Strategies to Optimize Machine Learning for Barcode Scanning

Mia Baden

Volgenau School of Engineering, George Mason University Fairfax, VA

Email address: Mbaden2@masonlive.gmu.edu

Abstract—*Everything that has been bought or borrowed through online shopping or in-person almost always has a barcode or a QR code. Barcodes and QR codes are a string of bits that form a code that can be read by a scanner. All it takes to determine the type of product is a single scan. Currently, all packaging and stocking of products require scanners to scan one barcode at a time. Researchers are working to detect multiple barcodes at the same difficulty replicating this in the work environments. In this paper, I will introduce why the machine learning alternatives to scanning barcodes increase the usability of barcode scanners and propose a possible solution to the problem of misidentification of QR codes.*

*Keywords— Barcode Detection, Python, OpenCV, Computer*

*Vision, Image Processing.*

**Introduction**

Currently, barcodes readers use infrared light to scan a single barcode at a time. QR scanners use a camera phone to scan one QR code at a time. Modern technology can reduce the time it takes to scan and process these items and save companies money.

For example, in a company called Metro Air Service owned by Dnata, a subbranch of the Emirates company workers open cartridges filled with packages that in total weigh an average of 30,000lb per shift according to Mark Gortayo; the assistant manager at Metro Air Service in Sterling. The package scanners use traditional infrared light to scan a single package at a time. The total estimated scanning time takes about 2 hours of the 4-hour shift. If the scanners were able to scan 3 barcodes in a single scan, the work would save an hour and 20 minutes for each employee per shift. Using scanners that detect multiple codes at once leads to an increase in productivity.

1. **MECHANISMS OF THE BARCODE**

The barcode is a set of “black and white bars which it can represent numbers from 0 to 9, letters from A to Z” [6]. The barcode not only has these bits that provide us with the data, but also are armed with formatting that makes barcodes readable from both sides. The barcode includes a “start character, data character, check character, terminator and blank areas.” [4] There are varying types of barcodes and each has mechanisms to check if the bits are right. They also have a certain capacity of information that they can hold. For example, Code128 has the following features: “two-way scanning” [4], which is universal among barcodes. Every barcode has 2 bars on the left and 3 bars on the right. If the barcode is upside down it can rotate and still read the numbers correctly by reading which side has 2 bars. The barcode also includes “A customizable check character”[4] that counts the total number of all of the bars, converts to a number, and using a modulo operation sees whether the check bit number matches the outcome of the operation. [4] The barcode is a reliable type of encoding that, with the use of infrared light can read the information accurately one scan at a time.

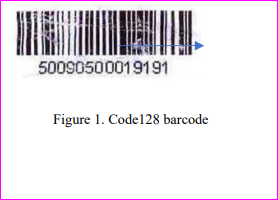


Figure 1. Code 128 barcode [4]

**3. MECHANISMS OF THE QR CODE**

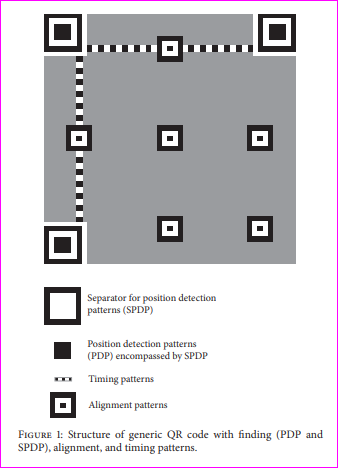
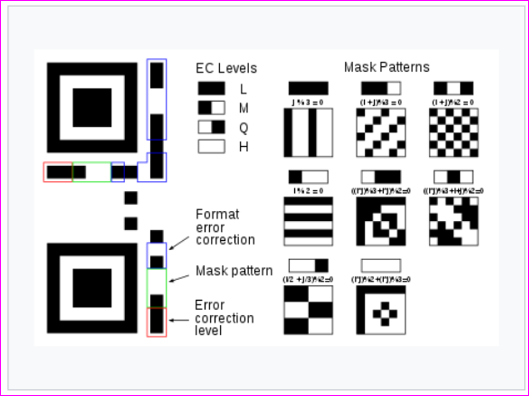
The Two-Dimensional version of a barcode are called a QR code or the “Quick Response Code”. This type of encoding was invented by a Japanese company called “Denso Wave”. [6] Unlike the barcode the encoding on the QR code is a lot more complex. The bits in this encoding are written into 2D matrix which stores the same alphanumeric data within blocks of black and white squares, called “modules”. [6]. 

