A More Robust Model For QR Code Detection

1st Jingtong Zhang
Khoury College
Northeastern University
Vancouver, Canada
zhang.jingto@northeastern.edu

2nd Yuqin Luo
Khoury College
Northeastern University
San Jose, United States
luo.yuqin@northeastern.edu

3rd Ying Bi
Khoury College
Northeastern University
San Francisco, United States
bi.yi@northeastern.edu

***Abstract*—A more robust model invariant to translation, rotation, and scale for detecting QR codes with better performance under conditions like low brightness, long-distance, and occlusion. Previous work like QRCodeDetector implementation in the OpenCV library reaches a fairly limited accuracy. In our work, we introduce a pipeline for QR code detection that concentrates on the thresholding, perspective translation, and feature matching of position box with normalization, thus adapting to more extreme scenarios. We collect a small dataset where the QR Code displayed under multiple conditions, the detection accuracy shows our method is not as good as the WeChat implementation (Neural Network Based), but it is much better than OpenCV Library implementation (Non-Neural Network Based).**

***Index Terms*—detection, thresholding, normalization, feature match, neural network**

I. Introduction

Quick Response Code (QR Code) is a type of matrix Barcode (or two-dimensional Barcode) invented in 1994 by the Japanese automotive company Denso Wave. In practice, QR codes often contain data for a locator, identifier, or tracker that points to a website or application. The Quick Response system became popular outside the automotive industry due to its fast readability and greater storage capacity compared to standard UPC Barcodes. Applications include product tracking, item identification, time tracking, document management, and general marketing.

A QR code consists of black squares arranged in a square grid on a white background, which can be read by an imaging device such as camera. The required data is then extracted from patterns that are present in both horizontal and vertical components of the image. This project aims to improve the recognition capability and accuracy by introducing advanced pre-processing and feature matching methods. The recognition success rate is highly dependent on a series of factors such as angle, light, distance, etc. Thus, our recognition model intends to be more robust for these occlusion, low-light, warping and small-scale environments. As a comparison, the recognition results from OpenCV and WeChat WeChat on the same dataset will be displayed and analyzed.

II. Related Work

In this section, we briefly introduce previous work on QR code detection. Considering the remarkable performance of deep learning methods for QR code detection, we simply categorize previous work as non-neural based and neural based methods.

Non-neural Based Methods. The traditional methods for detecting the QR codes mainly focus on the pattern matching in particular the position box matching. Therefore, the initial work on QR code detection tried to improve the processing methods to improve the performance. Liu *et al.* [1] proposed an approach for the QR code recognition on mobile phone consisting of series of operations: Gray Conversion, Binarization, Filter, Orientation, Patterns Location Alignment, Grid Generating. Those pre-processing steps provide initial idea and solid basis for our self-implemented version, especially the adaptive multi-level thresholding algorithm that could significantly improve recognition accuracy. Belussi *et al.* [2] performed an exhaustive analysis of the characteristics named FIP (Finder Pattern), TP (Timing Pattern), and AP (Alignment Pattern).The concrete description of the FIPs properties could be used to locate those three position boxes in our implementation.

Neural Based Methods. With the prevail of convolutional layer and deep network, the QR code detection task could be solved by these neural networks instead of concentrating mainly on the image processing. These neural based model may perform better when facing with the occlusion in particular the position box occlusion that is undesirable for the traditional methods. Chou *et al.* [3] proposed an algorithm that localizes and segments two-dimensional QR codes with simply two convolution layers and two sub-sampling layers to do the feature extraction. The CNN based model provided by WeChat as third party extension in OpenCV includes two CNN-based models: A object detection model and a super resolution model. Object detection model is applied to detect QR Code with the bounding box and super resolution model is applied to zoom in QR Code when it is small. Although the implementation detail is not released, we could try to figure out how the object detection model trained with this paper.

III. Methods

We designed our own algorithm based on analysis of QR code characteristic, binary image, contour extraction, perspective transformation and geometric analysis. The specifics of our solution contains 4 preparation methods and filters and 2 core algorithms as follows:

Thresholding Firstly, we do OTSU thresholding before detecting and decoding QR code images. In OTSU Thresholding, a value of the threshold isn't chosen but is determined automatically. A bimodal image (two distinct image values) is considered. The histogram generated contains two peaks. A generic condition would be to choose a threshold value that lies in the middle of both the histogram peak values.

