and load the new images as an image data store.

2. Load Pre-trained Network

Load a pre-trained GoogLeNet network. Use analyzes Network to display an interactive visualization of the network architecture and detailed information about the network layers. The first element of the Layers property of the network is the image input layer. For a GoogLeNet network, this layer requires input images of size 224-by-224-by-3, where 3 is the number of color channels.

3. Replace Final Layers

The convolutional layers of the network extract image features that the last learnable layer and the final classification layer use to classify the input image. These two layers, 'loss3-classifier' and 'output' in GoogLeNet, contain information on how to combine the features that the network extracts into class probabilities, a loss value, and predicted labels. To retrain a pre-trained network to classify new images, replace these two layers with new layers adapted to the new data set.

Extract the layer graph from the trained network.

The classification layer specifies the output classes of the network. Replace the classification layer with a new one without class labels. Train Network automatically sets the output classes of the layer at training time.

4. Freeze Initial Layers

The network is now ready to be retrained on the new set of images. Optionally, we can "freeze" the weights of earlier layers in the network by setting the learning rates in those layers to zero. During training, train Network does not update the parameters of the frozen layers. Because the gradients of the frozen layers do not need to be computed, freezing the weights of many initial layers can significantly speed up network training. If the new data set is small, then freezing earlier network layers can also prevent those layers from over fitting to the new data set. Extract the layers and connections of the layer graph and select which layers to freeze. In GoogLeNet, the first 10 layers make out the initial 'stem' of the network. Use the supporting function freezeWeights to set the learning rates to zero in the first 10 layers. Use the supporting function createLgraphUsingConnections to reconnect all the layers in the original order. The new layer graph contains the same layers, but with the learning rates of the earlier layers set to zero.

5. Train Network

The network requires input images of size 224-by-224-by-3, but the images in the image data store have different sizes. Use an augmented image data store to automatically resize the training images. Specify additional augmentation operations to perform on the training images: randomly flip the training images along the vertical axis and randomly translate them up to 30 pixels and scale them up to 10% horizontally and vertically. Data augmentation helps prevent the network from over fitting and memorizing the exact details of the training images.

To automatically resize the validation images without performing further data augmentation, use an augmented image data store without specifying any additional preprocessing operations.

Specify the training options. Set InitialLearnRate to a small value to slow down learning in the transferred layers that are not already frozen. In the previous step, we increased the learning rate factors for the last learnable layer to speed up learning in the new final layers. This combination of learning rate settings results in fast learning in the new layers, slower learning in the middle layers, and no learning in the earlier, frozen layers.

Specify the number of epochs to train for. When performing transfer learning, we do not need to train for as many epochs.

An epoch is a full training cycle on the entire training data set. Specify the mini-batch size and validation data. Compute the validation accuracy once per epoch.

Train the network using the training data. By default, trainNetwork uses a GPU if one is available (requires Parallel Computing Toolbox and a CUDA enabled GPU with compute capability 3.0 or higher). Otherwise, trainNetwork uses a CPU. We can also specify the execution environment by using the 'ExecutionEnvironment' name-value pair argument of trainingOptions. Because the data set is so small, training is fast.

6. Classify Validation Images

Classify the validation images using the fine-tuned network, and calculate the classification accuracy.

3.4 Conceptual Analysis for Barcode Localization and Image Pre-Processing

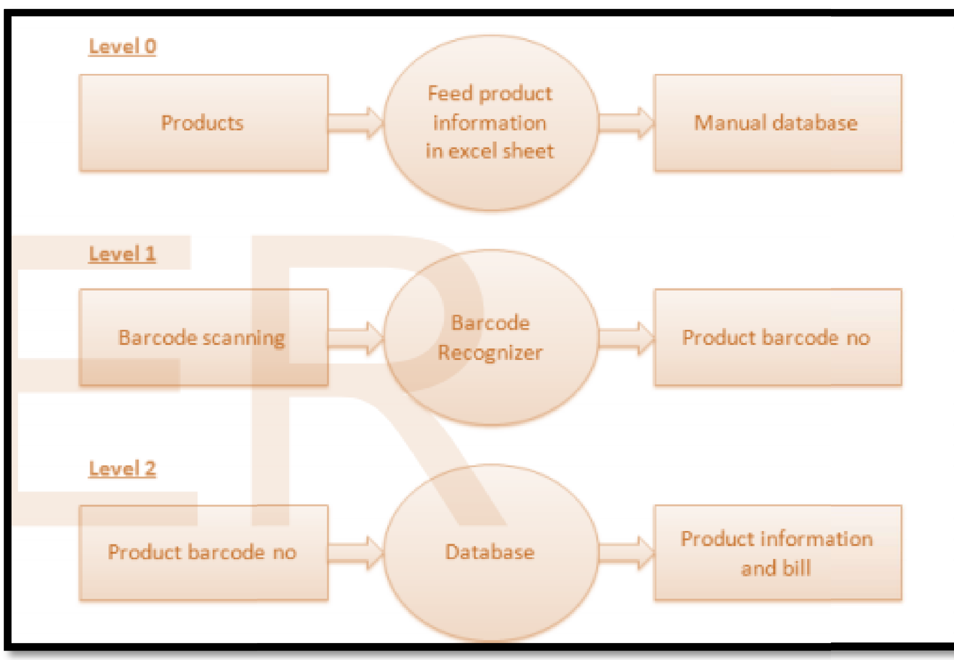


Fig. 3.7 Conceptual analysis diagram of project

We have presented an algorithm for decoding (localization and reading) barcodes through a webcam, using image processing techniques such as filtering, cropping, and contrast enhancement and edge detection. This localization technique is fast and accurate even for cluttered images. Barcodes having a flat surface and a width to height ratio between 2 to 3 are successfully decoded. We have implemented this algorithm in MATLAB R2020a and can work on any of the later versions of MATLAB. We have scanned multiple products by executing this algorithm in a loop, and a final bill of products is created in an excel sheet.

3.4.1 Barcode Localization at Level 1

We found out that the simple and fast algorithm presented by Orazio Gallo and Roberto Manduchi [2] provides excellent results even in challenging situations. This localization algorithm assumes that the image of the barcode is captured by the camera-oriented system such that its vertical axis is approximately parallel to the bars as shown in [Fig3.8]. This image is further converted to a grayscale image as shown in Fig. 3.8. Thus, with respect to the barcode, one should expect an extended region represented by strong horizontal gradients and weak vertical gradients. Accordingly, we compute the horizontal and vertical derivatives, Ix(n) and Iy(n), at each pixel n. Then we combine them in a nonlinear fashion given as $Ie(n) = |Ix(n)| - |Iy(n)|$ It is acceptable to assume that many points within a barcode should have a large value of Ie(n). We run a block filter of size 31x 31 over Ie (n), obtaining the smoothed map Is(n).

Fig. 3.8 Original Image

Fig. 3.9 Gray scale Image

Fig. 3.10 Gaussian filtered Image

Fig. 3.11 Barcode region Image

The size of the filter was selected based on the range of the size of the input barcode images and the minimum size of the barcode readable by our method. Note that block filtering can be implemented efficiently so that only a few operations per pixel are required. In the end, we binarize Is(n) with a single threshold that is selected using the method proposed by Otsu [13], and the output image is as shown in [Fig. 3.10]. This binarization is only used to localize the barcode, while the decoding is performed on the greyscale image. Due to thresholding, the map Is(n) may contain more than one blob. Instead of computing the connected components of the thresholded map, we just select the pixel, No that maximizes Is(n), with the assumption that the correct blob (the one corresponding to the barcode) contains such a pixel. In our research, this assumption was always found to be correct. We then expand a vertical and a horizontal line from No, and form a rectangle with sides parallel to the axes of the barcode image and containing the intersections of these lines, with the edge of the blob. Note that a quiet zone, a white region around the barcode that facilitates localization, borders the leftmost and rightmost bars of a barcode. The quiet zone and the large size of the block filter, ensures that the vertical sides of the rectangle fall outside the area of the barcode by at least a few pixels as shown in [Fig. 3.11].

