Ahsanullah University of Science and Technology

THESIS REPORT

THESIS I
CSE-4100

A Study on Image Processing to Facilitate Business System by Multiple Barcode Detection

Submitted by:

Tasnim Mashrur Mahee
ID: 15.01.04.013

Atiqul Islam Chowdhury
ID: 15.01.04.014

Mushfika Sharmin Rahaman
ID: 15.01.04.016

Rifat Ahamad
ID: 15.01.04.028

Supervised by:

Nazmus Sakib
Assistant Professor,
Department of Computer Science
and Engineering,
AUST.

November 27, 2018

Summary

The pillar of automatic identification is Barcode technology which is used comprehensively in real time applications with various types of codes. The different types of codes and applications sometimes faces special problems, so the improvement of the effectiveness must be done persistently. In addition, every sector of the world is now being high speed processing places and that's why something different is needed to make the traditional barcode system more faster and accurate. This paper's proposal mainly deals with multiple product's barcode to be detected simultaneously. The proposed algorithm which will have implementation for supershop billing system. The method will recognize the barcodes using image processing. Images will be taken using mobile camera sensor. It will detect 1D barcodes such as EAN-13 barcodes, Code-128 barcodes, 2D barcodes such as QR codes. Moreover it will be angle invariant, requires no user interaction and can be executed on available computers. This model helps consumers to minimize the time for the billing system in shopping. We have implemented this in Python IDE using OpenCV library.

Contents

List of Figures iii

List of Tables iv

1 Introduction 1

2 Literature Review 3
- 2.1 Barcode structure . . . 3
- 2.2 Types Of Barcode . . . 4
  - 2.2.1 1D Barcode font . . . 4
  - 2.2.2 2D Barcode font . . . 5

3 Proposal 9
- 3.1 Basic flowchart of our proposed model . . . 9
- 3.2 Detailed flowchart of our proposed model . . . 10

4 Implementation 13
- 4.1 Single 1D barcode detection and decoding . . . 14
- 4.2 Single 2D barcode detection and decoding . . . 14
- 4.3 Multiple 1D barcode detection and decoding . . . 14

5 Experimental Results 16
- 5.1 Detection process . . . 16
  - 5.1.1 Auto generated single 1D barcode . . . 16
  - 5.1.2 Auto generated multiple 1D and 2D barcode . . . 16
  - 5.1.3 Real time single 1D barcode . . . 17
  - 5.1.4 Real time multiple 1D barcode . . . 17
  - 5.1.5 Real time single 2D barcode . . . 18
- 5.2 Decoding process . . . 18
  - 5.2.1 Decoding single 1D barcode . . . 18
  - 5.2.2 Decoding single 2D barcode . . . 19
- 5.3 Dataset . . . 20
- 5.4 Detection Accuracy Checking Through Resolution . . . 21

6 Conclusion and Future Work 22
- 6.1 Conclusion . . . 22
- 6.2 Challenges . . . 22
- 6.3 Future Works . . . 22

Bibliography 24

List of Figures

2.1 Structure of a 1D barcode . . . 3
2.2 QR code recognition system overview (7) . . . 7

3.1 Basic flowchart of our proposal . . . 10
3.2 Detailed flowchart of Barcode Detection Method . . . 12

4.1 Determination of how far away the products can be positioned from the camera . . . 14

5.1 Single barcode detection. . . . 16
5.2 Single barcode detection from an image. . . . 16
5.3 Multiple barcode detection . . . 17
5.4 Single 1D barcode detection from mobile image . . . 17
5.5 Multiple 1D barcode detection from mobile snap . . . 18
5.6 Single 2D barcode detection from mobile image . . . 18
5.7 Detecting and decoding single 1D barcode from an image . . . 19
5.8 Detecting and decoding single 2D barcode from an image . . . 19
5.9 Detecting and decoding multiple 1D barcodes from an image . . . 20

List of Tables

5.1 Dataset for detecting and decoding barcodes 20
5.2 Arte database: Resolution vs Accuracy 21
5.3 Arte database: Resolution vs Time 21

Chapter 1

Introduction

As consumers, we see barcodes and barcode scanners used all the time: purchasing from any retail store, renting a car, attending major events, flying, and even going to the doctor. They're in our social media apps and on store windows. A barcode is “A machine-readable code in the form of numbers and a pattern of parallel lines of varying widths, printed on and identifying a product.” A bar code can best be described as an ”optical Morse code.” Series of black bars and white spaces of varying widths are printed on labels to uniquely identify items. The bar code labels are read with a scanner, which measures reflected light and interprets the code into numbers and letters that are passed on to a computer. Because there are many ways to arrange these bars and spaces, numerous symbologies are possible. But in truth a barcode is so much more. Barcode systems help businesses and organizations track products, prices, and stock levels for centralized management in a computer software system allowing for incredible increases in productivity and efficiency.

Barcode data-collection systems provide enormous benefits for just about any business. With a bar code data-collection solution, capturing data is faster and more accurate, costs are lower, mistakes are minimized, and managing inventory is much easier.

Now we are living in such time where everything, every process in each sector are being automated. Advantages commonly attributed to automation include higher production rates and increased productivity, more efficient use of materials, better product quality, improved safety, shorter workweeks for labour and reduced factory lead times. So, to make the maximum profit from a business it must be follow the trend which means it must be adopt automation and makes the highest and proper use of it. Automation requires much less or not at all human interaction. But traditionally in super shops used to see a lot of human interactions are made. People buy products. Then make one or more queue for scanning those products where some employees are appointed for scanning barcodes with a barcode scanner. It's a traditional view of a super shop. Here, we have to focus on two things. Firstly, a lot of time is wasted of both consumers and sellers for the scanning method. Moreover, the barcode scanner which is normally used is very much. If any people start an ordinary super shop but wants to put in barcode system in his shop the costing will be inappropriate for him. So these factors can be overcome if we can use something else instead of that barcode scanners and be able to scan multiple products simultaneously. In this way we can easily minimize the cost and can get rid of from long time spending at queue. That's how this idea can play a great role to maximize the profit in a business.

A barcode's density is determined by the ”X” dimension. Density refers to the amount of information that can be captured in the bar code in a particular space, usually a linear inch. While not intuitively obvious, high density bar codes have low numbers and low density bar codes have high numbers. This is because individual characters consist of some combination of bars and spaces that are each multiples of ”X”. When ”X” is small, the area required for each character is less than when ”X” is large; thus the bar code can hold more per linear inch and is said to be of higher density. Similarly, increasing the width of the narrowest element (”X”) increases the space required for each character and reduces the number of characters per inch. Because the resulting code is often quite large, very low density codes are often associated with applications such as warehousing that require reading bar codes from a significant distance (3 to 30 feet).

All bar codes have start/stop characters that allow the bar code to be read from both left to right and right to left. Unique characters placed at both the beginning and end of each bar code, the stop/start characters provide timing references, symbology identification, and direction of read information to the scanner. By convention, the unique character on the left of the bar code is considered the "start" and the character on the right of the bar code is considered the "stop". Immediately preceding the start character and following the stop character is an area of no markings called the quiet zone. Because there is no printing in this area, a scanning signal is not produced, thus the term "quiet." The quiet zone helps the scanner find the leading edge of the bar code so reading can begin. As a rule, the quiet zone should be ten times the "X" dimension or 1/4", whichever is greater.

A linear barcode typically holds any type of text information. In contrast, a 2D barcode is more complex and can include more information in the code: price, quantity, web address or image. A linear barcode scanner cannot read a 2D barcode, requiring the use of an image scanner for reading the information embedded in a 2D barcode. Mobile phones with cameras, like iPhones and Android phones, and many other devices can read 2D barcodes through their integrated cameras.