FindHomography We applied findHomography() and warpPerspective() in perspectiveTransformation. findHomography() is an iterative method for estimating a mathematical model from a data set that contains outliers with the default method as RANSAC. In this project, perspective transformation is really important because we want to align the QR code image properly even in different angles.

WarpPerspective We applied the OpenCV method of warpPerspective function to deal with normal transformation. It transforms the image in a straight manner after Perspective Transformation is applied to it.

MinAreaRect OpenCV provided this function cv.minAreaRect() for finding the minimum area rotated rectangle. We used this function combined with cv.points() to draw boxes according to collected vertices. We will extract multiple contours to analyze and roughly get candidate corners, based on the position of these corners, it is easy to find a circumscribed rectangle.

X corners Based on the geometric characters of QR code, in ROI, we checked the center pixel first. It has to be black and be symmetric from the center to both left and right direction. The black and white area ratio should be 1:1:3:1:1, which is the feature for geometric analysis.

Y corners In vertical direction, the process should be similar to horizontal direction. Since we have a strict condition in the X axis, we can simplify the previous algorithm in the Y axis as the overall ratio of black and white area should be 3 : 2.

Fig. 1. Standard QR code detection by our method

Fig. 2. Rotated QR code detection by our method

IV. Experiments and Results

A. Detection results with our method

We used three QR code images for the experiment. The first 1 one is a standard QR Code image. The second 2 image is from the Eastern Airline ticket with a rotated QR code. The last 3 one is also from the Eastern Airline ticket, but this QR code are located in a curled paper. Our method can detect the QR code successfully, so our method can do the detection when the pictures are taken from the better light conditions even though the QR code is rotated, scaled or curled.

B. Compare the result with other QR code detection method

For this experiments of this project, our team utilized the original OpenCV QRCode detection method and the WeChat QRCode detection method to recognize the same pictures. The picture set of the experiments has three different objects with QR codes. The pictures of the objects are under different light conditions and some QR codes are occluded partially in some pictures.

1) OpenCV QRCode detection method: The OpenCV QRCode detection is the basic QR detection method and this method has become the benchmark against which all other methods are compared. Based on the source code *et al.* [4] and document of OpenCV *et al.* [5], the latest version detection method can detect the minimum-area quadrangle containing the code. The method also allowed the user to set the different epsilon when it scans the whole QR code vertically and horizontally. This epsilon can help the method to determine the 1:1:3:1:1 pattern of the QR code. The OpenCV method also supports the decode function that can decode the QR code in the image when it finds the QR code.

2) WeChat QRCode detection method: The WeChat QRCode detection method is using the Convolution Neural Network method to scan and detect the QR code *et al.* [6]. They have two Convolution neural networks inside the detection model. The detector model: The CNN model can help us to find the QR code located in the image roughly. The second model is the super-resolution model. The scanner are using

Fig. 3. Curled QR code detection by our method

the super-resolution model to enhance the resolution of the QR code. Thus, this model is able to find the QR code even if it is very small when we take a picture from far away.

Our team used three different QR codes for this experiment, including the China Eastern Airlines QR Code, the QR code from the IQIYI Inc, and the QR code of the Coca-Cola company. These three QR codes have unique features. The QR code images from IQIYI INC are taken from the Ipad, so it is the clearest QR code. The China Eastern Airlines QR Code images are from the printed airline ticket, so its resolution is a little bit worse than the electric one. The Coca-Cola QR code images are from the Coke bottle, so the QR code is located on the curled plane.

We did three different experiments for all of the three QR codes. The images were taken from different conditions. Each experiment is testing all detection methods for all QR codes.

