An Efficient Bar/QR Code Recognition System for Consumer Service Applications

Arju Aman, Aryan Singh, Ayush Raj and Sandeep Raj
Department of Electronics and Communication Engineering
Indian Institute of Information Technology Bhagalpur
Bhagalpur, 813210, India.
Email: {arju.ece.1709, aryan.ece.1710, ayush.ece.1713, sraj.ece}@iiitbh.ac.in

***Abstract*—In recent years, consumer electronics for providing better customer service has sought a significant growth. For production, storage and supply of goods to consumers, it is essential to have correct information, recognize and store the information efficiently in computers. Therefore, it is essential to have an efficient and handy bar code recognition system. This paper presents an efficient method for bar code and QR code recognition together. The method automatically detects the Bar code QR code and displays the complete information of the product. The method is developed in python environment using OpenCV library. However, OpenCV does not have any dedicated modules that can be used to read and decode Bar codes and QR codes. The database is developed at the authors' Institute where bar codes and QR codes are separately assigned to more than 100 items such as books, sofa, tables and chairs. The image of the bar code or QR code is captured in real-time and further processed using the proposed method. The code is being decoded, compared with the Data frame of the stored product and finally, displays the result i.e, the complete information about the product. The execution time of the proposed method is 0.25 seconds. The proposed method can be further prototyped on micro-controllers to develop an efficient bar code recognition system.**

***Index Terms*—Consumer service, Bar code and QR code Detection, Computer Vision, Image Processing, Python.**

I. Introduction

In recent times, there are enormous volume of goods are being produced and utilized on daily basis by the consumers. To have an effective control over the production, and distribution of goods, it is essential that each individual item is provided with an appropriate information or data to identify itself in a computer [1]–[3]. In order to serve the purpose, the Bar codes and QR Codes are widely used technique to convert each individual goods identity into a datum. Bar codes have taken over the world in making systematic order of goods/items [1]–[3]. If a product is packed to deliver in market, it has a Bar code which is used for all kinds of inventory/stocktaking work. The standard Bar code technology has significantly improved accuracy which leads to better management of inventory with low-cost implementation [1]–[6]. One can simply scan bar code and get all the relevant information that are stored in the database linked to it. 2D Bar code has received wide scope of consideration since the day it showed up for its preferences of huge stockpiling limit, high secrecy, solid protection from harm with several other features as well [5]–[14]. It became a significant research point in the field of pattern recognition and image processing.

Many researchers have investigated the domain of Bar code and QR code detection. Several methods have been reported in the literature. In 1932 Wallace Flint proposed a similar punch-card framework which was costly and lumbering like one produced for the 1890 U.S. Statistics. In 1952 Silver and Woodland form the main Bar code Reader for which the patent was granted. In 1967 Association of American Railroads starts utilizing Bar codes to ID railroad vehicles. In 1969 Computer Identics Corporation introduces the principal genuine Bar code frameworks at General Trading Company and General Motors offices. In 1970 NAFC (National Association of Food Chains) builds up the Ad-Hoc Committee for U.S. General stores on a Uniform Grocery-Product Code to set scanner tag advancement rules. In 1972 RCA starts a 18-month trial of a bull's-eye standardized tag framework in a store in Cincinnati. In 1973 UPC was introduced, stage from which Bar codes took off. In 1974 pack of Wrigley's biting gum was the main retail item sold utilizing a Bar code scanner at a Marsh store in Troy, Ohio. The abroad research on 2D Bar code innovation started in the late 1980s and they have built up an assortment of code frameworks. The home examination started in 1993. Based on processing some outside pertinent materials, two 2D Bar code identification national norms, FPD417(GB/T 17172-1997) and Quick Response Code(GB/T 18284-2000). In 1984, Bar code scanner were present in 33% of grocery stores. In 1994 Toyota subsidiary, Denso Wave, created QR codes to assist in more swift vehicles and part trackings. According to a report around 2004 80% to 90% of the top 500 companies in the United States were using Bar codes. Quick Response Code (QR Code) discharged by Japanese Denso Company, became ISO all inclusive standard in 2000. Not withstanding the highlights, for example, enormous data limit, high dependability, the capacity of communicating characters and picture data and solid security favorable circumstances, it likewise has the accompanying fundamental highlights, fast perusing, mistake revision ability [12]–[14].