There are mainly two tasks for barcode readers: barcode locating and barcode decoding . Barcode localization methods have two objectives, speed and accuracy. For industrial environment, accuracy is crucial since undetected (missed) codes may lead to loss of profit. Processing speed is a secondary desired property of the detectors. On smartphones, the accuracy is not so critical, since the device interacts with the user and re-shoting is easilypossible, but a fast (and reasonably accurate) barcode detection is desirable. Various techniques are used to locate and decode barcodes from photographs, from the classical line scanning technique (1), through the widely studied morphological approaches (10), and recent studies using wavelets (12). Once an image is obtained, the first step is to localize the barcode, or to find its location within the image. Many methods to do this have been developed. Again, after the barcode is localized using one of the methods, it must be decoded in order to obtain the product's information. We have followed these processes by implementing this using Python. Firstly we have processed an algorithm to detect the barcode from the snap, then we have to decode it. And the main benefit in this process is, we can detect multiple barcodes at a time and can detect the barcode of cylindrical objects too. We have used Zamberletti algorithm using opencv and pyzbar library to detect and decode the barcodes from the snap.

If we think of shopping system, then we can see that there is a laser for detecting barcodes from a product. But if there are multiple products, then the system need more time to detect many products. And it is tough to detect multiple product barcodes one by one at a time. So, our method is helpful for solving this issue so that it takes less time in shopping system to detect all the barcodes. Our proposed mrthod is implemented by modified Zamberletti Algorithm using ZBar library.

A linear barcode typically holds any type of text information. In contrast, a 2D barcode is more complex and can include more information in the code: price, quantity, web address or image. A linear barcode scanner cannot read a 2D barcode, requiring the use of an image scanner for reading the information embedded in a 2D barcode. Mobile phones with cameras, like iPhones and Android phones, and many other devices can read 2D barcodes through their integrated cameras.

There are mainly two tasks for barcode readers: barcode locating and barcode decoding . Once an image is obtained, the first step is to localize the barcode, or to find its location withinthe image. Many methods to do this have been developed. Again, after the barcode is localized using one of the methods in the previous section, it must be decoded in order to obtain the product's information. We have followed these processes by implementing this using Python. Firstly we have processed an algorithm to detect the barcode from the snap, then we have to decode it. And the main benefit in this process is, we can detect multiple barcodes at a time and can detect the barcode of cylindrical objects too. We have used Zamberletti algorithm using opencv and pyzbar library to detect and decode the barcodes from the snap.

If we think of shopping system, then we can see that there is a laser for detecting barcodes from a product. But if there are multiple products, then the system need more time to detect many products. And it is tough to detect multiple product barcodes one by one at a time. So, our method is helpful for solving this issue so that it takes less time in shopping system to detect all the barcodes.

Chapter 2

Literature Review

Day by day we are going towards an era in which every process is being automated. This means much less or not at all human interaction. But traditionally super shop billing system requires a lot of human interaction. Again, the barcodes scanner's costing is much higher according to their longevity. And most importantly, the work of multiple barcode detection from an image was done on 1D barcodes and they are not angle invariant. We have analyzed some papers which are related to our topic. Some of them worked on 1D, some of them worked on 2D barcodes. But there are some additional terms which can make the barcode system more automated i.e. human interaction free and accelerate the whole system.

Let's take a look at a barcode in more detail:

2.1 Barcode structure

a. Margin: Spaces, normally white, where nothing printed are required at each end of the code and they should be 10 times that of a narrow bar.

b. Barcode symbol: The area composed of bars and spaces is known as the "bar code symbol".

c. Barcode: The symbol togethrer with the left and right margins make up what is called bar code.

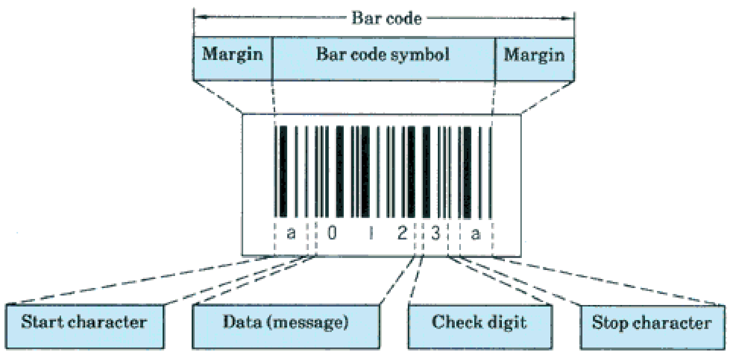


Figure 2.1: Structure of a 1D barcode

d. Start character: Indicates the start of the data. Depending on code system, start character varies.

e. Data(message): Area indicating the actual data.

f. Check digit: Confirms that there are no error in reading by means of one digit of compiled data.

g. Stop character: Indicates the end of the data.

2.2 Types Of Barcode

There are two types of barcodes: linear – or 1D, and 2D. The most visually recognizable, the UPC (Universal Product Code) is a linear 1D barcode made up of two parts: the barcode and the 12-digit UPC number. The first six numbers of the barcode is the manufacturer's identification number. The next five digits represent the item's number. The last number is called a check digit which enables the scanner to determine if the barcode was scanned correctly.

2.2.1 1D Barcode font

a. Uniform Product Code (UPC): "UPC" stands for Universal Product Code. UPC bar codes were originally created to help grocery stores speed up the checkout process and keep better track of inventory, but the system quickly spread to all other retail products because it was so successful. It is widely used in the United States, Canada, United Kingdom, Australia, New Zealand, in Europe and other countries for tracking trade items in stores. UPC (technically refers to UPC-A) consists of 12 numeric digits, that are uniquely assigned to each trade item. Along with the related EAN barcode, the UPC is the barcode mainly used for scanning of trade items at the point of sale.

1 23456 78910 4

b. Code 39 (Code 3 of 9): The Code 39 specification defines 43 characters, consisting of digits 0-9, the letters A-Z (upper case only), and the following symbols: space, minus (-), plus (+), period (.), dollar sign, slash (/), and percent. A special start/stop character is placed at the beginning and end of each barcode. The barcode may be of any length, although more than 25 characters really begins to push the bounds of practical physical width.An additional character (denoted '*') is used for both start and stop delimiters. Each character is composed of nine elements: five bars and four spaces. Three of the nine elements in each character are wide (binary value 1), and six elements are narrow (binary value 0). The width ratio between narrow and wide is not critical, and may be chosen between 1:2 and 1:3.

Wasp Barcode

c. PostNet: The PostNet barcode is used by the United States Postal Service to automatically sort mail. The PostNet code consists of evenly spaced bars of two different heights. Each character is represented by five bars, two tall and three short. The character set includes the digits 0 through 9. The code begins and ends with a tall bar ('frame bar'), and may contain a 5-digit ZIP code, a 9-digit ZIP+4 code, or an 11-digit Delivery Point Code. A Modulo 10 check digit ('correction character') is inserted after the ZIP code and before the ending frame bar.

d. Code 128: Code 128 is a high-density linear barcode symbology defined in ISO/IEC 15417:2007. It is used for alphanumeric or numeric-only barcodes. It can encode all 128 characters of ASCII.

123456789

e. Bookland: The Bookland EAN barcode is used internationally to identify books as well as video and audio cassettes and software. The unique number assigned to each item is the International Standard Book Number (ISBN).

f. Interleaved 2 of 5: Interleaved 2 of 5 (ITF) is a numeric only barcode used to encode pairs of numbers into a self-checking, high-density barcode format. In this symbology, every two digits are interleaved with each other to create a single symbol. If a number string containing an odd number of digits needs to be encoded, a leading zero must be added to produce an even number of digits in the Interleaved 2 of 5 barcode.

g. Codabar: Codabar is the barcode developed by Monarch Marking Systems in 1972. It is the barcode introduced at early stage following "2 of 5". It is widely used for applications that require serial numbers, such as management of blood banks, slips for door-to door delivery services and member cards. Codabar has 4 bars and 3 spaces (total 7 elements) with each narrow or wide width representing one character (letter). Codabar can represent characters including numbers (0 to 9), letters (A, B, C, D) and symbols (space, minus, plus, period, dollar sign, slash).