Figure 2. QR Formatting [3]

Some of the features of this encoding are the 3 squares that allow cameras to locate the barcode, “the two lines that connect the 3 squares; called timing patterns that determine the size of the pixels”, [3]. A mask on the perimeter of the 3 boxes that “determines the finder pattern, that inverts some of the pixels to make them different from other QR codes.” [3]. (All of these formatting features are displayed in Figure 2 above)

The camera first identifies the boxes, then looks at the timing patterns, then the masking pattern.

 Figure 3. QR Masking patterns

In figure 3 the different types of masking patterns are shown and how the machine knows what type of masking pattern is in the QR code. The masking pattern is placed over the data and inverts specific bits that lie on top of the mask. If the bit that lies beneath the masking pattern is black and the masking pattern is black it inverts the bit into a white bit. After the camera unmasks the numbers underneath, the scanner reads the bits in a “vertical zig-zag manner.” [3]

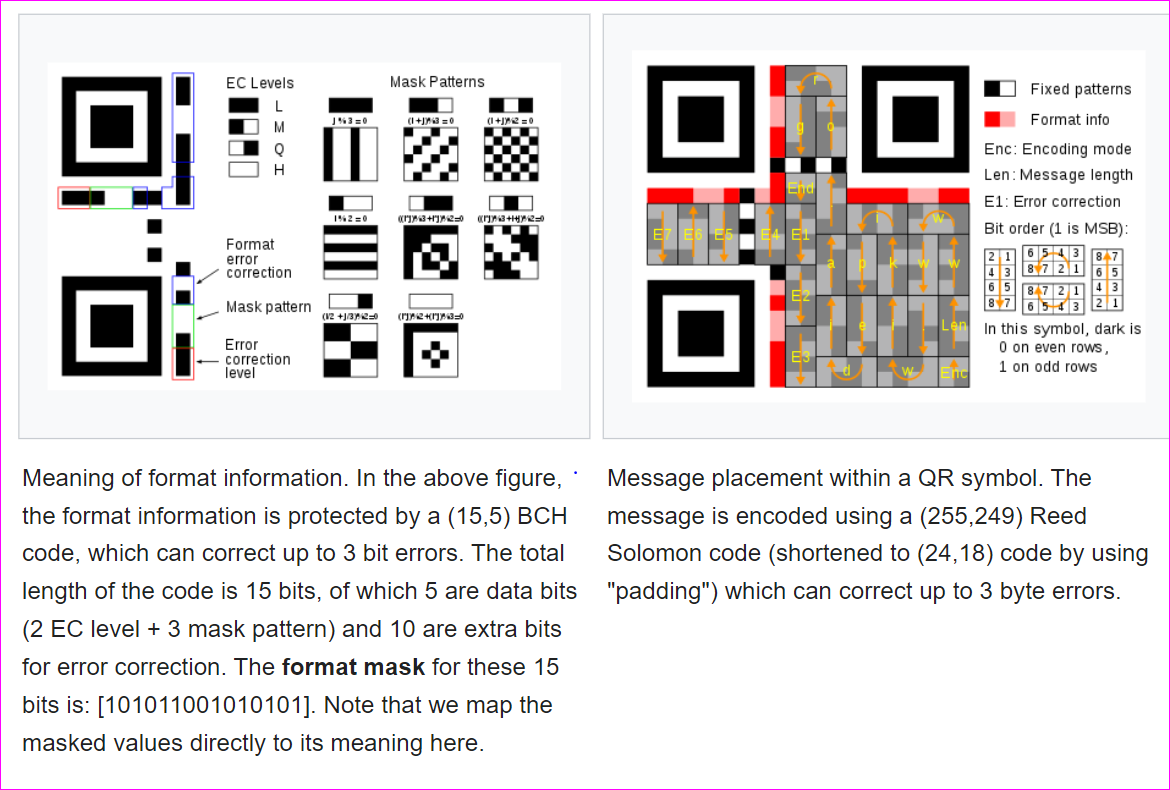


Figure 4.

The bits are read in the order that is demonstrated in Figure 4 on the right of the QR code image. The cells as shown on the right consist of 8 bits which correspond to an individual binary number. [3] Within the QR code image, the words encoded are shown. The rest of the bits are used for an “error correction algorithm that was invented by Reed Solomon that divides a QR code into several disjoint RS blocks and perform the error correction on each block”[6]

**2. MECHANICS OF MACHINE LEARNING**

4.01 Barcode Detection Algorithms



Figure 5.