3.4.2 Image Pre-Processing at level 1

1. **Cropping**

   When an image is taken for scanning, along with the barcode, other unnecessary information like letters and words surrounding the barcode also gets included. Thus, it is very crucial to remove the noise and get only the barcode region for proper scanning and decoding. After barcode localization, only the barcode region is visible and surrounding information is eliminated by making all other pixels value to 0. Then, we perform cropping by observing the intensity of each pixel and extracting the rows of a barcode, i.e. the pixels having the intensity greater than 0. [Fig. 3.11] depicts the cropped image of barcode form [Fig. 3.11].

Fig. 3.12 Cropped Image

2. **Contrast Enhancement**

In a real-time image of a barcode, the contrast between white
In a real-time image of a barcode, the contrast between white
and black bars is and black bars are poor due to blurriness. Hence, it is necessary to improve the contrast of the image to distinguish between bars. This is done by making the black bars one shade darker on the grayscale compared to white bars as shown in [Fig. 3.13].

Fig. 3.13 Contrast improved Image

3. **Converting to ideal barcode image**

The contrast-enhanced image is firstly binarized. But due to the real-time acquisition, this image is distorted. To convert into an ideal image, every column of an image is scanned and checked for a maximum number of pixel intensity of either 0 or 1. If a particular column has more number of pixels with intensity 1 than 0, then the whole column is converted to pixels with intensity 1. [Fig. 3.14] shows an ideal barcode image.

Fig. 3.14 Ideal Barcode Image

4. **Edge Detection**

An edge is a pixel at which there is a sudden intensity change. An array of such pixels is created that forms the edges of the bars of the barcode image. Bar widths are calculated by subtracting consecutive elements of the array of edges. These bar widths are given as input to the decoding algorithm.

2.5 System Development Methodology

Framework advancement technique is a procedure through which an item will get finished or an item gets freed from any issue. Programming improvement process is depicted as various stages, systems and steps that give the entire programming. It takes after arrangement of steps which is utilized for item advance. The advancement technique followed in this venture is smart billing based on object classification and barcode reading.

Requirement Analysis: This phase is concerned about collection of requirements of the system. This process involves generating document and requirement review. This includes information about MATLAB Image processing tool followed by MATLAB guide and Deep Learning Tools.

System Design: Keeping the requirements in mind the system specifications are translated in to a software representation. In this phase we emphasizes on: - algorithm, data structure, software architecture etc. development over MATLAB.

Coding: In this phase we started coding in order to give a full sketch of product. In other words, system specifications are only converted in to machine readable compute code.

Implementation: The implementation phase involves the actual coding or programming of the software. The output of this phase is typically having a GUI which tries to maximum features as included in project objective.

Testing: In this phase all programs (models) are integrated and tested to ensure that the complete system meets the software requirements. The testing is concerned with verification and validation.

Maintenance: The maintenance phase is the longest phase in which the software is updated to fulfill the changing customer need, adapt to accommodate change in the external environment, correct errors and oversights previously undetected in the testing phase, enhance the efficiency of the software. Since we have released only first version (1.0). We expect to bring changes as per the customer needs.

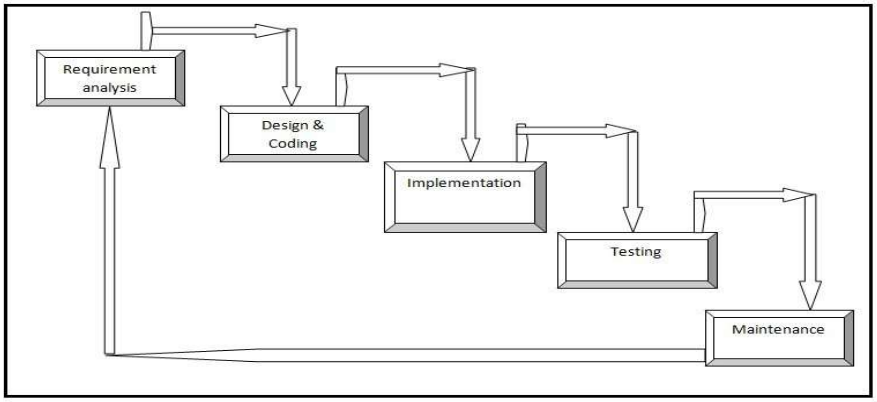


Fig 2.15 System methodology

2.6 Reasons for Choosing this Project

i. Clear project objectives.
ii. Stable project requirements.
iii. Progress of system is measurable.
iv. Strict sign-off requirements.
v. Helps us to be perfect.
vi. Logic of software development is clearly understood.
vii. Better resource allocation.
viii. Improves quality. The emphasis on requirements and design before writing a single line of code ensures minimal wastage of time and effort and reduces the risk of schedules page.
ix. Less human resources required as once one phase is finished those people can start working on to the next phase.

2.7 Summary

This chapter includes features involves in project followed by system flow chart and algorithm behind. It also includes Conceptual Analysis for Barcode Localization and Image Pre-Processing and transfer training for a pre-trained model. Here, we also discuss about system methodology development and reasons for choosing this project.

CHAPTER 4

REQUIREMENTS SPECIFICATION

The project have software requirement as the platform where algorithm can implemented with the help of certain tools that are supported by the software. There will be some hardware requirements as per the software specifications.

4.1 Software Requirements

4.1.1 MATLAB

Millions of engineers and scientists worldwide use MATLAB to analyze and design the systems and products transforming our world. The matrix-based MATLAB language is the world's most natural way to express computational mathematics. Built-in graphics make it easy to visualize and gain insights from data. The desktop environment invites experimentation, exploration, and discovery. These MATLAB tools and capabilities are all rigorously tested and designed to work together.

MATLAB helps ideas beyond the desktop. We can run were analyses on larger data sets, and scale up to clusters and clouds. MATLAB code can be integrated with other languages, enabling we to deploy algorithms and applications within web, enterprise, and production systems.

4.1.2 MATLAB Tool Box

Inside MATLAB, there are toolboxes with different functions and features which being provided for the users. For this project, it only used two of MATLAB toolboxes and those are Image Processing Toolbox and Image Acquisition Toolbox. These two toolboxes will be created to provide a function which related with image processing and image acquisition task. Here the other list of toolboxes besides image processing and image acquisition that available in MATLAB [21].