2.2.2 2D Barcode font

a. PDF417: Large amounts of text and data can be stored securely and inexpensively when using the PDF417 barcode symbology. The printed symbol consists of several linear rows of stacked codewords. Each codeword represents 1 of 929 possible values from one of three different clusters. A different cluster is chosen for each row, repeating after every three rows. Because the codewords in each cluster are unique, the scanner is able to determine what line each cluster is from.

b. Data Matrix: Data Matrix is a very efficient, two-dimensional (2D) barcode symbology that uses a small area of square modules with a unique perimeter pattern, which helps the barcode scanner determine cell locations and decode the symbol. Characters, numbers, text and actual bytes of data may be encoded, including Unicode characters and photos.

c. Maxicode: Maxicode is an international 2D (two-dimensional) barcode that is currently used by UPS on shipping labels for world-wide addressing and package sortation. MaxiCode symbols are fixed in size and are made up of offset rows of hexagonal modules arranged around a unique finder pattern. MaxiCode includes error correction, which enables the symbol to be decoded when it is slightly damaged.

d. QR Code: QR-Code is a two-dimensional (2D) barcode type similar to Data Matrix or Aztec, which is capable of encoding large amounts of data. QR means Quick Response, as the inventor intended the symbol to be quickly decoded. The data encoded in a QR-Code may include alphabetic characters, text, numbers, double characters and URLs. The symbology uses a small area of square modules with a unique perimeter pattern, which helps the barcode scanner determine cell locations to decode the symbol.

There was a method of extracting information from the barcode at a lesser cost compared to typical electronic barcode scanners (11). They had analyzed their diagram by following three levels, they are: level 0, level 1, level 2. They saved the product information in database. After that they take a snap with USB webcam and make ready the image for scanning by removing noise and eliminate unnecessary surrounding information. They perform cropping by observing the intensity of each pixel and extracting the rows of a barcode. By that time the contrast of the image improve to distinguish between bars by making the black bars one shade darker on the grayscale compared to white bars. Then the contrast-enhanced image is firstly binarized and edge detection is done successfully. After that the barcode number is decoded using the array of bar widths and then, after matching the product information, the product bill had been updated. EAN-13 barcode was used for this experiment.

After that, if we take a look, then we see that there was another experiment on barcodes, especially single and plural barcodes which were detected at a time and the accuracy was 0.7 seconds to detect the barcodes (9). The method was to extract candidate barcode skeletons to locate barcodes in an image. The method was capable of searching rotated barcodes in a high-resolution image. Firstly the image is preprocessed by skeleton extraction in the downscaled image plane and verifying barcode location in the original image plane. After the preprocess, cross-scanning is employed to scan the binary downscaled image in a first direction, and then the binary downscaled image is scanned in a second direction perpendicular to the first direction. In this way skeleton extraction is done via cross-scanning. Then a grading scheme is used to determine whether a detection window may contain portion of a barcode, the scheme is designed by exploiting the nature of barcodes as much as possible. After the wining detection windows are obtained, it approaches to scan barcodes in order to correctly extract gray-level values in the original image plane. It uses the Sobel operator to track the main direction. Finally it runs a retrospective tracking.

Another work was to detect QR codes from cylindrical object like bottles, cans (7). These works influences us to work on 1D and 2D barcodes at a time that means multiple barcode detection. Extraction of finder patterns is one of the major procedures for locating a QR code. In the preprocessing stage, source gray image is converted to binary image using the adaptive thresholding method (2) which is very robust and less computational complexity; Finder pattern candidates is located by roughly scanning and further confirmed by contour tracing (3) and corner detection. Three finder patterns located at the corners on the QR code have fixed ratio(1:1:3:1:1) of the widths of the black and white regions. While being scanned in arbitrary direction, these patterns are able to help quickly locating the QR code and speeding up the QR code reading. It obtain the candidates by roughly scanning in horizontal and vertical direction.

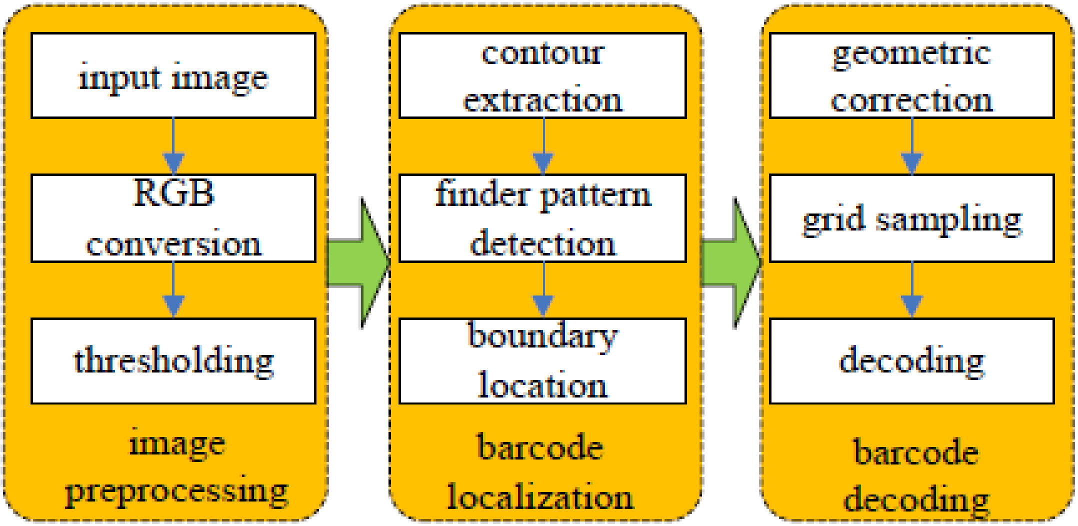


Figure 2.2: QR code recognition system overview (7)

After finding the three position detection patterns, a corner searching algorithm is run to obtain the four corner points as it is necessary to get the boundaries as a precondition to determine the fourth corner under the circumstances that other three corner points have been located previously. After obtaining boundary the curved QR code is transformed to the square plane through geometrical correction. Since QR codes might be perspective distorted because the camera can be held in a rotation or elevation angle.

If we give a look, then we can see that there was a paper on detecting barcodes through cross identification using mobile platform (15). In this paper approach has been made for fast and robust color barcode detection. A feature detector is proposed for finding crosses formed by differently colored cells. Next, candidate barcode regions are generated using a thresholding method and those regions containing more than a specific number of crosses are declared to be barcode regions. Comparisons with the base line Hough transform method for matrix code detection are performed. To find the crosses it uses a cross detector to search for reliable X-shaped and T-shaped crosses. When the detector is placed at or near the center of a cross, each pixel in the probes records a color value. Then it combines different cross detectors. The cross detector may still get false positive responses in identifying cross feature points. For this reason false crosses are filtered where the standard deviation of pixel values is fairly small. After performing its detector on the input image, candidate barcode regions are generated by Niblack's thresholding method, which is robust to varying light conditions and complex background. Finally, shape consistency tests are performed on detected barcode regions to further exclude false detections.

A paper proposed a method of identifying the QR and Aztec barcodes by using connected component labeling algorithm to compute the tag for each connected component, then search for the innermost connected component of the finder patterns in the tag image (8). The method can identify rotated or defocused barcode images. At first it label the connected components in the barcode image. Standard thresholding algorithms, such as the Otsu's algorithm, can be used for thresholding (5) . The CCL algorithm assigns every connected component a unique number, called the tag. This algorithm scan the image from left to right and from top to bottom. Since the tag value increases in the scan direction, it is understandable that the innermost component in the finder patters is a local maximum in the tag image. Then it is checked that whether the current pixel is in the innermost connected component of the finder pattern of QR code. That's how it works.

Another work has been done on making this system angle invariant and less or not at all user interactive (14). Nowadays mobile applications has been using to identify products from pictures and also that can make online comparisons with the existing similar products. This approaches mainly focus on decoding degraded barcodes and treat the underlying barcode detection task as a side problem that can be addressed using appropriate object detection methods. However, the majority of modern mobile devices do not meet the minimum working requirements of complex general purpose object detection algorithms and most of the efficient specifically designed barcode detection algorithms require user interaction to work properly. But in this work, they presented a novel method for barcode detection in camera captured images based on a supervised machine learning algorithm that identifies one dimensional barcodes in the two-dimensional Hough Transform space and remarkably it is angle invariant, requires no user interaction and can be executed on a modern mobile device. Firstly, in this process, the rotation angle of the barcode is determined with Hough space using Multi Layer Perceptron (MLP). Then, determine the set of all the segments of the image by applying the same technique of Matasyz (4). After that, two histograms are defined that describe the intensity profile of the rows and the columns. Finally, a smoothing filter is applied to each histogram to remove low value bins corresponding to isolated non-barcode segments and the bounding box of the barcode is determined at last.