This paper presents a new method for Bar code and QR code recognition. The method involves the graying operation on images acquired, filtering, binarization, data recognition and decoding of the code. The proposed method is developed using

978-1-7281-8259-9/20/$31.00 ©2020 IEEE

the python environment and applied on the images captured from the different items. The results obtained justifies that the method is efficient in recognition of QR codes and bar codes.

The later portion of the paper is summarized as: a brief description on the bar and QR code is discussed in section II while section III discusses the proposed methodology. Section IV presents the results and discussion of the experiments performed while section V concludes the paper.

II. Bar codes and QR-codes

This section presents a brief description of the Bar codes and QR codes along with their different types used to perform the experiments.

A. Bar code

A Bar code comprises of equal bars and spaces of various width and dividing between them. The symbol has coding region and function graphic which includes looking through example, separator, situating design and adjusting design [13], [14]. The function graphic can't be utilized for data encoding. Bars and spaces constitute a barcode, and is a machine-meaningful portrayal of numerals and characters.

The numbers below the Bar code act as an ID to an item containing all the important information. They are categorized on basis of linearity. The different types of linear bar codes (1D) are Code 25, EAN-2, Code 11, EAN-5, EAN-8, Codabar etc. An example of linear barcode is shown in Fig. 1.

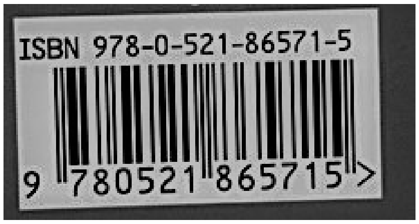


Fig. 1. Linear barcode

Another class of bar code is the Matrix Bar code (2D Bar code). The examples of the matrix bar codes are Dot Code, Aztec code, Dot code. etc. A typical example of 2D barcode is shown in Fig. 2.

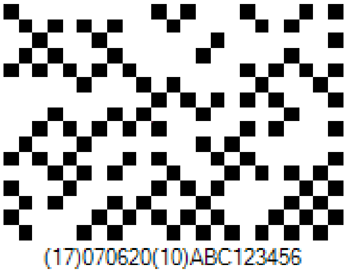


Fig. 2. Example of a 2D Dot code barcode.

In addition the most commonly used bar codes widely used in the present scenario are EAN (JAN), Interleaved 2 of 5, Codabar (NW-7), CODE39, CODE128, GS1 Databar (RSS).

B. QR-Code

QR stands for Quick Response (mobile phone can rapidy detect them with their cameras). It is used to take a fragment of data from temporary media devices and use it in our cellphones. It is an encoded bit of information. The information that we derive from the QR code can be alphanumeric, twofold, numeric or Kanji. The incentive in the back of why they may be extra useful than the Bar code is they can keep (and punctiliously gift) notably more meaningful information, including url joins, geo directions, etc. QR codes are utilized for various advertising materials nowadays and etc.

Fig. 3. Example of a QR Code.

III. Proposed Method

This study presents a method to detect and process the bar code and QR code respectively. The method consists of five stages. They are image acquisition, graying, filtering, binarization, data recognition, and decoding. The dataflow of the proposed method is depicted in Fig. 4.

A. Database

The experiments are performed on the images taken from the camera. The images are captured from the different items available including books etc. A database of the different bar code and QR code images is created. The corresponding information of the products are stored in CSV (Comma Separated Values) file type, which allows to add barcodes and QR code in quick manner. The column headers are used to separate the information to create a CSV file. These files are ordered in one format. In the first column, a value is assigned to each of the QR code/Bar code for validation. The second column represents the simple text, structured text, HTML or web content that will be displayed or highlighted after the

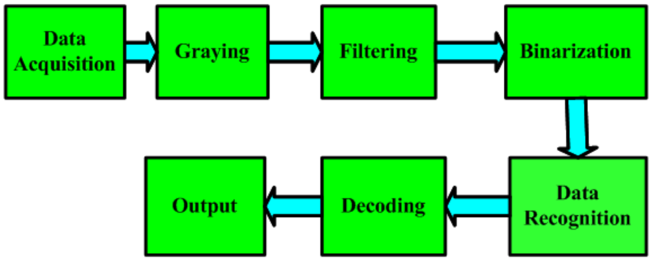


Fig. 4. Block diagram of proposed method

scan. In the third column, a brief description of the product is presented. A total of 100 items has been assigned to a bar/QR code on which the proposed method is implemented for performing the experiments.