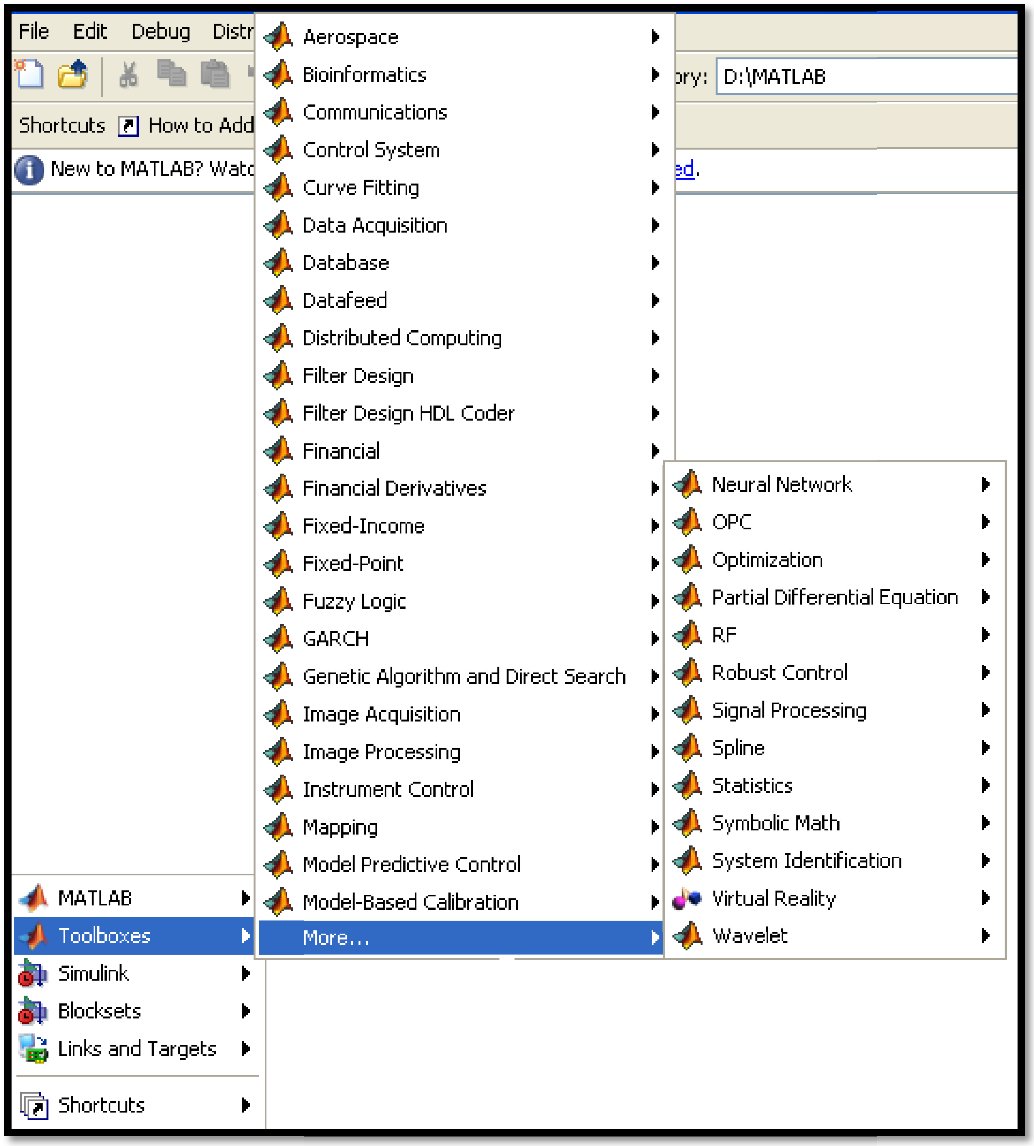


Fig 4.1: MATLAB Toolboxes

4.1.3 Image Processing Toolbox

The Image Processing Toolbox is a collection of functions that extend the capability of the MATLAB numeric computing environment. The toolbox supports a wide range of image processing operations, including spatial image transformations, morphological operations, neighbourhood and block operations, linear filtering and filter design, transforms, image analysis and enhancement, image registration, de-blurring and region of interest operations [22].

The image processing toolbox is used when it comes to the image processing process on the barcode. Here some of the MATLAB command which were used inside project system.

Image Processing Command	Description
imread	Read image for graphic file.
imshow	Display image in handle graphic figure.
im2bw	Convert image to binary image by thresholding.
medfilt2	2-D median filtering
imerode	Erode image
imdilate	Dilate image
rgb2gray	Convert RGB image or colormap to grayscale.

Table 4.1 Image Processing Commands and Descriptions

4.1.4 Image Acquisition Toolbox

With this toolbox, it is able to help to connect and conFig hardware, preview the acquisition and acquire and visualize image data. The hardware in this project would be the webcam. Image acquisition toolbox also was needed in order to capture and show the webcam by default. With the correct MATLAB command, the webcam was able to set to the default state and the image also can be captured. By using image acquisition toolbox and image processing toolbox, MATLAB would able to provide a complete environment for developing customized imaging applications.

Image Acquisition Command	Description
clear	Clear image acquisition object from memory
getsnapshot	Immediately return single image frame
imaqreset	Disconnect and delete all image acquisition objects.
videoinput	Create video input object.
get	Image acquisition object properties
imaqwinfo	Information about available image acquisition hardware

Table 4.2 Image Acquisition Command and Description

4.1.5 MATLAB Guide

Graphical user interfaces (GUIs), also known as apps, provide point-and-click control of software applications, eliminating the need for others to learn a language or type commands in order to run the application. We can share apps both for use within MATLAB and also as standalone desktop or web apps.

We can choose from the following three ways to create an app in MATLAB:

- **Convert a script into a simple app:** Choose this option when we want to share a script with students or colleagues and allow them to modify variables using interactive controls.
- **Create an app interactively**: Choose this option when we want to create a more sophisticated app using a drag-and-drop environment to build the user interface.
- **Create an app programmatically:** Choose this option when we want to create an app's user interface by writing the code self.

4.1.6 Some Guide functions used in our project

Push Button

This code is an example of a push button callback function in GUIDE. Associate this function with the push button Callback property to make it execute when the end user clicks on the push button.

```
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
display('Goodbye');
close(gcf);
```

The first line of code, display('Goodbye'), displays the string, 'Goodbye', in the Command Window. The next line gets a handle to the UI window using gcf and then closes it.

Check Box

This code is an example of a check box callback function in GUIDE. Associate this function with the check box Callback property to make it execute when the end user clicks on the check box.

```
function checkbox1_Callback(hObject, eventdata, handles)
% hObject handle to checkbox1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of checkbox1
if (get(hObject,'Value') == get(hObject,'Max'))
display('Selected');
else
display('Not selected');
end
```

The check box's Value property matches the Min property when the check box is not selected. The Value changes to the Max value when the check box is selected. This callback function gets the check box's Value property and then compares it with the Max and Min properties. If the check box is selected, the function displays 'Selected' in the Command Window. If the check box is not selected, it displays 'Not selected'.

Panel

Make the Panel Respond to Button Clicks

We can create a callback function that executes when the end user right-clicks or leftclicks on the panel. If we are working in GUIDE, then right-click the panel in the lawet and select **View Callbacks** > **ButtonDownFcn** to create the callback function. This code is an example of a ButtonDownFcn callback in GUIDE.

```
function uipanel1_ButtonDownFcn(hObject, eventdata, handles)
% hObject handle to uipanel1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
display('Mouse button was pressed');
```

When the end user clicks on the panel, this function displays the text, 'Mouse button was pressed', in the Command Window.

Table

This code is an example of the table callback function, CellSelectionCallback. Associate this function with the table CellSelectionCallback property to make it execute when the end user selects cells in the table.

```
function uitable1_CellSelectionCallback(hObject, eventdata, handles)
% hObject handle to uitable1 (see GCBO)
% eventdata structure with the following fields
% Indices: row and column indices of the cell(s) currently selected
% handles structure with handles and user data (see GUIDATA)
data = get(hObject,'Data');
indices = eventdata.Indices;
r = indices(:,1);
c = indices(:,2);
linear_index = sub2ind(size(data),r,c);
selected_vals = data(linear_index);
selection_sum = sum(sum(selected_vals))
```

When the end user selects cells in the table, this function performs the following tasks:

1. Gets all the values in the table and stores them in the variable, data.
2. Gets the indices of the selected cells. These indices correspond to the rows and columns in data.
3. Converts the row and column indices into linear indices. The linear indices allow we to select multiple elements in an array using one command.
4. Gets the values that the end user selected and stores them in the variable, selected vals.

