QR Code Detection Using Convolutional Neural Networks

Tzu-Han Chou, Chuan-Sheng Ho, Yan-Fu Kuo

***Abstract*—Barcodes have been long used for data storage. Detecting and locating barcodes in images of complex background is an essential yet challenging step in the process of automatic barcode reading. This work proposed an algorithm that localizes and segments two-dimensional quick response (QR) barcodes. The localization involved a convolutional neural network that could detect partial QR barcodes. Majority voting was then applied to determine barcode locations. Then image processing algorithms were implemented to segment barcodes from the background. Experimental results shows that the proposed approach was robust to detect QR barcodes with rotation and deformation.**

I. Introduction

Barcodes are symbols with encoded data that can be read by optical scanners. Originally, barcodes consist of a series of parallel lines with varying thickness and spaces. Because the limited storage capacity of the line-type patterns, barcodes later evolved into figures of two-dimensional (2D) forms such as rectangles and dots. These barcodes of 2D geometries, also referred to as 2D barcodes, are widely applied in various fields in recent years. The detection of arbitrarily located and rotated 2D barcodes in the images of complex background is essential to reduce the amount of user interaction involved in the process of barcode decoding. This study aimed to detect quick response (QR) barcodes using convolutional neural networks (CNNs).

The topic of automatic barcode localization in images has been addressed by some literature. Zhang et al. [1] established approaches to detect non-uniformly illuminated and perspective distorted 1D barcode based on their textual and shape features. Zamberletti et al.[2] proposed angle variant 1D barcode detection algorithms using multilayer perceptron network. Xu et al. [3] developed an approach for detecting blur 2D barcodes based on coded exposure algorithms. Ohbuchi et al. [4] proposed a QR code detector that searches the image for finder patterns. Leong et al. [5] introduced the identification of barcode keypoints and lines by using speeded up robust features. Although these methods have high detection rates on certain barcodes, their performances may be affected by different environmental conditions. The above methods are based on handcrafted features using prior knowledge of specific conditions. Defining handcrafted features is labor-intensive and time consuming. In addition, inappropriate or insufficient handcrafted features may result in suboptimal detection rate.

H. C. Author is with the National Taiwan University, Bio-Industrial Mechatronics Engineering Dept. (e-mail: pizza15425@ gmail.com).

C. H. Author is with the National Taiwan University, Bio-Industrial Mechatronics Engineering Dept. (e-mail: d01631007@ntu.edu.tw).

Y. K. Author is with the National Taiwan University, Bio-Industrial Mechatronics Engineering Dept. (corresponding author to provide phone: 2-33665329; fax: 2-23627620; e-mail: ykuo@ntu.edu.tw).

978-1-4799-1851-5/15/$31.00

CNNs are multilayer perceptron classifiers. They consist of convolution and subsample operations inspired from visual cortexes [6]. CNNs are developed directly using raw images, as compared with other presumption approaches that usually require prior knowledge to define hand-engineered features as the classifier inputs. The approach has shown to be robust to noises and variations. LeCun et al. [7-9] proposed strategies of deep network design with back-propagation in recognition and proposed an architecture of CNN which became popular because of its outstanding performance in handwritten digit recognition. Then other CNNs were proposed for detecting faces [10, 11], identifying vehicle license plates [12], tracking pedestrian movement [13], reading speed signs [14], and recognizing facial expression [15].

This work aims to locate QR codes of arbitrary orientations and scales in complex backgrounds. The specific objectives were to (1) collect large sets of QR code as database, (2) identify if local patches of images are partial QR codes using CNN, (3) segment QR codes locations using optimized image processing methods, and (4) evaluate the performance of the proposed model.

II. The Proposed Approach

The proposed approach determined if local patches of an image are partial QR codes using CNN classifiers. The detection process first involved a spatial pyramid that scaled the input images and then partitioned the scaled images into several local patches. The local patches were subject to the CNN classifiers for partial QR code detection. Once positive detection were determined, region smearing technique was applied to determine patches of the same QR barcodes. The barcode images were then segmented from their background using optimized image processing technique, including Hough transform and perspective control .The flow chart of the testing process is shown in Figure 1.