C. Small images recognition experiments

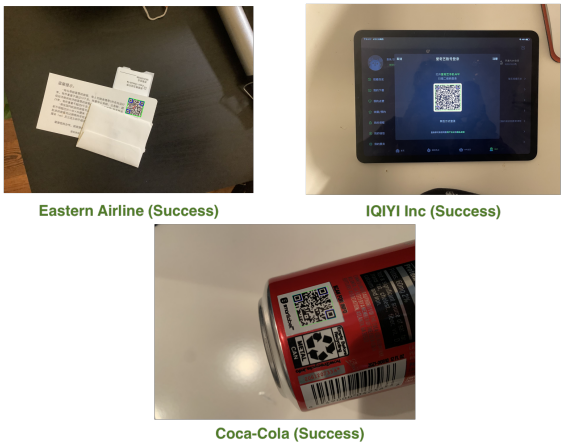

Fig. 4. Far QR codes detection result from our method

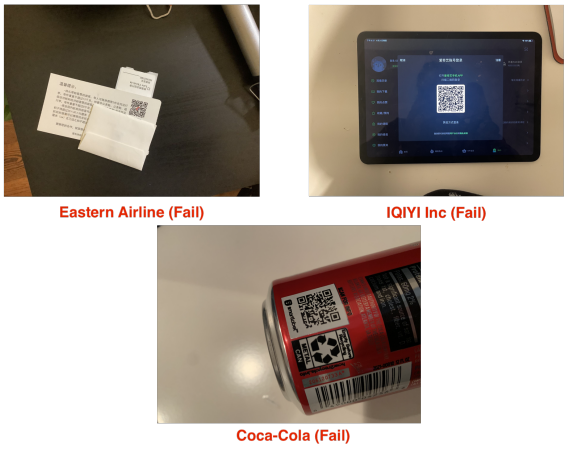

Fig. 5. Far QR codes detection result from OpenCV method

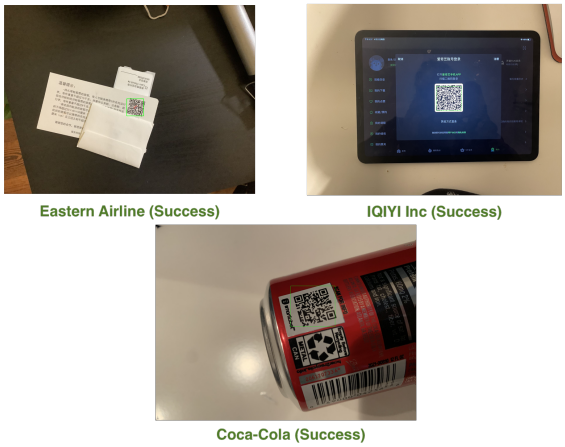

Fig. 6. Far QR codes detection result from Wechat method

The first experiment is based on small images. Our method and the Wechat method can detect the QR code successfully for all of the images, but the OpenCV method failed for all of the images. The result is Shown in Fig.4, Fig.5, and Fig.6.

The basic OpenCV QR code detection method looks cannot handle the small QR code detection. Our method utilizes the normalization method and wrap-perspective function so we can handle the small size QR code. As we mentioned above, the Wechat QR code detection is using the super-solution CNN model so it can detect the small QR code, so also can detect all of the QR codes.

D. Low light images recognition experiments

The second experiment is based on QR code images that was taken by the low light condition. The Wechat method can detect all of the QR codes successfully. Our method is failing on the Coca-Cola bottle image. The OpenCV method is still failing for all of the images. The result is Shown in Fig.7, Fig.8, and Fig.9.

As all of the images were taken on the low light condition, the white and black border is not clear and the pixel level

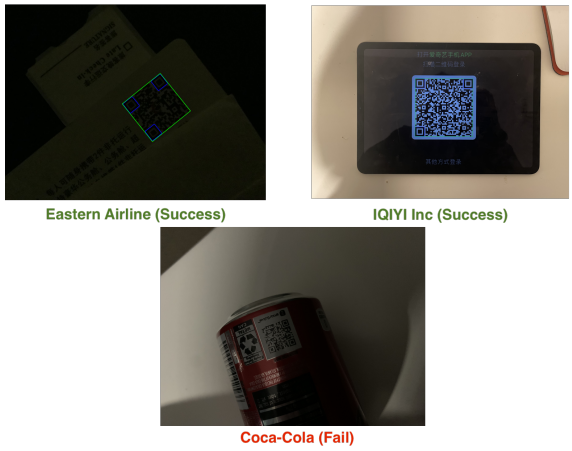

Fig. 7. Low light QR codes detection result from our method

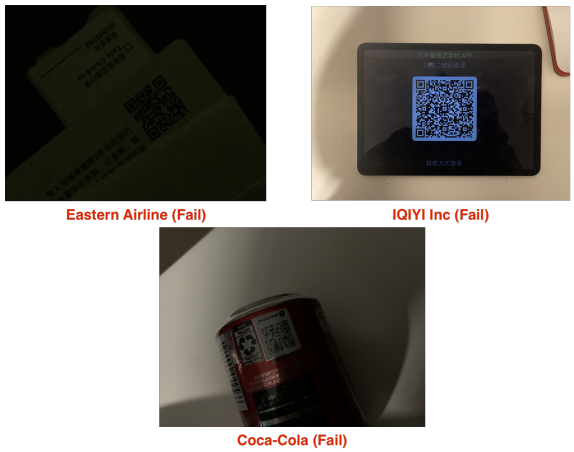

Fig. 8. Low light QR codes detection result from OpenCV method

detection is difficult to find the QR code pattern. However, the Wechat method can fit this situation as their model can learn the features based on other training data set.