Axes

The code in this section is an example of an axes ButtonDownFcn that triggers when the end user clicks on the axes.

```
function axes1_ButtonDownFcn(hObject, eventdata, handles)
% hObject handle to axes1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
pt = get(hObject,'CurrentPoint')
```

The coordinates of the pointer display in the MATLAB Command Window when the end user clicks on the axes (but not when that user clicks on another graphics object parented to the axes).

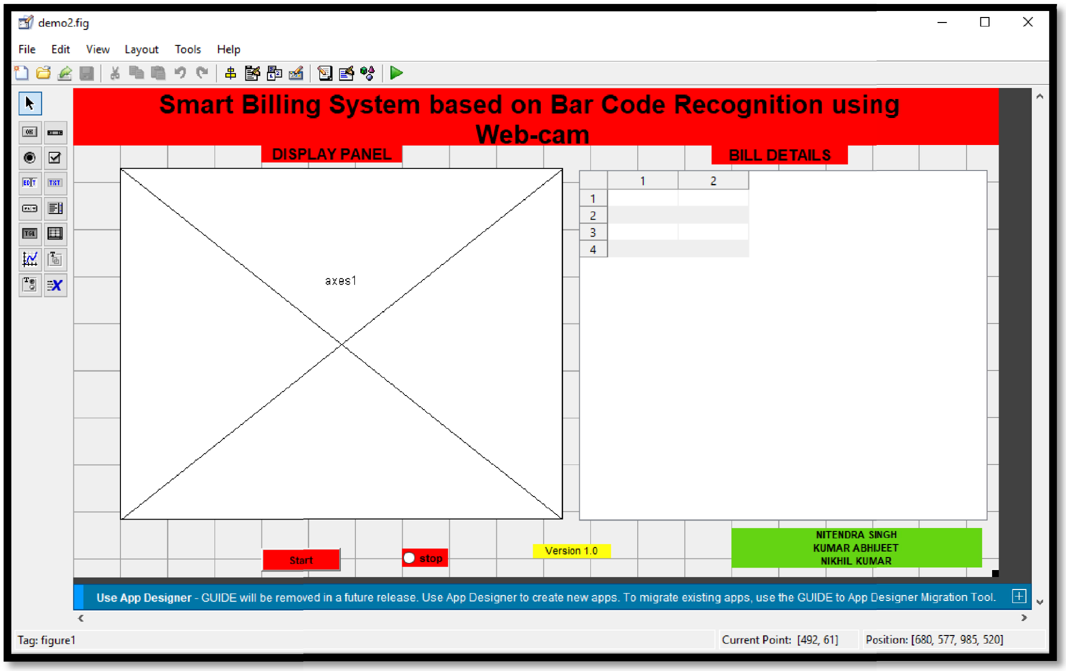


Fig 4.2 System GUI Design

4.1.7 IP WEBCAM

Using wer Android device as an IP camera

Using the popular and free Android application IP Webcam we can turn wer android device into a fully functioning wireless IP camera that we can use within iSpy, including video, audio, talk, text-to-speech and commands.

How do I add it?

i. Connect wer android device to wer wireless network
ii. Install IP Webcam on wer android device
iii. Start the IP Webcam application. See Video preferences and set the video resolution to a smaller format (example 740x480)
iv. Press "Start Server" at the bottom of the main menu

4.1.8 Deep Learning Toolbox

Deep Learning Toolbox provides a framework for designing and implementing deep neural networks with algorithms, pre-trained models, and apps. We can use convolutional neural networks (ConvNets, CNNs) and long short-term memory (LSTM) networks to perform classification and regression on image, time-series, and text data. We can build network architectures such as generative adversarial networks (GANs) and Siamese networks using automatic differentiation, custom training loops, and shared weights. With the Deep Network Designer app, we can design, analyze, and train networks graphically. The Experiment Manager app helps we manage multiple deep learning experiments, keep track of training parameters, analyze results, and compare code from different experiments. We can visualize layer activations and graphically monitor training progress.

We can exchange models with TensorFlow and PyTorch through the ONNX format and import models from TensorFlow-Keras and Caffe. The toolbox supports transfer learning with DarkNet-53, ResNet-50, NASNet, SqueezeNet and many other pre-trained models.

We can speed up training on a single- or multiple-GPU workstation (with Parallel Computing Toolbo), or scale up to clusters and clouds, including NVIDIA GPU Cloud and Amazon EC2 GPU instances (with MATLAB Parallel Serve).

4.2 Hardware Requirements

i. Real-time implementation is done using the Laptop and Camera
ii. Laptop Specifications: Dell Inspiron 15, Processor Intel (R) Core (TM) i5-5200U, 2.2 GHz
iii. Camera Specifications: SONY IMX HD 1080p

4.3 Summary

This chapter includes various software requirements including various tools that will provide various functions support to project. It also includes hardware requirement.

CHAPTER 5

SYSTEM IMPLEMENTATION

System implementation includes various blocks and their functioning. It also includes certain pre-processing formalities relating to both image classification and barcode detection. Data base approach is also going to discuss. And finally about Final bill generation.

5.1 Barcode recognition/ detection

5.1.1 Feature Calculation

The function block calculates the mean values and the standard deviation for the pixels in the barcode range. The feature function converts pixels on a specific line into a feature vector parameter. We calculated the feature vector for all pixels in the scan lines. We estimated the range of pixel intensity values in the barcode range. A pixel is classified as black if its value is f-low or less. Black pixels are set to 1. Another pixel is classified as white if its value is f-high or greater. White pixels are set to -1. Values greater than f-high and values less than f-low are saturated. The remaining pixels are set proportionally to values between -1 and 1. Next to h is a Gaussian filter. It is used to smooth the image history that identifies the barcode area [20-22].
Such as

f_high = f_mean + f_std * RATIO_STD; eqn5.1

f_low = f_mean - f_std * RATIO_STD; ......eqn 5.2

5.1.2 Barcodes Detection

1. **Bar Detection**
   This function is "BAR" detection and bar of barcode feature signal. Firstly, tries to find a black bar. If the black bars do not exist, then it will be zero. Then a contiguous sequence of pixels with zero or positive feature value is considered a black bar. Secondly, we found a white bar. Similarly, a contiguous sequence of pixels with a negative feature value is considered a white bar [20-22].

2. **Detection**
   Actually, the detection method gives the sequence of indexes to barcode protection bar parameters. And also checks the maximum number of protective pattern separators and the maximum number of sequences of protection patterns of constant values. Such as,

   1.Count of both black and white bars.
   2.Calculate sequence and its length.

3.Skip the first bar if it is invalid.
4.Find out all possible separators.

3. Codebook Generator

The CODEBOOK function generates the codebooks for the GTIN-13/EAN-13 or UPC-A. According to the codebook for the first digit encoded in the L/G patterns and for the other 12 digits 0 to 1. Then convert the codebook of symbols to digits [20-22].