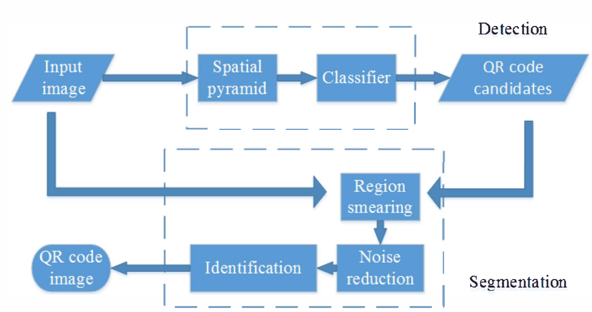

Figure 1. Detection process.

A. *Database of partial QR code patches*

Databases of QR code images and non-QR code images were established. In the process of image collection, 200 QR codes of version 7 were randomly generated using an online barcode generator. They were then printed using an electrophotography printer (LaserJet M1132, HP; 600dpi) at a size of 15 mil per module. The images of the QR code printouts were acquired by a barcode scanner (9200 series, CipherLab), mimicking the process of 2D barcode scanning. The QR code printouts were placed approximately 10 and 20 cm away from the scanner during the acquisition. The distance was set for the QR code images to have module sizes of approximately 4 and 7 pixels. Non-QR code images were collected from the background area of the barcode image (e.g., texts, figures, and blank areas.)

Training samples for the subsequent CNN classifier development were created from the database. The samples were patches of partial QR code images. In the process of sample patch creation, the images were downsampled to establish QR codes with various module sizes. The rescaled images were then segmented into 32 32 patches in a non-overlapping manner. As a result, a total of 989 QR code patches and 3180 background patches were gathered.

B. *CNN architecture*

A CNN system modified from [9] was developed for identifying partial QR patches. The input to the system was an image patch of 32 32 pixels. The network determined if the patch was part of a QR code. The CNN system consisted of six layers, including two convolutional layers C1 and C2, two subsampling layers S1 and S2, and two classification layers N1 and N2 (Fig. 2). Layers from C1 to S2 contained a series of planes, also referred to as feature maps. The feature maps functioned as trainable feature extractors. Layers N1 and N2 were comprised of fully connected neurons. The extracted features were fed to layers N1 and N2 to perform classification.

Layer C1 and Layer C2 contained 6 feature maps of 28×28 pixels and 12 feature maps of 9×9 pixels, respectively. In the process of the feature map calculation, convolution operations were first performed on the previous layer and trainable kernel matrices of 5×5 pixels. Each resulting convolution matrix was summed with a trainable bias and was fed into a sigmoid function to form a feature map. Therefore, Layer C1 contained 156 (25×6+6) trainable parameters, and Layer C2 contained 312 (25×12+12) trainable parameters. Layer S1 contained 6 feature maps of 14×14 pixels, and layer S2 contained 12 feature maps of 5×5 pixels. These feature maps were the results of subsampling by a factor of 2 on the feature maps in C1 and C2. Layers N1 and N2 were classical multilayer perceptron network. Layer N1 contained 300 neurons each of which connected to a pixel in layer S2. Layer N2 comprised 2 neurons fully connected to all the neurons in N1. The N2 neurons were outputs of a sigmoid function on the weighted sum of all neurons in N1 added biases. Therefore, layers N1 and N2 contained 600 trainable weights and 2 trainable biases. The output value of neuron in N2 was used to classify the input image as QR code or background.

Stochastic back-propagation was applied to train the 1070 parameters of the CNN model. The algorithm shuffled the training data and arranged them into different batches. The batches were subsequently applied for updating the model parameters through back-propagation. The randomization of the training data increased the probability of escaping from local minima during the training, and therefore improved convergence. The system was trained by cycling through all the batches for 2000 epochs.