Some people works on the segmentation of images with 1D barcode, but also analyze the operation of different methods for 2D barcode images as well (6).They tried to detect automatically, rapidly and accurately the barcode location by the help of extracted features. They also propose a novel algorithm that outperforms the other literatures in both accuracy and efficiency in detecting 1D codes. The algorithm is based on based on bottom-hat filtering and other morphological operations.

Another work proposed that it can be used to detect and decode multiple Two-Dimensional (2D) barcodes (i.e. Datamatrix) (13). Most of the techniques discussed in literature, work for single 2D barcode detection and decoding. The barcodes are not wrapped in any reflective material such as polyethylene foiling. The proposed work however, not only uses image processing techniques but also proposes an optical hardware setup to address the issues related to decoding of 2D barcodes under reflection and poor illumination conditions. The captured image contains multiple 2D barcodes and each barcode is located on a white rectangular region and first, they locate the white rectangular patterns. CogPMAlign Tool is therefore first trained and applied to the captured image to find these patterns. Taking each white pattern as the search region, a 2D barcode is detected and decoded within this area using Cog2DSymbol tool. The decoded 2D barcodes can then be stored or transferred to the other devices over a network using Ethernet/IP. That's how this process works.

Chapter 3

Proposal

Barcode detection system is very important for all kinds of shop, so that the products of the shop can be arranged in a better way. After reviewing all of these papers, we thought that there were many scopes to improve the method for barcode detection in many ways. We worked on multiple 1D barcode detection. We can also search for better filtering option without degrading the accuracy. We would put effort to detect barcode on complex background.

Our method is to detect multiple 1D and 2D barcodes from a snap. We have used EAN-13, Code 128 as 1D barcode and QR Code as 2D barcode. For the detection of the barcodes, we have followed two steps. They are:

- To find the position of a barcode
- To decode the barcode.

3.1 Basic flowchart of our proposed model

This method has improved the detection of 1D and 2D barcodes at a time which was not easy earlier. Fig.3.1 shows the flow diagram of our working process. Our algorithm works fluently with the help of some python library.

a. Preprocessing: After taking an image as input, we have to preprocess this just like resize the image, filtering the image. Initially image had the resolution of 3146X3146 px which was reduced to 528X704 px. The image is converted to grey scale image. Then converted to ZBAR compatible format to find image symbols.

b. Barcode decoding: In this stage, the pyzbar library decodes the barcode. In the preprocessing stage image is already in ZBAR compatible format. Then scanlines are used to iterate over the symbols to extract type, data and location information.

c. Barcode displaying: After decoding next step is displaying the symbols. A quad is formed outside the barcode while displaying 2D barcode image. If location is not a quad an outer boundary box is drawn.

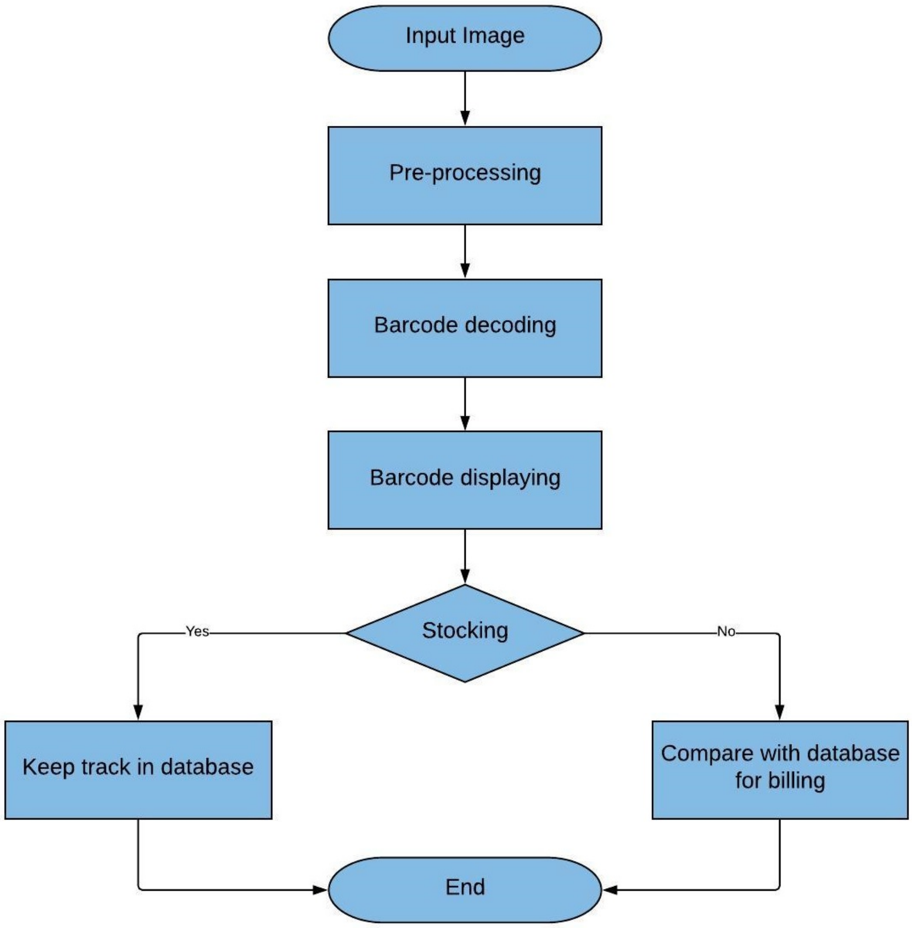


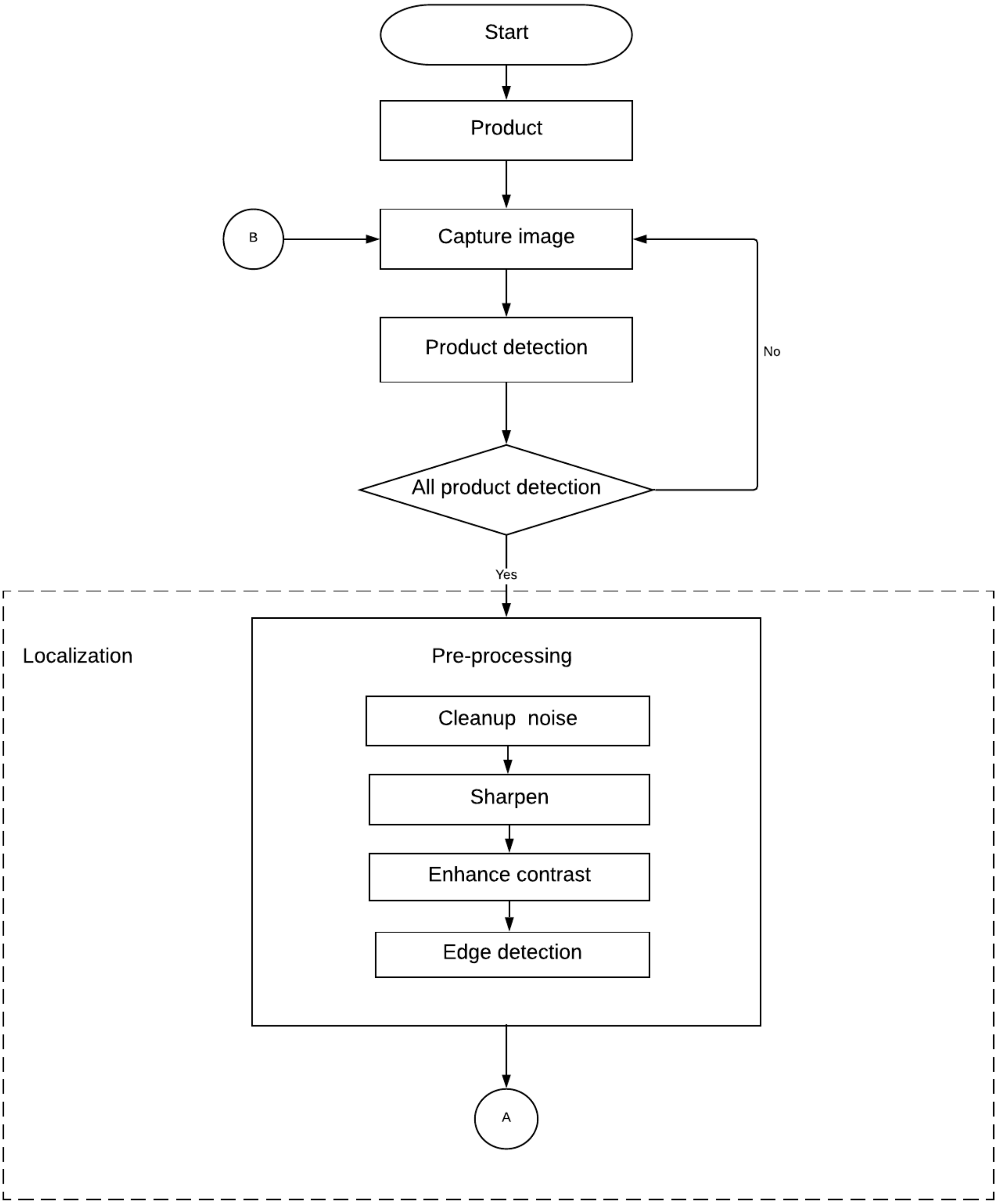
Figure 3.1: Basic flowchart of our proposal