4. Barcode Recognition

The barcode recognition block picks up the barcodes and attempts to match the barcode with the number of pixels generated from bar detection. Egotistic scan line orders from left to right to detect digits 8 to 13. Similarly, from right to left, detects scan line order from digits 8 through 13. Then the block used the scan line direction (left to right or right to left) to find the potential match for digits 2 through 7, although there 1 and 2 is I/O port numbers, 1 is F and 2 is threshold for bar width variance in [Fig.5.1]. Finally, the block recognizes the strings, digits and bars, then it could be recognized a complete barcode [20-22].

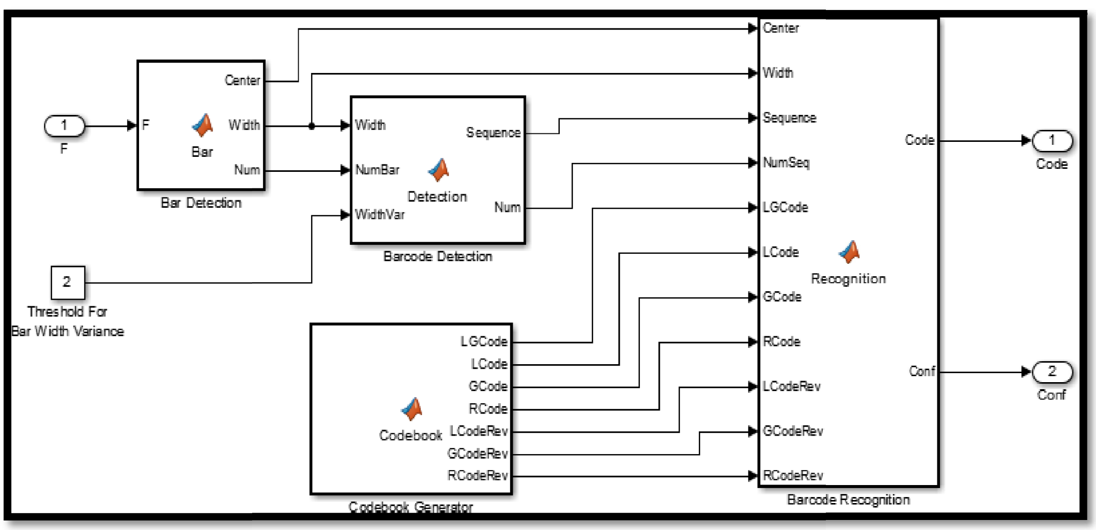


Fig 5.1 Barcode recognition/ detection block

5.2 Webcam Display

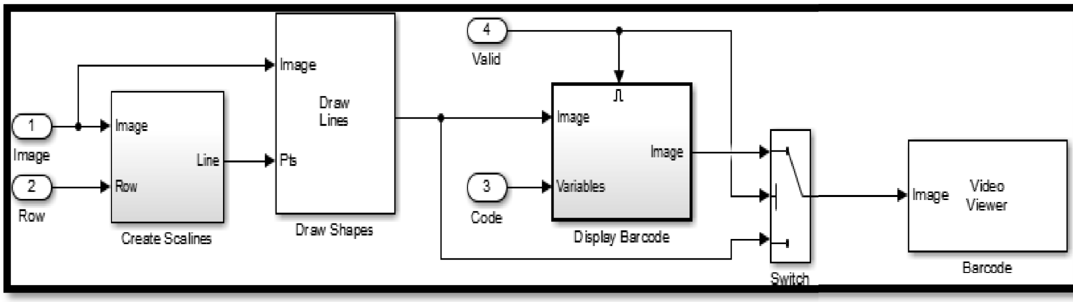


Fig 5.2 Webcam display

The input image comes from a constant image block. The video display block sends video data to a DirectX-supported video output device or camera. Alternatively, we can send the video data to a separate monitor or display the data in a window on our own computer screen shown in [Fig.5.2]. We used the Pts port to specify the appropriate shape coordinates. And also defined the color value of the shape's border. As well, choose to fill that shape and specify its opacity. The input into the Pts port must be an M-by-2L matrix in which each row specifies a different polyline. Each line must be (x1, y1, x2, y2, xL, yL) shape, which specifies the indentation and downstream points in recurrently order [22].

5.3 Image Classification using GoogleNet

i. Load the pre-trained GoogLeNet network. This step requires the Deep Learning Toolbox. If we do not have the required support packages installed, then the software provides a download link.

net = googlenet;

ii. The image that we want to classify must have the same size as the input size of the network. For GoogLeNet, the first element of the Layers property of the network is the image input layer.

iii. The network input size is the InputSize property of the image input layer.

inputSize = net.Layers(1).InputSize

iv. Resize the image to the input size of the network by using imresize. Classify the image and calculate the class probabilities using classify.

v. Based on class and class probability if greater than 0.5 such class item can be added for the billing process.

5.4 Database

Initially, a database of products is created in an excel sheet using MATLAB R2020a. It is very easy to import and export data from an excel sheet to MATLAB. Barcode data can be entered into and read from an excel sheet in MATLAB conveniently. The database consists of fields such as barcode number, product name, manufacturer's name and product price. Other fields can also be added as per requirement. (Note that in an excel sheet, the 13 digit barcode number is not displayed fully in the column but is visible in the form of an exponential number. To read the whole number it is necessary to click on that particular cell.)

After the whole number is decoded, it is compared with the database and when a match is found the corresponding information is displayed in another excel sheet. As more number of barcodes is scanned their information is added to the same sheet and a bill is created (by adding the price of each product) as shown in [Fig. 5.3].

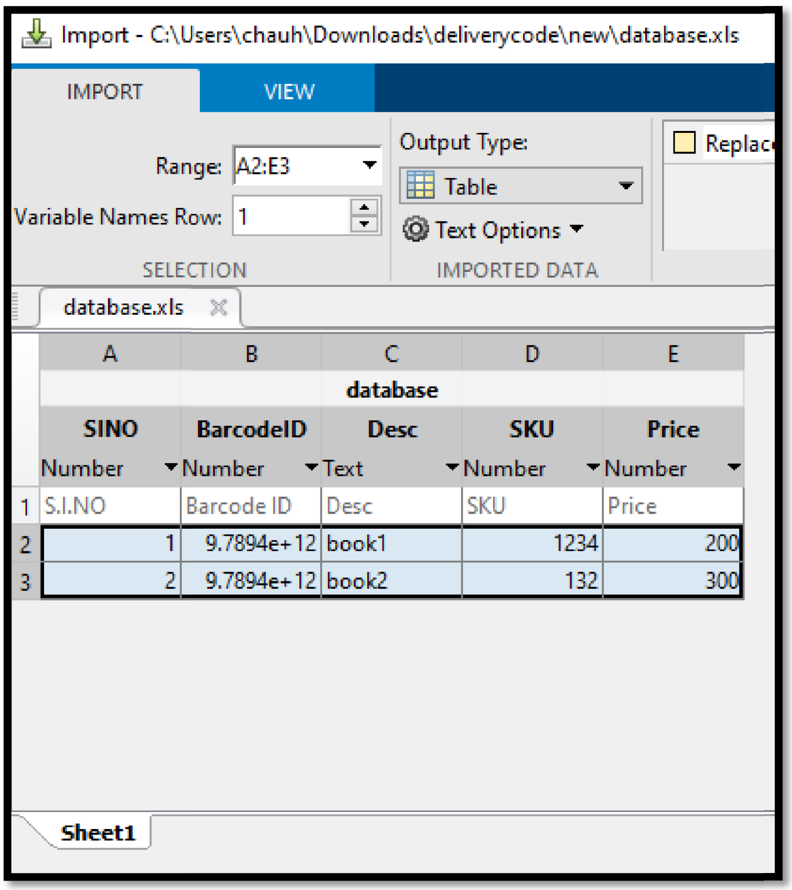


Fig 5.3 Final Bill

5.5 Summary

This chapter includes implementation of different algorithms for the detection and recognition of barcode and image classification by transforming them into blocks of programming codes. Further, it also includes steps involve in the setup of webcam and database.