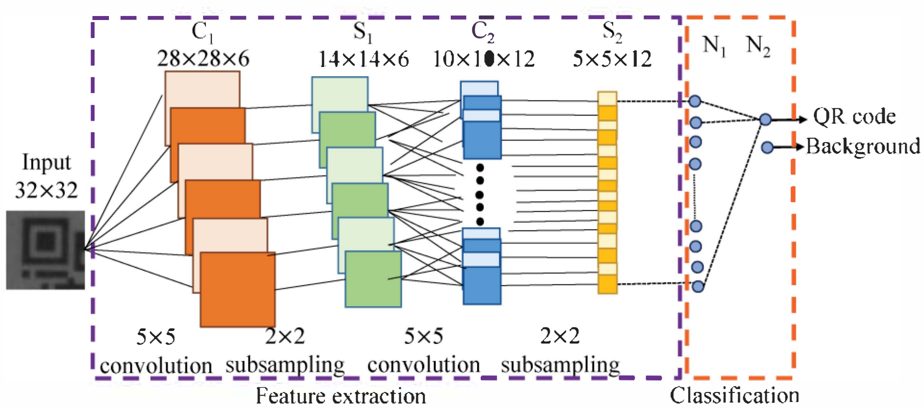

Figure 2. CNN architecture.

C. Detection of QR codes at various scales

Spatial pyramid (SP) [11] was applied to enable the detection of QR codes at various scales. In the SP process, the input image was subsampled to various scales, forming a pyramid of images (Fig. 3). The images were then partitioned into several patches (32×32). The patches were subject to the developed CNN classifier for detecting QR codes. Once detected, the locations of the patches in the subsampled images were projected back to the input image for the subsequent process. In this study, the target range for the modules of QR codes to be detected was set to 3 to 13 pixels. Since the training samples were with module sizes of approximately 2 and 3.5 pixels, the SP factors were set to 0.7, 0.5, and 0.3 (Table 1).

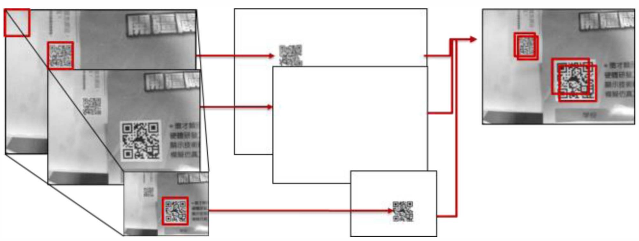

Figure 3. Image pyramid search

TABLE I. DECTECTION RAGE OF QR CODE

Spatial pyramid	*Module size detected*
0.7	5.7 pixel / module
0.7	2.8 pixel / module
0.5	7 pixel / module
0.5	4 pixel / module
0.3	13.3 pixel / module
0.3	6.6 pixel / module

D. Barcode segmentation

Image processing techniques were applied to determine the complete region of the barcode in an image for further decoding (Fig. 4). The determination process included 3 steps: region smearing, noise reduction, and barcode identification. In the region smearing process, Bradley thresholding was first applied to binarize the original image. The module size of the barcode was determined as the switch frequency between black and white regions in two consecutive QR-code patches. The module size was then applied for generating a structuring element for the subsequent edge detection, morphological closing, and morphological filling. The resulting image was expected to contain an image blob that is a complete QR barcode. Next, morphological opening was applied to reduce the noise sparkles in the image. After that, the blob associated with the highest density of positive QR-patch detection was identified as the barcode region. Hough transform was performed to detect the boundaries surrounding the blob. Lastly, perspective transformation was applied to restore the distorted barcode [4] to a square (Fig. 5).

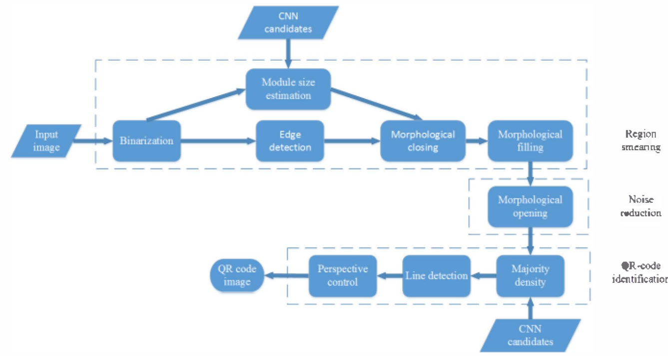

Figure 4. Barcode segmentation steps.