3.2 Detailed flowchart of our proposed model

From the analysis of our flow diagram, we have planned to work on some portions which are given below:

- Single 1D barcode detection decoding
- Single 2D barcode detection decoding
- Multiple 1D barcode detection decoding
- Multiple 2D barcode detection decoding
- Multiple 1D 2D barcode detection decoding
- Angle invariant barcode detection decoding
- Barcode detection in complex background with higher accuracy
- Matching the decoded barcode values with the database

This method will be beneficial for any kind of supermarket to detect the barcodes. It can be used in super shop for the management of all products. Firstly, price of all products can be saved easily in the database by scanning the barcode while storing products. And most importantly while selling products, price detection according to the products will be very efficient as we can use multiple products at a time. Overall it doesn't take much time for detecting the products. Finally, human interaction won't be needed in future for the billing system.


Start
Product
B
Capture image
Product detection
No
All product detection
Yes
Localization
Pre-processing
Cleanup noise
Sharpen
Enhance contrast
Edge detection
A

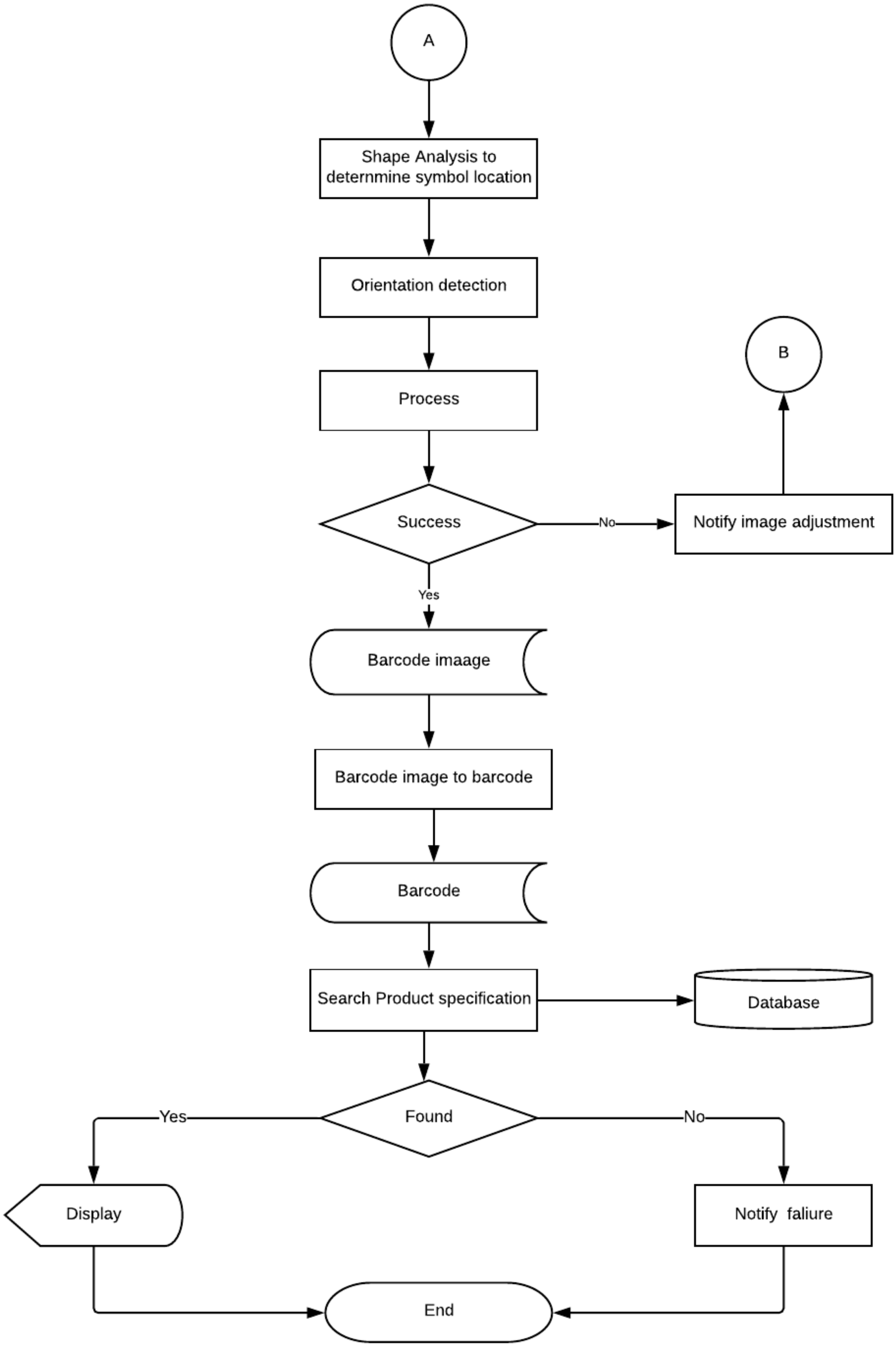


Figure 3.2: Detailed flowchart of Barcode Detection Method

Chapter 4

Implementation

Barcodes work through the combination of a symbology (the barcode) and a scanner that can read the symbols and convert them into useful information, often information about an item's origin, price, type, and location. The scanner reads the barcode and automatically enters the information stored in it into a system – often some type of database. Thinking the working process of barcodes, we have made a plan to detect the barcodes accurately and in an easy way. We have used Python language for experimenting the portions of our process.

There are many platforms that have been used for the detection of the barcodes. We mainly follow Python language to run the experiment. As python is a standard language, so it is useful to implement the process in a better way. The IDE that we have used for our implementation is Spyder IDE. Here another important part is OpenCV library, which is necessary for our process. OpenCV does not have any dedicated modules that can be used to read and decode barcodes and QR codes. However, what OpenCV can do is facilitate the process of reading barcodes and QR codes, including loading an image, grabbing a new frame from an image, and processing it. Once we have the image or frame we can then pass it to a dedicated Python barcode decoding library such as a Zbar. The ZBar library will then decode the barcode or QR code. OpenCV can come back in to perform any further processing and display the result.

For the detection and decoding of the barcodes, firstly we detected the barcode(s) from a snap and start the preprocessing. As we used ZBar Bar Code Reader library, so this library helps us to improve the detection process of the barcodes. ZBar Bar Code Reader library is an open source software suite for reading bar codes from various sources, such as video streams, image files and raw intensity sensors. It supports EAN-13/UPC-A, UPC-E, EAN-8, Code 128, Code 39, Codabar, Interleaved 2 of 5 and QR Code. Included with the library are basic applications for decoding captured bar code images and using a video device (eg, webcam) as a bar code scanner.

The decoding process is implemented by calling decode function that is a built in function in python. We have shown "Type" and "Value" of the barcode(s) in decoding process. The decoding process is important because in a shop, when we buy the products, then the product type and product value is important to match with the database for billing perfectly.