Chapter 6

RESULTS

The project proposed smooth barcode detection and image classification of real time images with MATLAB as well as image processing. Here we will discuss the working of project steps and result analysis.

6.1 Experimental Results for Barcode Scanning

We have developed the algorithm of barcode recognition consisting with RGB block, barcode recognition block, Image classification and billing. There are some issues that need to be counteracted in order to develop this barcode detection system with MATLAB. They are scan lines that have been used to detect barcodes that appears in red. If an EAN-13 is correctly detected and validated, the code appears at the top of the image. That can be used for billing process. The barcode recognition system is expected to detect barcodes in each direction. So, we illustrated how to make a simulation and we got the result shown in [Fig.6.1].

At the end of this project session, the program encoding was successfully written until the program was able to display barcode images on the frame and also converted the colored image together into red, blue and green (RBG). This system can effectively detect EAN-13 and UPC code with minimal noise for increasing performance.

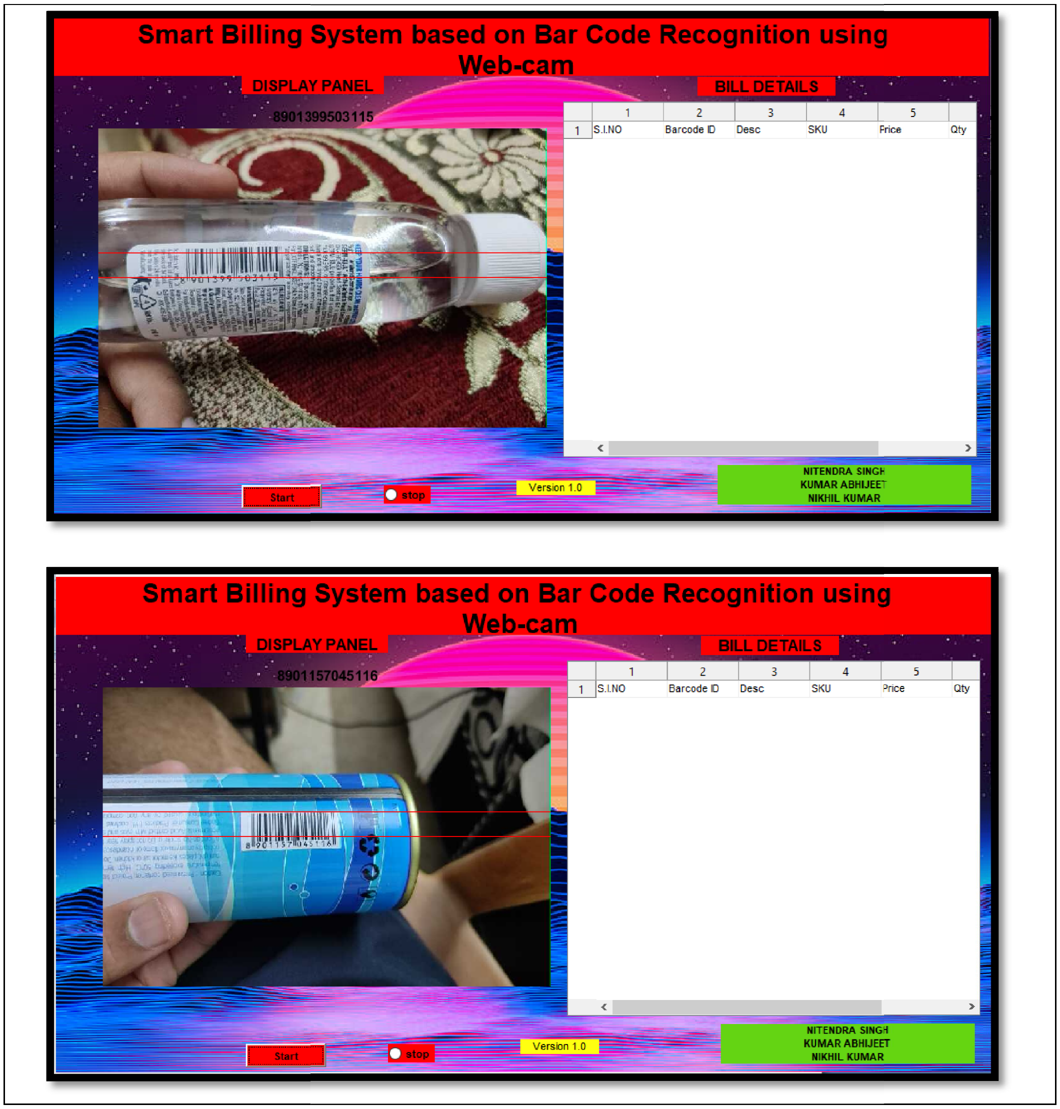


Fig 6.1 Experimental Results of barcode scanning

6.2 Execution Procedure

1. A database of products is created in an excel sheet using MATLAB R2020a. It is very easy to import and export data from an excel sheet to MATLAB As shown in [Fig. 6.2]. Barcode data can be entered into and read from an excel sheet in MATLAB conveniently. The database consists of fields such as barcode number, product name, Stock keeping unit and product price. Other fields can also be added as per requirement.

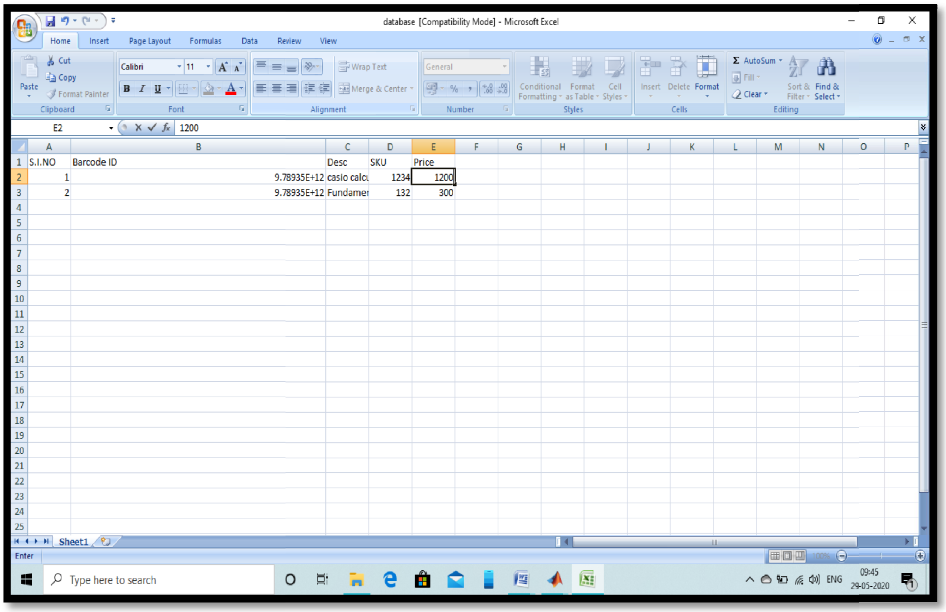


Fig. 6.2 Database Excel File

2. Once the database file is saved. Another thing required is to updatedatabase by excuting the matlab program updatedatadase.m. This will generate MATLAB .mat file i.e alldata.mat. This file saves all the attributes of database excel file in the form of table where each attribute can be access through iteration statement. As shown in [Fig. 6.3].