B. Adaptive Filtering

Filtering is the key step for enhancing the images. Filtering is a technique for enhancing an image. One can filter photo to emphasis sure features or put off other features. In the domain of Image processing, filtering operations result in smoothing, sharpening, and side enhancement. The value of any given pixel inside the output photo is decided by means of making use of some set of rules to the values of the pixels inside the neighborhood of the corresponding input pixel. Generally, the image captured has noise, for instance, undesirable highlights that makes the image tough to look. This step removes the noise from picture. There are different kinds of noise such as Gaussian noise, salt and pepper noise, and etc. Further, this step improves the SNR of an image.

$$\hat{f}(x,y) = \frac{\sum\limits_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum\limits_{(s,t)\in S_{xy}} g(s,t)^{Q}} \qquad (1)$$

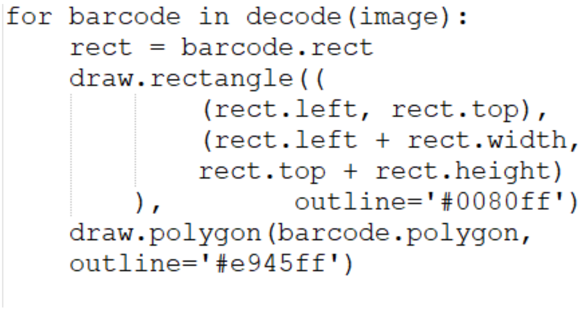
The contraharmonic filter is best for removing salt-and-pepper noise [15] , which is implemented using eq. (1). It is a noise of the form of black and white dotted pixels on images and generally finds due to abrupt change in the images. Here, S_{xy} is the coordinate set, when we consider rectangular window on image, and the center is at (x,y). g(x, y) is the distorted image. Q is the order of the filter. The pepper noise is removed for positive values of Q, and the salt noise is removed for negative values of Q. Both the removal of salt and pepper noises can't be done simultaneously.

C. Graying

In the graying step, estimation of every pixel is a solitary example to get a measure of light i.e., it conveys the intensity information. Grayscale pictures are just like highly contrasting or in dark monochrome. They are made of different and unique shades of gray. Where the intensity is the strongest, there is white color, and where the intensity is least, it is black color.

D. Binarization

The next step is the binarization which aims that pixels having a force over the limit turns white and the others dark. The even sweep is done first, wherein the edge pix. To hit upon the rims, at each pixel first derivative is used, so as to get the edges. The logic at the back of this is that the factor wherein a facet exists, there's an abrupt depth change, which causes a spike in the first derivative's cost, therefore making that pixel an 'edge pixel'.Hysteresis threshold is an upgraded version of normal threshold, it makes use of two thresholds rather than one. The motive for this is, if the brink is excessively high, we may additionally leave out a few real edges (proper negatives) and if the edge is excessively low, a ton of focuses named edges that absolutely aren't edges (bogus positives) is obtained. One threshold cost is ready high, and one is set low. All points which might be over the 'excessive threshold value' are diagnosed as edges, then all points which can be above the low threshold fee however underneath the high threshold fee are evaluated; the points that are close to, or are buddies of, points which have been recognized as edges, also are identified as edges and the relaxation is discarded.



```
for barcode in decode(image):
    rect = barcode.rect
    draw.rectangle((
            (rect.left, rect.top),
            (rect.left + rect.width,
            rect.top + rect.height)
        ),        outline='#0080ff')
    draw.polygon(barcode.polygon,
    outline='#e945ff')
```

Fig. 5. Code Snippet explaining how pyzbar decodes the barcode

E. Data recognition

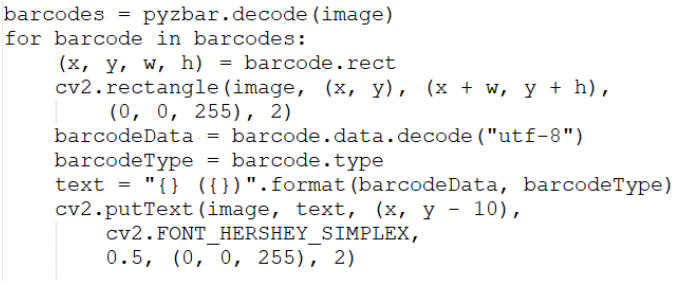
Initially the images are captured from the camera to create a database. It detects 2D Bar code. The individual relies upon the Scanner tag areas. Sobel kernels are used in order to find gradient in the abssissa and ordinate. Next, pixels with high gradient value are selected. Next, morphological expansion is done so as to combine close by yet not significantly associated items to have the option to make an area. Morphological erosion is carried out than to get rid of skinny objects from the photo and dispose of now not essential segments which had been mixed by using the dilation. The very last step is a solidity test. Fake given items are sided and best Bar code areas stay after this step. It is introduced in two significant stages. To begin with, the degree of clamor is diminished by Gaussian smoothing. At that point, pixel angle esteems are determined, in this way getting a sort of edge improvement in the picture.