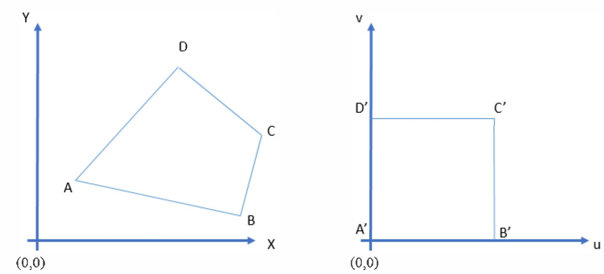

Figure 5. An illustration of the inverse perspective transformation

III. EXPERIMENTAL RESULTS AND ANALYSIS

A. Results on test data sets

The performance of our method was evaluated using 125 images gathered from the internet or provided by CipherLab (Taipei, Taiwan; Fig. 6). The QR-code images are accessible at http://goo.gl/e3bXj9. The images have complex backgrounds and are nonuniform in illumination. The QR codes in the images are of various versions, module sizes, rotations, and tilt angles. The proposed approach attained an accuracy of 95.2%. Fig. 6 shows the detection results.

Figure 6. Results obtained on test dataset

B. *Blur robustness and module size Sensitivity*

The blur robustness and module size sensitivity of the CNN classifier was investigated. In blur robustness analysis, 1798 patches of partial QR barcodes were collected. These patches were not included in the CNN model training and were positively classified as QR patches by the developed classifier. Blurring was applied to these patches using a Gaussian smoothing filter with a 3×3 mask. The standard deviation (SD) of the filter ranged from 0 to 2.5 with an increment of 0.1 (Fig. 7). The blurred patches were then fed to the CNN classifier.

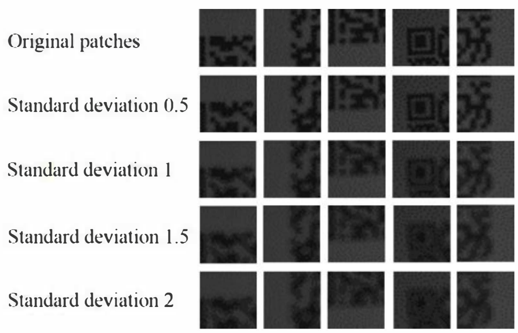

Figure 7. Sample blurred pathces for robustness analysis

Figure 8 displays the detection rate of the patches blurred using various SD. The detection rate remained reasonably high (98%) when the SD was 0.5. An obvious drop of the detection rate was observed after the SD reached 0.5. However, the detection rate reached 82.5% even with a SD of 2.5.

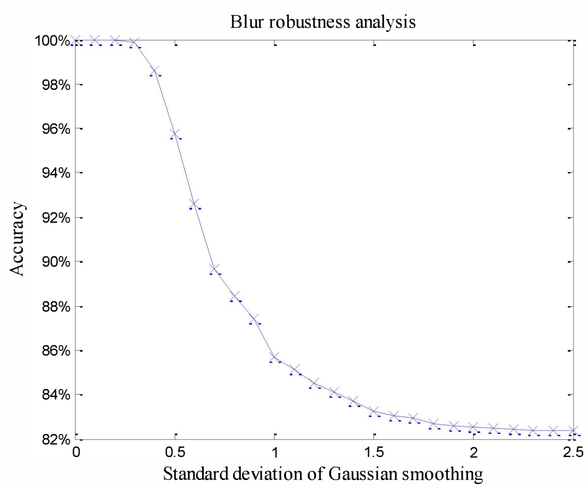

Figure 8. Standard deviation of Gaussian smoothing

The detection rate for QR-code patches of various module sizes was analyzed. Twenty QR-code images for each module sizes of 3, 5, 7, and 9 pixels were acquired. The images were then cropped into patches to test the detection rate. Figure 9 shows the detection rates of different module sizes using various SP factors. The figure indicates that QR codes with smaller module sizes can be detected by using higher SP factors. By contrast, lower SP factors gives higher detection rates for QR codes of larger module sizes. An SP factor of 0.5 shows relatively consistent detection rate for various module sizes, which is used as the dominant detection for QR code with 7 pixel per module size.