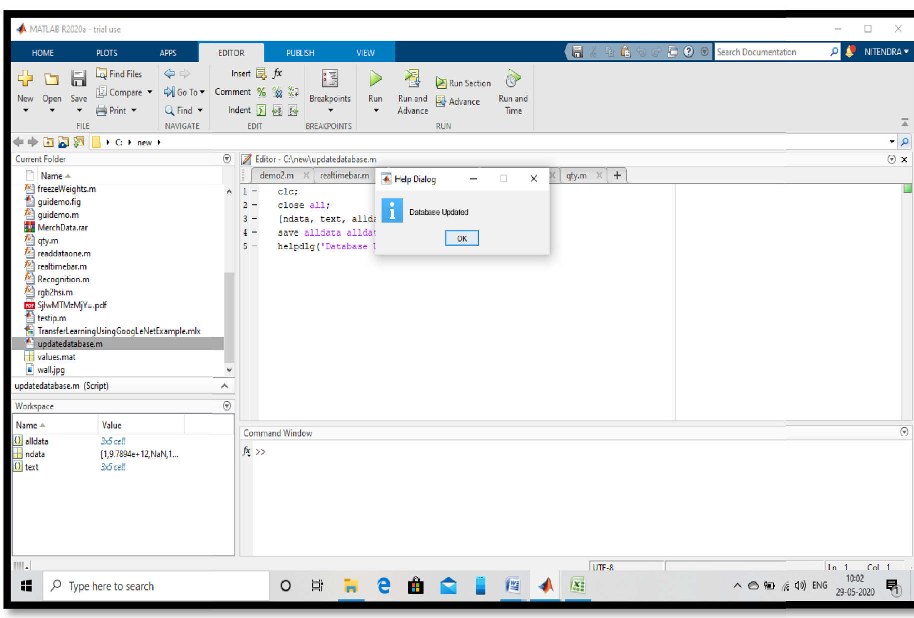


Fig 6.3 Updating database into MTLAB

3. Run demo2.m file in order to access the GUI of the project. It consist of Display panel to access the camera and table to show real time billing status. Connect wer android device to wer wireless network. Install IP Webcam on wer android device. Start the IP Webcam application. See Video preferences and set the video resolution to a smaller format (example 740x480). Press "Start Server" at the bottom of the main menu. The IP address generated needs to be updated in realtime.m file. As shown in [Fig. 6.4].

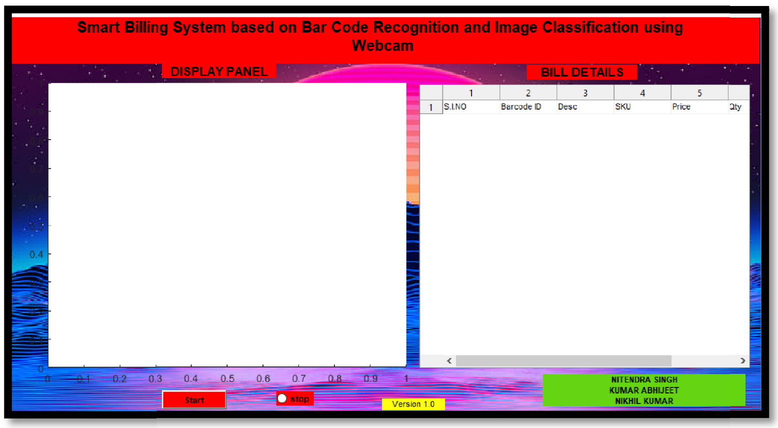


Fig. 6.4 Project GUI

4. Push start button for start scanning the product. Both barcode scanning and image classification can be done simultaneously and added up in the bill. As shown in [Fig. 6.5] and [Fig. 6.6].

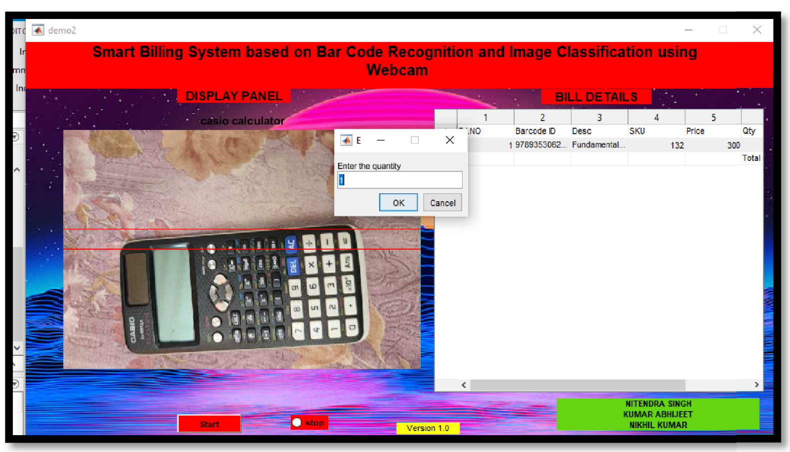


Fig. 6.5 Image classification result

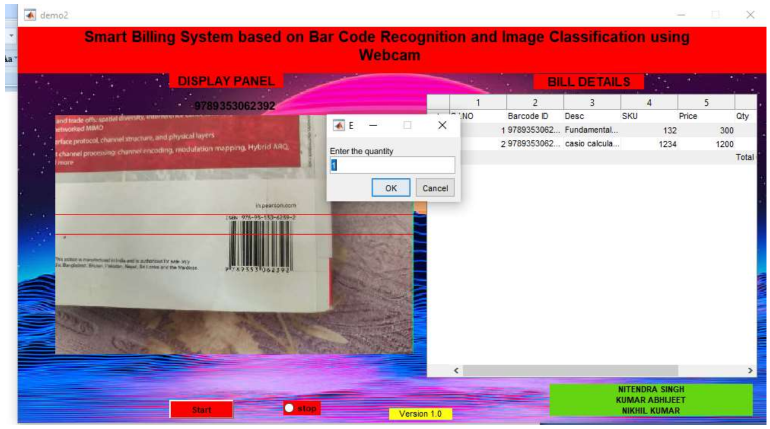
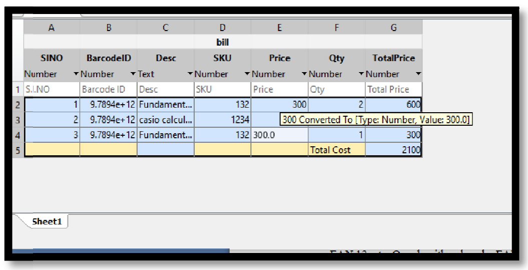


Fig. 6.6 Barcode scanning

5. After the whole number is decoded, it is compared with the database and when a match is found the corresponding information is displayed in another excel sheet. As more number of barcodes is scanned their information is added to the same sheet and a bill is created (by adding the price of each product) as shown in [Fig. 6.7].



	A	B	C	D	E	F	G
				bill			
	SINO	BarcodeID	Desc	SKU	Price	Qty	TotalPrice
	Number	Number	Text	Number	Number	Number	Number
1	S.I.NO	Barcode ID	Desc	SKU	Price	Qty	Total Price
2	1	9.7894e+12	Fundament...	132	300	2	600
3	2	9.7894e+12	casio calcul...	1234	300 Converted To [Type: Number, Value: 300.0]		
4	3	9.7894e+12	Fundament...	132	300.0	1	300
5						Total Cost	2100

Sheet1

Fig. 6.7 Final Bill

6.3 Results outcomes

This project provides an algorithm to develop a supermarket billing system, which recognizes the barcode and image using image processing tools and deep learning tools. MATLAB software is used to test the algorithm.