F. Decoding

Argparse library is responsible for parsing command line arguments. *pyzbar.decode* is used for finding and decoding the Bar codes in our picture. It first *converts the image into RGB*, then internally calls for *draw.rectangle* or *draw.polygon* function, whichever required and then maps the proper dimensions of the shape, here it is done internally as depicted in Fig. 5

G. Implementation of the method

The proposed method is developed using the python language for automatic detection and decoding of Bar/QR codes. It is stated ealier that the proposed method also has a decoding column. This is accomplished by using the python libraries namely argparse, cv2 and pyzbar. ArgumentParser() function of argparse library helps in decoding the input command and getting the image location. From the cv2 library, we use 'imread()' to load the input image, 'rectangle()' to put



```
barcodes = pyzbar.decode(image)
for barcode in barcodes:
    (x, y, w, h) = barcode.rect
    cv2.rectangle(image, (x, y), (x + w, y + h),
        (0, 0, 255), 2)
    barcodeData = barcode.data.decode("utf-8")
    barcodeType = barcode.type
    text = "{} ({})".format(barcodeData, barcodeType)
    cv2.putText(image, text, (x, y - 10),
        cv2.FONT_HERSHEY_SIMPLEX,
        0.5, (0, 0, 255), 2)
```

Fig. 6. Image Detection and Information Extraction Code Snippet

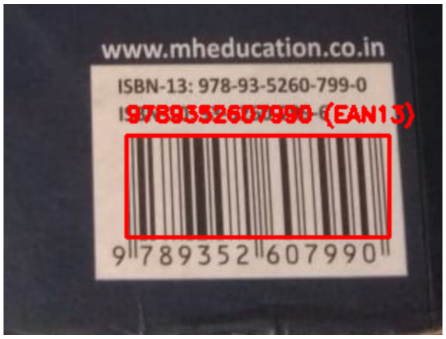


Fig. 7. Barcode detected from a book by our program

a rectangular box around the barcode/qrcode, 'putText()' puts the ISBN code we got on top of the rectangular box where the barcode is detected, and filter2D() applies filter to remove the salt and pepper noise. A loop (preferrably for loop is used) is used over the 'barcodes' which is shown in Fig. 6.

In this loop, *barcode.rect* object helps in extracting the (x,y) coordinates from the bounding box and they enable us in detecting the location of the barcode int he input image. Then we encircle the location of barcode with the red recatngular box, or little bit other shaped polygon, whichever required. Then we decode the Bar code into a "utf-8" string and extract the type of Bar code. It is utmost priority to call the *.decode("utf-8")* function. The aforementioned function converts from a string to a byte array. Henceforth, we format and draw the Bar code Data and Bar code type on the image.

IV. Results and Discussions

The experiments are conducted on personal computer (Intel core i5 (CPU), 2.5 GHz, 8GB RAM) in the Linux environment. The proposed methodology is developed using the python programming language and validated over the images of Bar/QR codes taken from the products. The basic idea to perform this work is to capture and decode the Bar/QR codes automatically using image processing techniques.

After we get the output from the barcode, then comes the part of linking with a database. The barcode output we got was '9789352607990', which we stored in a variable and matched it with the ISBN column of the dataframe, and it gave the corresponding details of the book like it's Title, Author Name, Pages, Publication, Cost and ISBN which were 'Object-Oriented Programming with C++', 'E Balaguruswamy', '588', 'McGrawHill Education', '600'and '9789352607990' respectively. When we ran the program on the QR-code, the output we got from it was "http://l.ead.me/beagle", which led us to the facebook page of the book's QR-code source. Using Python Pandas' Data frame tool, we easily link the output of our Bar Code scanner with the our csv file, and the program matches the output ISBN code we got to the ISBN column of dataframe and gives the details of the book if it is in our library or not, and details like who borrowed it etc.