**SPIRAL SEARCH** - In the field of computer vision there are different types of barcode detection algorithms. One of the older algorithms called “Ohbuchi’s algorithm”[1] locates both QR and EAN-codes using a spiral search (Figure 6 (2)), which means going in a clockwise spiral of the bits to find specific black bars in the barcode.

**BLOB DETECTION** - In Open CV there is a library called BLOB detection “that would focus on an object that is of a specific size.” [3] Similar to an eye going to look for shapes in Eye Spy, this neural network algorithm tries to look for specific shapes that resemble the images fed into the machine previously. For example, when kids try to look for a candy cane in the I Spy book, they look for the shape of the cane or the red and white pattern, ignoring shapes and colors that aren’t similar to the object they are looking for. “This mechanism called BLOBS or binary large objects consist of 8 connected pixels that uses iterative filtering.” [3] This is applied to remove the regions that are not geometrically like the standard QR code. This is determined by feeding the machine with pictures of barcodes and training the machine to find barcodes.

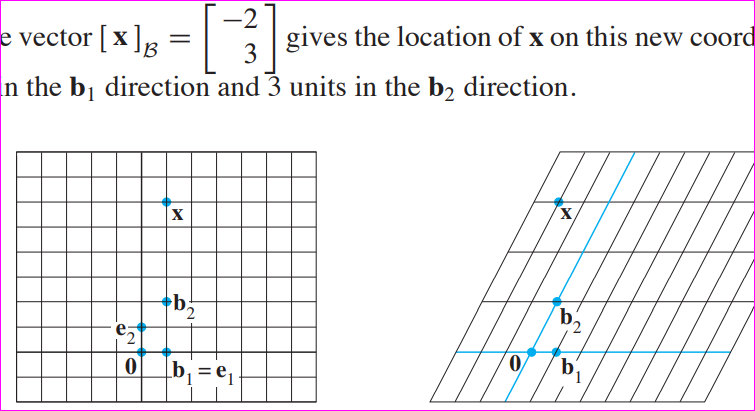
BLOBS and other mechanisms like it require databases to train them on such as the popular “Muenster Barcode DataBase and the extended Arte-Lab database”. [1] The BLOBS detected in Figure 4 shows the algorithm finding the separator pattern (From Figure 2). Once the barcode has been located, the algorithms can begin the decoding process stated above in Section 2 and 3 and decode every synthetic QR code in the database.

One popular tool to decode the barcode is the Open CV’s (open computer vision) ZBar library. An open-source barcode and QR reader. In addition to returning a Boolean value indicating the readability of the QR code, ZBar is also able to provide whether a module is decoded

correctly. [6]

**4.10 Edge Case mechanisms**

QR code detection is low performance for rotated and distorted single or multiple symbols in images with distortion, blurring, and excessive overlapping data. [2-3-7-1]. This can be an issue in a warehouse where cartridges are dark and the packages are covered in dust or can be at different angles.

 Figure 5

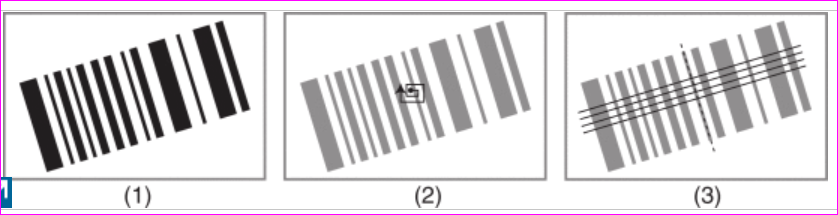


Figure 6

**3.21 Barcode Distortions**

Once a separator pattern is detected, is usually not in the perfect 90-degree angle. (Figure 6 (3)) The process in which the axis on which the image lies can be changed is called changing the basis of a matrix into the standard basis. The combinations of these approaches for QR codes were, in Hansen’s experiment a success of 90 percent rate for images that have single barcodes.