Proposed system provides a convenient method of extracting information from the barcode at a lesser cost as compared to typical electronic barcode scanners. This system can be used by businesses requiring accurate results at low cost.

6.4 Summary

This chapter includes steps for the execution of the project work and it also includes various discussions over accuracy of the project working.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 General Discussion

MATLAB merely can be used as steering for the prosperity of modern computational tools. So, we have created barcode detection and image classification from real time approach with. This system is convenient for selling products. The method can be used by anyone easily and perfectly. These finding corroborations have the greater accuracy of the proposed method than the others [21]. Fig.6.1 shows the result of the barcode detection performance of the proposed method. Fig.6.5 shows the result of image classification method. Therefore, we seem that the type requires special programming for transferring the input data to the application program. In point-of-sale management, barcode and image classification method can provide detailed, up-to-date information about the business, accelerate decisions and have more confidence. For example:

i. Fast selling items can be visualized rapidly automatically and perfectly.
ii. Slow-selling items can be recognized to prevent Inventory build-up.
iii. Historical data can be used to accurately predict seasonal fluctuations.
iv. Items can be regenerated on the shelf to reflect both
v. Sales prices and price increases.
vi. No chance makes a mistake any digit of barcodes.

This project proposed smooth barcode detection and image classification of real time images with MATLAB as well as image processing. It is expected that the system has a high efficiency as it can remove unwanted noise. This simulation can detect barcodes and image from any direction. Here, real time images of barcodes are used as input. Input images are EAN-13 and image of object itself. Although we have effectively detected EAN-13 and image with minimum noise, this increases performance. So, we seem that the system is working smoothly and perfectly. So, it is the best and different system than others.

The project will continue until the system is fully functional, along with an additional feature that can help this system in the future. In order to achieve greater efficiency for this project, the following proposal for future progress is used. For future system development, it is suggested that the system should also include a payment option through QR code. So that the user can collect payment immediately after billing.

We have presented an algorithm for decoding (localization and reading) barcodes through a webcam, using image processing techniques such as filtering, cropping, and contrast enhancement and edge detection. This localization technique is fast and accurate even for cluttered images. Barcodes having a flat surface and a width to height ratio between 2 to 3 are successfully decoded. We have implemented this algorithm in MATLAB R2020a and can work on any of the later versions of MATLAB. We have scanned multiple products by executing this algorithm in a loop, and a final bill of products is created in an excel sheet.

This is an efficient algorithm, which can be further improved by adding angle correction method and can be provided to small grocery shops that cannot afford costly barcode and image scanners. This saves a lot of time than manual calculation and also removes the possibility of human error. We have extensively tested our algorithm on challenging images and the accuracy is found to be above 75 percent.

7.2 Future Work

The focus will be to implement of the more object tracking and image processing, as it can avoid high investments in industrial control computers while maintaining improvement of computing and optimized system structure. The hardware implementation would make it possible to use the project in real time under practical conditions. Also payment thorough QR code scan be added. Further with certain deep learning tools customer behavior can be predicted so as to make business approach as per the customer needs.

REFERENCES

[1] Gallo, O.,Manduchi, R., "Reading 1D Barcodes with Mobile Phones Using Deformable Templates," Pattern Analysis and Machine Intelligence, IEEE Transactions, vol.33, no.9, pp.1834,1843, Sept. 2011 doi: 10.1109/TPAMI.2010.229

[2] Luping Fang; Chao Xie, "1-D Barcode Localization in Complex Background," Computational Intelligence and Software Engineering (CiSE), 2010 International Conference, pp.1,3, 10-12 Dec. 2010 doi: 10.1109/CISE.2010.5677072

[3] Xianyong Fang, FuLi Wu, Bin Luo, Haifeng Zhao and PengWang,"Automatic Recognition of Noisy Code-39 Barcode," Artificial Reality and Telexistence– Workshops, 2006. (ICAT '06), pp.79,82, Nov. 29 2006-Dec. 1 2006,doi: 10.1109/ICAT.2006.45H

[4] W. N. W. Shuhaimi, "Real Time Barcode Reader for Laboratory Attendance (Software Part), Bachelor Thesis, 2007.

[5] M. E. H. A. Aziz, "Real Time Barcode Reader for Laboratory Attendance (Hardware Part), Bachelor Thesis, 2007.

[6] Brain M., 2000 'How UPC Barcodes Work?', http://electronics.howstuffworks.com/upc.htm, last viewed 7th November 2008.

[7] Seideman T., 1993, 'Barcodes Sweep the World', Vol. 8 Issue 4, http://www.americanheritage.com/articles/magazine/it/1993/4/1993_4_56.shtml, last viewed 7th November 2008.

[8] "Barcode." Wikipedia, the Free Encyclopedia. Wikimedia Foundation, Inc. 8. Dec. 2011. Web. 9. Dec.2011.

[9] Kuroki, Tetsuo Satou, Takayuki Yoneoka, Noriaki Kayaniori, Ypichi Takagi and Tadaaki Kitamura "Barcode Recognition System Using Image Processing", IEEE international conference,2006.

[10] Ashutosh Singla, Lin Yuan lin, Ebrahimi, "Food/Non-food Image Classification and Food Categorization using Pre-Trained GoogLeNet Model", Proceeding MADiMa '16 Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management, Pages 3-11, 2016

[11] Zhongqing Yu. Application Research on Embedded Barcode Recognition System Based on Image Processing[D] Chinese Marine University, 2007.Web. 2 Dec. 2011.

[12] Pavlidis, T., Swartz, J. and Wang, Y.P., 1990. Fundamentals of bar code information theory. Computer, (4), pp.74-86.

[13] Palmer, Roger C. The bar code book: reading, printing, and specification of bar code symbols. Helmers Pub., 1989.

[14] Tekin, E. and Coughlan, J., 2020, May. A Bayesian algorithm for reading 1d barcodes. In 2020 Canadian Conference on Computer and Robot Vision (pp. 61-67). IEEE.

[15] Zhongqing Yu. Application Research on Embedded Barcode Recognition System Based on Image Processing[D] Chinese Marine University, 2007.Web. 2 Dec. 2011.

[16] Marek Dąbrowski, Tomasz Michalik Orange Polska, "How effective is Transfer Learning method for image classification," IEEE Transactions, vol.33, no.9, pp.1834,1843, Sept. 2011 doi: 10.1109/TPAMI.2010.229

[17] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62-66, Jan. 197

[18] Navyashridhar C S, Dr. Mohan K G, "A Low Cost Barcode Reader Health Application On An Android Mobile Device", International conference on computer science and engineering,28thApril-2013,Bengaluru,ISBN: 978-93-83060-05-465.

[19] Alex Krizhevsky, I. S. (2012). ImageNet Classification with Deep Convolutional Neural Networks. NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, (pp. 1097-1105). Lake Tahoe, Nevada.

[20] Palmer, Roger C. The bar code book: reading, printing, and specification of bar code symbols. Helmers Pub., 1989.

[21] Tekin, E. and Coughlan, J., 2020, May. A Bayesian algorithm for reading 1d barcodes. In 2020 Canadian Conference on Computer and Robot Vision (pp. 61-67). IEEE.

[22] Shams, R. and Sadeghi, P., 2007, April. Bar code recognition in highly distorted and low-resolution images. In 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07 (Vol. 1, pp. I-737). IEEE.