TABLE I
Comparative Study

S. No.	Works	Method	Runtime (in s)	Accuracy (%)
1	Tuinstra et. al. [16]	Basic Morphological Operations	1.188	82.82
2	Coughlan et. al. [17]	Image scanning	0.387	73.84
3	Juett et. al. [18]	Bottom-hat filtering	1.314	86.79
4	**Proposed Method**	**Image Processing**	**0.25**	**87**

A brief comparison is presented in Table I with the existing works to justify the superiority of the proposed method. From Table I, it can be concluded that the proposed method has reported better accuracy and less computational time than existing methodologies. Hence, the method can widely used for bar/QR code recognition systems. The proposed method can be used for several applications such as age check, opening client accounts, business tracking, coupons, tickets, fraud location, authorization and etc.

V. Conclusion and Future Scope

This paper presents an image processing based method for automatic analysis of Bar/QR codes. The proposed method involves the filtering of the images to remove the noises associated in the images. Further, the images are converted into gray scale and binarization is performed. The experiments conducted using this yielded 87% accuracy. The method is validated on the data developed in-house by gathering the data of different items and developed in the python language. The proposed method is suitable for practical applications since the execution time taken by image is approximately 0.25 seconds to distinguish Bar code. The future scope of this work is implement the proposed method on a microcontroller platform for developing a proof-of-concept prototype for automatic detection of Bar/QR codes.

References

[1] D. Parikh, G. Jancke, "Localization and Segmentation of A 2D High Capacity Color Barcode," Applications of Computer Vision, pp. 1-6, 2008.

[2] M. Kuroki, T. Yoneoka, T. Satou, Y. Takagi, T. Kitamura, N. Kayamori, "Barcode Recognition System Using Image Processing," IEEE Int. Conf. on Emerging Technologies and Factory Automation Proceedings, pp. Sept. 1997.

[3] R. Puri and V. Jain, "Barcode Detection using OpenCV-Python," Int. res. j. of Advanced Eng. and Sci., vol. 4, no. 1, pp. 97-99, 2019.

[4] Z. Wang, A. ConradBovik, "A universal Image Quality Index," IEEE Signal Process. Lett., vol. 9, pp. 81–84, 2002.

[5] E.Ohbuchi, H.Hanaizumi and L.A.Hock, "Barcode Readers using the Camera Device in Mobile Phones," Int. Conf. On Cyberworlds, pp. 260-265, Nov. 2004.

[6] GB/T 1828-2000, Chinese National Standard-Quick Respond Code (QR Code), Standards Press of China (2001)

[7] Object Oriented Programming With C++. Balagurusamy. Tata McGraw-Hill Education, 2008 - C++ (Computer program language)

[8] QR-code generator homepage https://www.qr-code-generator.com/qr-codes-on/books/. Last accessed 14 May 2020

[9] W. N. W. Shuhaimi, "Real time barcode reader for laboratory attendance (Software Part)," Bachelor Thesis, 2007.

[10] Niblack, "An introduction to digital image processing," Prentice-Hall International, pp. 115-116, 1986.

[11] Jiazheng Shi, A. Z. Scottsdale, "Real-Time barcode recognition using general cameras, Patent no. US 8.439, 260 B2, 2013.

[12] X. Fang, F. Wu, B. Luo, H. Zhao and P. Wang, "Automatic Recognition of Noisy Code-39 Barcode," Artificial Reality and Telexistence - Workshops, 2006. (ICAT '06), pp. 79-82, Dec. 2006.

[13] V. M. Thomas, "A Universal Code for Lifecycle Management of Products," Electronics & the Environment, Proceedings of the 2007 IEEE International Symposium, pp. 180-183, May 2007.

[14] O. Gallo, R. Manduchi, "Reading 1D Barcodes with Mobile Phones Using Deformable Templates," IEEE Trans. Pattern Anal., vol. 33, no.9, pp. 1834-1843, Sept. 2011.

[15] Nisha, Dr. Ravinder khanna, "A REVIEW OF IMAGE RESTORATION TECHNIQUES FOR SALT-AND-PEPPER NOISE WITH TEST RESULTS," IJARECE, Volume 4, Issue 2, February 2015.

[16] T. R. Tuinstra, "Reading barcodes from digital imagery," Ph.D. dissertation, Cedarville University, 2006.

[17] E. Tekin and J. M. Coughlan, "An algorithm enabling blind users to find and read barcodes," in Applications of Computer Vision (WACV), 2009 Workshop on, 2009, pp. 1–8.

[18] X. Q. James Juett, "Barcode localization using bottom-hat filter," NSF Research Experience for Undergraduates, 2005.