As barcodes are important for any kind of product, so the detection of the barcodes is also an important factor specially in shopping system. Till now, we have implemented the portions of our work which are:

- Single 1D barcode detection decoding
- Single 2D barcode detection decoding
- Multiple 1D barcode detection decoding

All the methods/works are done by modifying Zamberletti algorithm using ZBar library which is called PyZbar library in python. The detection process and decoding process shows accurate result for these types of barcodes. And for the decoding process, we have shown the "Type" and "Value" of the barcodes.

To detect barcodes, some factors must be noticed because the detection process depends on image quality and image enhancing issue. For the detection of the barcodes, the filtering, thresolding process are done by ZBar library as it is mainly designed for detecting the bars and portions of the barcodes. the detection process depends on the focus of the camera sensor. There is minimal focal length to capture a better image so that the barcodes from the image can easily be detected.

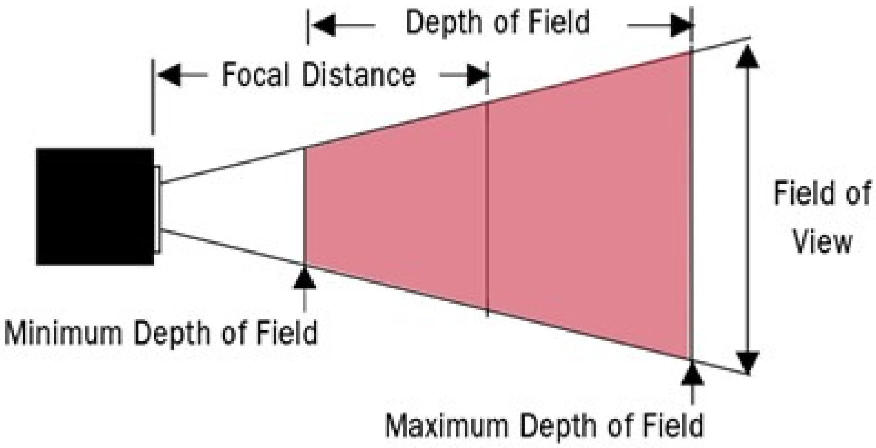


Figure 4.1: Determination of how far away the products can be positioned from the camera

The other important factor is the resolution, which is described earlier. The quality of an image depends on the resolution of the image and so the barcodes can easily be detected by it. Another important part is camera sensor which is necessary for capturing the image. There are many camera sensors, but for the business purpose the owner firstly needs a sensor which has low price but better quality. So the cost factor is an important part because there are various types of sensors. Many of them have high price, many of them have mid ranged price. So it also plays an important role to calculate the cost related to the system.

4.1 Single 1D barcode detection and decoding

There are many 1D barcodes. But among them, UPC-A, EAN-13 barcodes are mostly used. For the detection of single 1D barcode, firstly we have trained the dataset found from Applied Recognition Technology Laboratory. From the dataset, we have seen that there are many single 1D barcode images which have the resolution of 784 x 588. But sometimes the resolution matters most for an image, because mobile camera sensors have different resolutions. So it is necessary to check the barcodes in lower resolution. Basically, lower doesn't mean very poor quality. Minimum resolution of camera sensors is 800 x 600 that is used later for the detection and decoding process. By using the last resolution, we have got similar result comparing to the previous dataset. Thus, 1D barcode detection is done with better detecting accuracy.

4.2 Single 2D barcode detection and decoding

2D barcodes are rarely used for some products. Mainly QR codes are used as 2D barcodes. After implementing the detection of single 1D barcode, our next target was to detect single 2D barcode from an image. Here, the resolution of the image is similar to previous process that is 800 x 600. The format information and timing patterns are the most important factors to detect the QR codes. And if the resolution is good, then the QR codes can easily be detected. Here the length of the points are also plays an important role to detect the QR codes. If the length is equal or less than 4 that indicates the format information, then we can say that it is a QR code and it can easily be detected.

4.3 Multiple 1D barcode detection and decoding

Single 1D barcode detection may seem easy, because the barcode can easily be focused while detecting it from an image. But it is quite hard to detect multiple 1D barcode detection from an image. Because all the barcodes must be focused perfectly. For this implementation, the resolution plays an important role. If the resolution is about 800 x 600, then the multiple barcodes can't be easily detected. Because the machine can't read the barcodes perfectly if the resolution is low and the image quality is poor. So if the image resolution is about 1200 x 800, then it is quite easy to detect multiple barcodes, and we got the result perfectly by applying this.

Finally, we can say that the portions that we have done in our implementation, work perfectly for detection and decoding of any barcodes. Basically, 2D barcodes or QR codes are rarely used, so we have used the images from google for detecting and decoding 2D barcodes.

Chapter 5

Experimental Results

Our experiment has shown us the result of single and multiple barcodes. The barcodes are detected and decoded taking a less time. The analysis of barcode is done by ZBar library of python.

5.1 Detection process

5.1.1 Auto generated single 1D barcode

First of all, we experimented the algorithm with single 1D barcode and got the positive result of detection shown in figure 5.1.

Figure 5.1: Single barcode detection.

Then we had tried to detect single barcode from an image. And the barcode was detected successfully. Figure 5.2 illustrates the result.

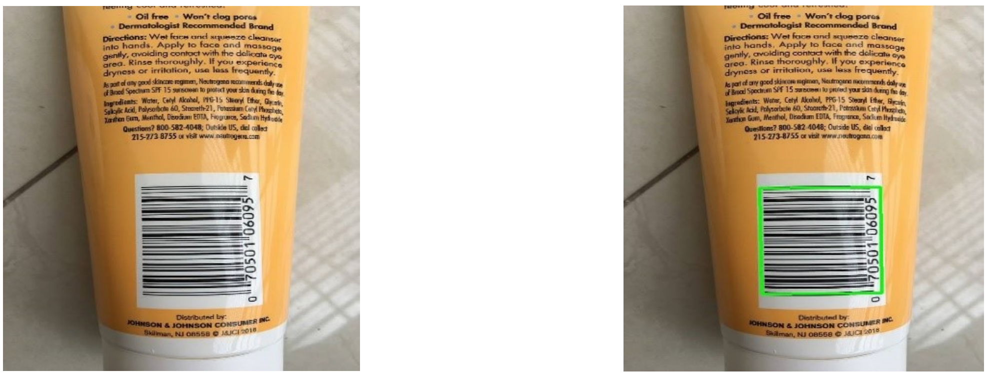


Figure 5.2: Single barcode detection from an image.

5.1.2 Auto generated multiple 1D and 2D barcode

Our next target was to detect multiple barcodes at a time. The background was white and there were multiple barcodes for the experiment. From Fig. 5.3, we see that there are multiple barcodes in one frame. Here, the data matrix is not barcode, so it can't be detected by zbar library.

Figure 5.3: Multiple barcode detection

Barcode detection part worked successfully on single 1D, single 2D and multiple 1D and 2D auto generated barcodes. It took less time to detect all the barcodes in an image.

5.1.3 Real time single 1D barcode

After doing the process, our challenge was to decode them from an image. Our proposal is to detect and decodes the barcode from a snap captured from mobile. So then we tried our detection process from a mobile image. Figure 5.4 illustrates the result.