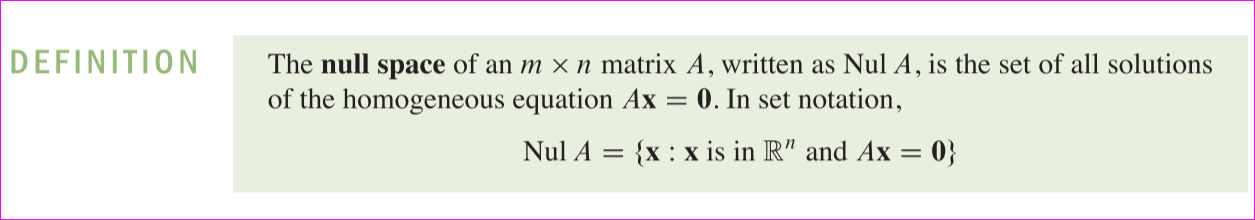


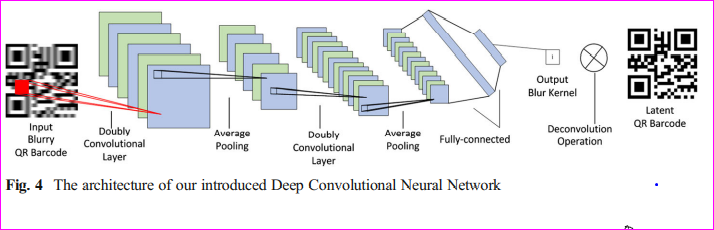
Figure 7. Null Space of a Matrix [1]

**3.22 Barcode Blurring**

Cameras do not capture images with absolute accuracy. Like human eyes, there is an optical point where things are clear but beyond that point, images start to blur. Similarly, if a camera is moving too fast it can also cause a blur. The recovering of the information within the blurry QR barcode images can be “accomplished by using a novel CNN (Convolutional Neural Network) architecture adapted to the characteristics of QR barcode.” [7]

In a Convolutional Neural network, filters are used to emphasize some features and deemphasize other features of the image. For example, a filter with vertical images would ignore horizontal lines and bring out the vertical lines in an image, which then can be used to detect a line.

In this CNN, each filter used to deblur the barcode requires a specific linear transformation called a Kernel or a null space of the matrix. “Once the proper kernel is applied and the original image is convolved with an “out-of-focus or motion blur kernel. The CNN architecture enables the machine to decide based on the training of the machine on previous barcodes, which filter to use” [7] (Figure 8) After the convolutions have completed the output is a non-blurred image.

 [7] Figure 8

**3.23 Barcode Overlapping data**

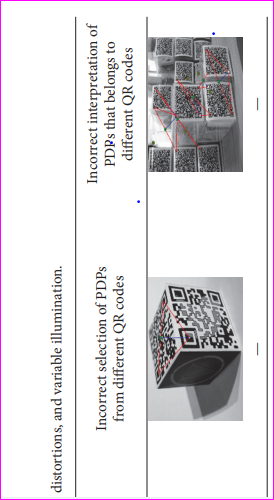


Figure 7. [4]

In the Blob detection [3] algorithm there arose a particular issue with having multiple QR codes close together. In these cases, the separator component of the QR code was detected but the Blob detection misclassified multiple barcodes as a single barcode. (Figure 7) This would cause issues in warehouses because many barcodes are stacked together in the First-Class section of the package delivery service.

For the Blob mechanism I propose that it should not only identify the blobs but the format information around the QR code since each format information that surrounds the separator pattern is unique. Once the Blob detects a singular separator pattern it should immediately register the encoding pattern and if the mask pattern and the error correction bit is the same it should register the encoding as a single QR code.

**REFERENCES**

[1] Hansen, D. K., Nasrollahi, K., Rasmussen, C. B., & Moeslund, T. B. (2017). Real-Time Barcode Detection and Classification using Deep Learning*.*

[2] Li, J., & Guan, W. (2018). The optical barcode detection and recognition method are based on visible light communication using machine learning. *Applied Sciences*, 8(12), 2425.

[3] Lopez-Rincon, O., Starostenko, O., Alarcon-Aquino, V., & Galan-Hernandez, J. C. (2017). Binary large object-based approach for QR code detection in uncontrolled environments. *Journal of Electrical and Computer Engineering*, 2017

[4] Xu, Y., Qu, Z., & Zheng, S. (2019, August). High-speed barcode recognition system based on OpenCV and Zbar. *Eleventh International Conference on Digital Image Processing* (ICDIP 2019) (Vol. 11179, p. 111791F).

[5] Raghav Puri, & Vikram Jain. (2019). Barcode Detection using OpenCV-Python. *International Research Journal of Advanced Engineering and Science*, 4(1).

[6] Chu, H. K., Chang, C. S., Lee, R. R., & Mitra, N. J. (2013). Halftone QR codes. ACM Transactions on Graphics (TOG), 32(6), 1-8.

[7] Pu, P. (2019). Quick response barcode deblurring via doubly convolutional neural network. Multimedia Tools and Applications, 78(1), 897–912. <https://doi.org/10.1007/s11042-018> 5802-2