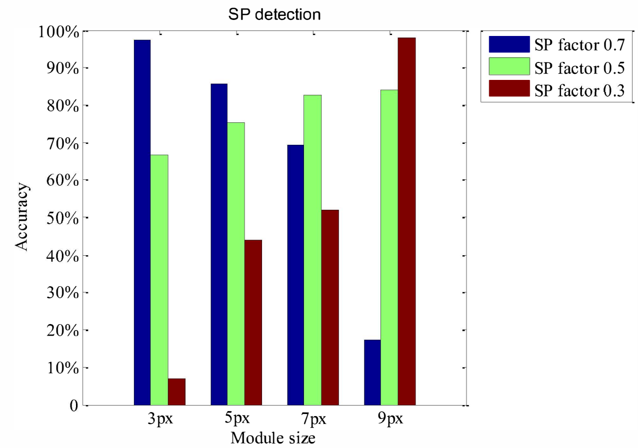

Figure 9. The detection rate of different module size in SP factors

IV. Conclusion

This study presented a framework for detecting QR codes in images of complex background. The approach identified partial barcode patches of various module sizes using a CNN and an SP scheme. The detected patches of the same barcode were then connected and segmented from the background for decoding. This strategy of partial barcode patch detection made it possible to identify barcode with large degrees of distortion. Analysis demonstrated that the proposed approach was robust to blur, rotation, and lighting uniformity of barcode images. The proposed approach reached a detection rate of 95.2% from a database composed of 125 QR codes.

References

[1] C. Zhang, J. Wang, S. Han, and M. Yi, "Automatic Real-Time Barcode Localization in Complex Scenes," in *Image Processing. 2006 IEEE Int. Conf,* pp. 497–500.

[2] A. Zamberletti, I. Gallo, and S. Albertini. "Robust Angle Invariant 1D Barcode Detection," in *Pattern Recognition (ACPR). 2013 2nd IAPR Asian Conf.*

[3] W. Xu, and S. McCloskey, "2D Barcode localization and motion deblurring using a flutter shutter camera," in *Applications of Computer Vision (WACV). 2011 IEEE Workshop*, pp.159-165.

[4] E. Ohbuchi, H. Hanaizumi, and L.A. Hock, "Barcode readers using the camera device in mobile phones," in *Cyberworlds. 2004 Int. Conf*, pp.260-265.

[5] L. K. Leong, and W. Yue, "Extraction of 2D barcode using keypoint selection and line detection," in *Advances in Multimedia Information Processing-PCM 2009. Springer*, pp. 826-835.

[6] D. H. Hubel, and T.N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," The Journal of physiology, 1962. 160(1), pp.106-154.

[7] Y. LeCun, "Generalization and network design strategies," Connections in *Perspective. North-Holland, Amsterdam*, 1989, pp. 143-55.

[8] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems. 1990. Citeseer.*

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998. 86(11), pp. 2278-2324.

[10] S. Lawrence, C. G. Lee, A. T. Chung, and D. B. Andrew, "Face recognition: A convolutional neural-network approach," *Neural Networks, IEEE Transactions on*, 1997. 8(1), pp.98-113.

[11] C. Garcia, and M. Delakis, "Convolutional face finder: A neural architecture for fast and robust face detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, Nov. 2004, pp. 1408-1423.

[12] Y. N. Chen, C. C. Han, C. T. Wang, B. S. Jeng, and K. C. Fan, "The application of a convolution neural network on face and license plate detection," *in 18th Int. Conf. Pattern Recognition*, 2006.

[13] M. Szarvas, U. Sakail, and J. Ogata, "Real-time pedestrian detection using LIDAR and convolutional neural networks," *in Intelligent Vehicles Symposium, 2006 IEEE.* 2006. IEEE.

[14] M. Peemen, B. Mesman, and H. Corporaal, "Speed sign detection and recognition by convolutional neural networks," *in Proc. 8th Int. Automotive Congress. 2011.*

[15] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," *in 2013 12th Int. Conf. Document Analysis and Recognition, 2003. IEEE Computer Society.*

[16] J. Canny, "A Computational Approach to Edge Detection, " *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 679–698, Nov. 1986.