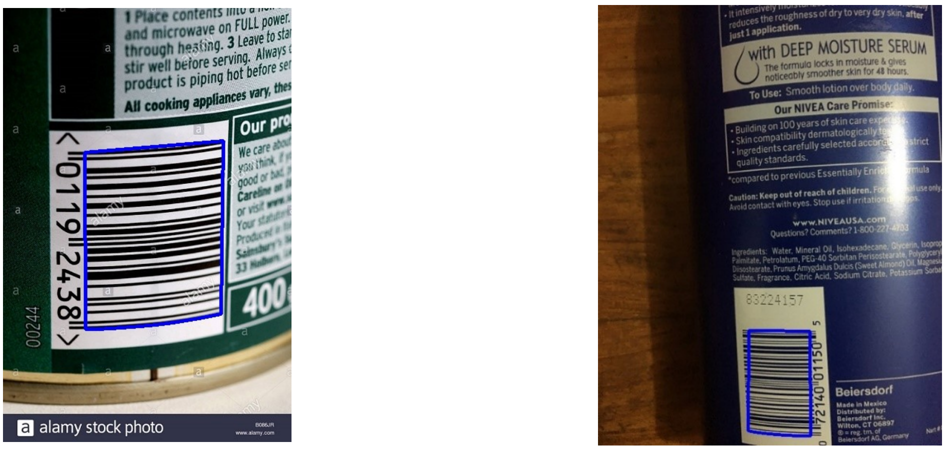


Figure 5.4: Single 1D barcode detection from mobile image

5.1.4 Real time multiple 1D barcode

Implementing single 1D barcode detection is quite easy because there is only one product to detect the barcode from this. But it is tough to detect multiple 1D barcodes because resolution plays an important role to capture the image of the products. Figure 5.5 shows some products to detect 1D barcodes from them.

Figure 5.5: Multiple 1D barcode detection from mobile snap

5.1.5 Real time single 2D barcode

After implementing the process for detecting 1D barcodes, we started to detect the 2D barcodes using the ZBar library. QR codes are mainly used as 2D codes. The detection of 2d codes are shown in figure 5.6.

Figure 5.6: Single 2D barcode detection from mobile image

5.2 Decoding process

5.2.1 Decoding single 1D barcode

In traditional scanning system, Barcode scanners read 1D barcodes horizontally. 1D laser barcode scanners are the most commonly used scanners, and typically come in a “gun” model. These scanners do not need to be in direct contact with the 1D barcode to work properly, but typically need to be within a range of 4 to 24 inches to scan.

1D barcodes are dependent on database connectivity to be meaningful. If you scan a UPC code, for instance, the characters in the barcode have to relate to an item in a pricing database to be useful. These barcode systems are a necessity for large retailers, and can help increase inventory accuracy and save time.

Barcode detection process was done by zbar library. But this library has another feature to decode the barcode at the same time. For decoding the barcodes, we have showed the type and data of the barcodes that are shown in Fig. 5.7.

Linear or 1D barcodes, like the UPC code commonly found on consumer goods, use a series of variable-width lines and spaces to encode data — what most people probably think of when they hear

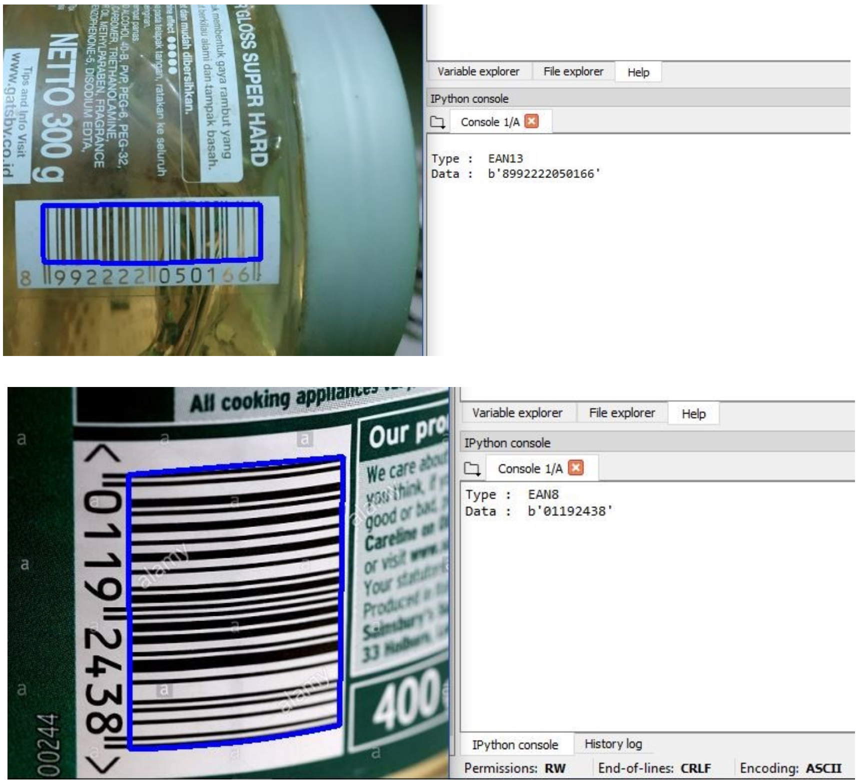


Figure 5.7: Detecting and decoding single 1D barcode from an image

"barcode." Linear barcodes hold just a few dozen characters, and generally get physically longer as more data is added. Because of this, users typically limit their barcodes to 8-15 characters.

5.2.2 Decoding single 2D barcode

In addition to holding more information, 2D barcodes can be very small, which makes them useful for marking objects that would otherwise be impractical for 1D barcode labels. With laser etching and other permanent marking technologies, 2D barcodes have been used to track everything from delicate electronic printed circuit boards to surgical instruments.

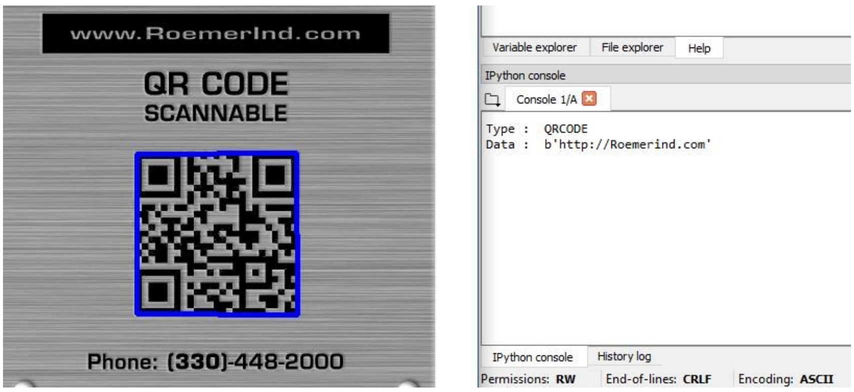


Figure 5.8: Detecting and decoding single 2D barcode from an image

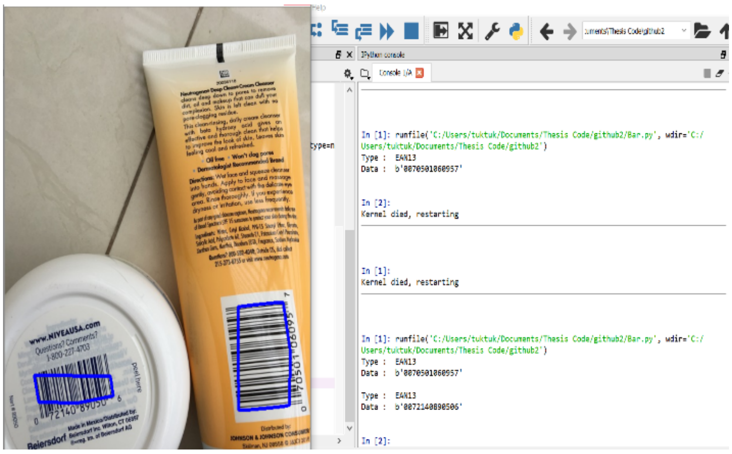


Figure 5.9: Detecting and decoding multiple 1D barcodes from an image

Last of all, we can ensure that our method will work for detecting and decoding any kind of barcodes from a snap with proper resolution. The simple and complex background doesn't matter to detect and decode the barcodes.