E. Occlusion images recognition experiments

The last experiment is based on images that has other object occlude the QR code. As the QR code of the Eastern Airline Ticket are printed in the paper and it not easy to put the object. Thus, we used another Basic QR code. The result is Shown in Fig.10, Fig.11, and Fig.12.

The original OpenCV method can detect the basic QR code. Our method can detect the QR code of IQIYI Inc and the QR code of Coca-Cola. The Wechat method can detect the basic QR Code and the QR code of Coca-Cola. The result from this experiment is a little bit interesting. The performance of our method and wechat method are same. The original OpenCV method can detect the Basic QR code as it has the standard QR code pattern. Our method failed as we cannot find the left bottom corner box during the contour finding step.

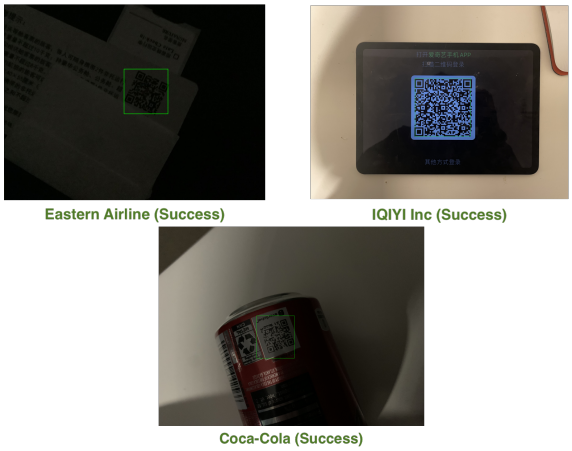

Fig. 9. Low light QR codes detection result from Wechat method

V. Discussion and Summary

From our experiment, we found the conditions of different kinds of images can affect the final result significantly. Although the QR code (Quick Response Code) should be easily detected by the Computer Vision method, the images from some specific conditions also cannot be detected and more QR Codes images cannot be decoded when they are detected.

TABLE I
THE SUCCESS DETECTION RATE OF THE WHOLE IMAGE SETS

	Our Method	OpenCV	Wechat
Small QR code cases	67%	0%	100%
Low light QR code cases	33%	0%	87%
Occlusion QR code cases	50%	17%	50%

As the table.I shown above, the Wechat solution has the best result among all of the methods and works well for all of the conditions of images. The original OpenCV method only detect one image from the whole data set. Our method can detect well in the small QR code data set and occlusion QR code data set, but we failed most of the cases for the image under the low light condition. Our work focuses on the pixel level detection and the Wechat method are using the CNN methods. Thus, we may need to combine our method and the wechat method in the future to get the better result.

References

[1] Liu, Yue, Ju Yang, and Mingjun Liu. "Recognition of QR Code with mobile phones." 2008 Chinese control and decision conference. IEEE, 2008.
[2] Belussi, Luiz, and Nina Hirata. "Fast QR code detection in arbitrarily acquired images." 2011 24th SIBGRAPI Conference on Graphics, Patterns and Images. IEEE, 2011.
[3] Chou, Tzu-Han, Chuan-Sheng Ho, and Yan-Fu Kuo. "QR code detection using convolutional neural networks." 2015 International conference on advanced robotics and intelligent systems (ARIS). IEEE, 2015.
[4] OpenCV QR detection source code https://fossies.org/linux/opencv/modules/objdetect/include/opencv2
[5] OpenCV QR detection doc https://docs.opencv.org/4.5.0/de/dc3/classcv
[6] Wechat QR detection doc https://learnopencv.com/wechat-qr-code-scanner-in-opencv/

Fig. 10. Occlusion QR codes detection result from our method

Fig. 11. Occlusion QR codes detection result from OpenCV method

Fig. 12. Occlusion QR codes detection result from Wechat method