5.3 Dataset

Dataset for detecting and decoding barcodes:

Table 5.1: Dataset for detecting and decoding barcodes

Product	Product type	Product Shape	Barcode Type
Cosmetics	Facewash	Quadruple	EAN-13(1D)
	Facewash	Cylindrical	EAN-13(1D)
	Lipstick	Cylindrical	EAN-13 (1D)
	Lipstick	Rectangular	EAN-13 (1D)
	Lotion	Rectangular	EAN-13 (1D)
	Lotion	Curved rectangular	EAN-8 (1D)
	Cream	Round	EAN-13 (1D)
	Cream	Curved Quadruple	EAN-13 (1D)
	Oil	Cylindrical	UPC-A (1D)
	Oil	Round	EAN-13 (1D)
Food	Spices	Curved Rectangle	EAN-13 (1D)
	Can	Cylindrical	EAN-13 (1D)
	Chips	Square	EAN-8 (1D)
	Chocolate	Rectangle	EAN-13 (1D
Stationary	Book	Square	Code-39 (1D)
		Rectangular	Code-39 (1D)
	Marker	Cylindrical	UPC-A (1D)
	Stapler	Rectangular	EAN-13 (1D)
Beverage	Can	Cylindrical	EAN-13 (1D)
	Bottle	Rectangular curved	EAN-13 (1D)
	Packet	Cubic	EAN-13 (1D)
Electrical Accessories	Mobile packaging	Rectangle	EAN-13 (1D)
	Speaker packaging	Square	EAN-13 (1D)
	Headset packaging	Square	EAN-13 (1D)

5.4 Detection Accuracy Checking Through Resolution

Accuracy checking:

Table 5.2: Arte database: Resolution vs Accuracy

Resulation	Accuracy
1600X1200	89.90 percent
784X588	71.60 percent

Checking time:

Table 5.3: Arte database: Resolution vs Time

Resulation	Time
1600X1200	4mins 48 seconds
784X588	2 mins 10 seconds

We worked with arte-lab dataset at first for detecting and decoding 1D barcodes. It took 4 minutes 48 seconds for the datasets to output the results. Then we changed the resolution in order to check for the accuracy in lower resolution. It took 2 minutes 10 seconds to decode the whole dataset but the accuracy decreased to 71.60 percent. Detection was possible in the low light condition of the image. In cases of barcodes located in different angles it failed to decode the barcodes. In some barcodes in complex background ZBar library couldn't perform the detection and decode operation.

After resizing the resolution of the image, we have got 71.60 percent accuracy which takes above 2 minutes. So we have thought to use our fabricated dataset to improve the detection accuracy within less time. So we have fixed the resolution and it is about 800 x 600. Then we have analyzed the time and it takes 47 seconds to detect the barcodes and it detects 90.1 percent accurately.

Arte-lab whole image dataset was taken on normal mobile phone. Now the smart phone's sensors are much superior in the quality so created our own dataset just to test the ZBar library's performance. Our dataset was much smaller than the arte-dataset. I showed much better result.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Barcode is now available for every product. To detect barcodes, if one totally sums up, he will see a huge amount of working hours is used. If in barcode detection process of a superstore, multiple products can be included at a time and with great efficiency and higher accuracy, then it will save huge time. It will be beneficial for a shop if they take less time for their product management and billing system. Thus the productivity will increase. Again, making a system less user interactive or no user interactive at all makes the whole process more functional. Our study is to detect multiple barcodes through camera sensor, which will be helpful for the supermarkets and also it will be very cheap process in according to traditional system. Many literatures have worked on similar type of topics but they were very much specific. Some worked on single 1D barcode, some work on detecting multiple 1D barcode. Some of the people worked on single 2D barcode, some on multiple 2D barcode. Further, some people also work on detecting angle invariant multiple 1D barcode. But, we'll try to implement all of those issues in one platform and obviously through mobile camera sensor. Hope, it will be much lucrative in business sectors.

6.2 Challenges

Though we get success to detect single 1D, single 2D and multiple 1D barcode, but this method always didn't provide correct result. Some factors such as noise, reflection, contrast, unfocused image etc. cause some problems while detecting the barcode. For single 1D and single 2D barcode most of the time we were getting accurate result but sometimes because of lack of lighting, reflection, blur image were causing failure in detection. In case of multiple 1D barcode detection accuracy was quite low because of same reasons mentioned before. But while detecting single 1D and 2D, we were able to detect the barcode from low resolution images and that was helpful for shorten the time of detection. But for the detection of multiple 1D barcode, it always requires high resolution images and sometimes in spite of having high resolution, some barcodes remain undetected because of having tiny or uncommon shapes. It happens in both our dataset and Arte-lab-dataset. Complex background also causes difficulties in detection. Above all, one of the main difficulties we face while detecting angle invariant barcodes. It can only detect images having angle variance of ± 5 .

6.3 Future Works

We have already worked on single 1D, single 2D and multiple 1D barcode detection and decoding. Nevertheless, we have an issue of making it angle invariant and also have distance issue. Again one of the main tasks is, to detect multiple 2D barcodes, also detecting 1D and 2D barcode contemporarily are also remain undone. So the main tasks that will be done in future are given below:

- The most important task to be done next to detect and decode barcodes from images taken in different camera point of views.
- Overcome obstacles of poor resolution, scene illumination, image noise while decoding.
- Fixing highest number of barcodes detected as well as resolution benchmark.
- 1D 2D barcode detection at the same time.
- Detecting and decoding barcodes in complex background.

- Integrating all the process and finally resulting in totally runnable system.

Another way which we think will might be more efficient for detection is to use object detection process. For this purpose, we have to split our dataset into train-test and tested data will be used for measuring accuracy. In this way, the system will be able to recognize each object and it will treat each object a separate image. So, whatever great accuracy we were able to achieve for single 1D barcode detection that will also be possible to gain from this aspect. Because each product will act as a single image for the system.That's how the whole process will be fully completed. The enhancement of image localization can be further conducted to accurately detect the barcode images of products.

Bibliography

[1] Robert Adelmann, Marc Langheinrich, and Christian Floerkemeier. Toolkit for bar code recognition and resolving on camera phones-jump starting the internet of things. *GI Jahrestagung (2)*, 94:366–373, 2006.

[2] Derek Bradley and Gerhard Roth. Adaptive thresholding using the integral image. *Journal of graphics tools*, 12(2):13–21, 2007.

[3] Fu Chang and Chun-Jen Chen. A component-labeling algorithm using contour tracing technique. In *null*, page 741. IEEE, 2003.

[4] C Galamhos, Jose Matas, and Josef Kittler. Progressive probabilistic hough transform for line detection. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1, pages 554–560. IEEE, 1999.

[5] Rafael C Gonzalez, Richard E Woods, et al. Digital image processing, 2002.

[6] Melinda Katona and László G Nyúl. A novel method for accurate and efficient barcode detection with morphological operations. In *Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on*, pages 307–314. IEEE, 2012.

[7] Xiaochao Li, Zhifeng Shi, Donghui Guo, and Shan He. Reconstruct argorithm of 2d barcode for reading the qr code on cylindrical surface. In *Anti-Counterfeiting, Security and Identification (ASID), 2013 IEEE International Conference on*, pages 1–5. IEEE, 2013.

[8] Syue-Cin Lin and Peng-Hua Wang. Design of a barcode identification system. In *Consumer Electronics-Taiwan (ICCE-TW), 2014 IEEE International Conference on*, pages 237–238. IEEE, 2014.

[9] Chun-Shun Tseng, Kai-Tse Wang, Ming-Chang Wu, Nai-Yuan Cheng, and Jung-Hua Wang. Retrospective tracking for barcode reading. In *Industrial Informatics (INDIN), 2010 8th IEEE International Conference on*, pages 114–119. IEEE, 2010.

[10] Timothy R Tuinstra. *Reading barcodes from digital imagery.* PhD thesis, PhD thesis, Cedarville University, 2006.

[11] Sampada S Upasani, Adarsh N Khandate, Ankita U Nikhare, Rupali A Mange, and RV Tornekar. Robust algorithm for developing barcode recognition system using web-cam.

[12] Steffen Wachenfeld, Sebastian Terlunen, and Xiaoyi Jiang. Robust recognition of 1-d barcodes using camera phones. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[13] I Zafar, U Zakir, and EA Edirisinghe. Real time multiple two dimensional barcode reader. In *Industrial Electronics and Applications (ICIEA), 2010 the 5th IEEE Conference on*, pages 427–432, 2010.

[14] Alessandro Zamberletti, Ignazio Gallo, and Simone Albertini. Robust angle invariant 1d barcode detection. In *Pattern Recognition (ACPR), 2013 2nd IAPR Asian Conference on*, pages 160–164. IEEE, 2013.

[15] Yu Zhang and Tong Lu. A fast color barcode detection method through cross identification on mobile platforms. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 416–420. IEEE, 2015.