QR code detection for augmented video streaming

Xuan-Vinh Nguyen* , Anh-Dung Ho*

University of Information Technology, Ho Chi Minh City, Vietnam
Email: {18521655,18520630}@gm.uit.edu.vn*†

Abstract. Video streaming is a multimedia service that continuously transmits the data over the Internet and presents the content on user screens without predownloading the entire video. Augmented video streaming is an advanced version of video streaming, where the video is enriched with additional embedded information in video frames. These additional data aim to provide a better user experience. Using QR code is one of the efficient approaches to incorporate information into video streams in this context. However, receiving the data in the embedded QR code is considered a challenging task owing to video quality and view angles. This paper proposes a lightweight two-stage QR code decoder for augmented video streaming using deep learning technologies. In the first stage, the position of the embedded QR code is detected using an online object detection algorithm. In the second stage, the detected region of the QR code is fed into a QR code reader to extract the embedded data. The experimental results show that the proposed decoder achieves high performances in terms of response time and decoding accuracy while being very lightweight, which is promising to be implemented in smartphones.

Keywords: QR code; detection; Deep Learning; YOLO

1 Introduction

Due to the ever-increasing need for video content, videos are required to be played back or played directly at the time of the recording without downloading. To fulfill this need, media streaming has appeared and gradually dominate the entertainment industry [1–3]. Augmented video streaming a type of media streaming where streaming videos are embedded with additional data. These embedded data can be used to enhance the user experience and provide more ways for the service provider to deliver content to end users. For example, links to the website of new events or products can be embedded in the videos which introduce these events and products. These additional data can be embedded through many means including text, icon, logo, gif, audio, and QR code. Among these options, the QR code, which was proposed by Mr. DensoWave in 1994 is especially suitable for augmented video streaming as it can be used to embed large amounts of information without significantly interfering with viewer experiences. Embedded QR codes can appear in videos in two forms: active and passive.

As shown in Fig. 1a, the QR code is actively embedded in the video to provide a link to the website of a music tour. In Fig. 1b, the printed QR code which provides the location appears video scene. Active embedded QR codes can intentionally provide information to viewers in marketing, education, entertainment, healthcare, and many other fields. In the case of Fig. 1a, the viewer can scan the shown QR code to go to the music event website or buy the ticket directly. Passive embedded QR code is not used intentionally but still supplying useful information to viewers. For example, in the case of Fig. 1b, the viewer can scan the show QR code to find the location of the place in the video. To these ends, a type of streaming QR code scanner, which is an application installed in a camera enable smart device must be used. Smartphones, which are widely available nowadays, are the perfect device for installing these applications thanks to high-resolution cameras and powerful processors equipped with them.

(a) Active QR code

(b) Passive QR code

Fig. 1: QR code embedded in videos.

In augmented video streaming, QR codes are continuously embedded in frames of the video to provide a stream of additional information. The constant changes of the QR code appearing in the video will introduce a lot of difficulties for current QR code scanners, which are developed for decoding static QR codes. This study focuses on proposing a new lightweight streaming QR code scanner to obtain embedded information in Augmented Video Streaming. The proposed algorithm is developed to be not only accurate in detecting the QR but also lightweight to guarantee the possibility of implementation on smartphones.

In the proposed algorithm, the process of extracting information from the QR code embedded in the video is achieved in two steps. In the first step, the position of the QR code in the video is determined by applying object detection algorithms such as Yolov4. In the second step, the areas containing the QR code located in the previous step will be cut out from each frame and fed to a QR code reader to decode to retrieve the embedded information.

To develop the above QR code detect program, a collection of augmented videos has been created. QR code is embedded in these videos in two forms: active and passive. In the first form, QR codes are actively embedded in each video frame. In the second form, the QR code appears as a printed image in the captured videos. The augmented videos were then played in various types of displays including monitor and projector screen. A smartphone was used to record the video of the display to create the dataset for the proposed streaming QR code scanner.The performance of the proposed algorithm is evaluated through many aspects including decoding accuracy, detectable distance, and detection rate. The experimental results show that the proposed algorithm has high performance in these aspects. In addition, this algorithm is also very compact, so it can be installed on smartphones.

2 Related Works

Various methods have been proposed for the barcode recognition problem over the years. One of the first barcode identification papers was that of Muniz *et al.* [4] for a barcode scanning application on prescriptions. This early approach for QR code recognition soon became outdated by later methods.

The appearance of the camera on mobile phones has inspired many articles on barcode identification by mobile phone cameras. In [5], Ohbuchi *et al.* developed a mobile application that can detect both QR and EAN codes. In [6], Wachenfeld *et al.* proposed a method for identifying 1D barcodes, in which decoding is used as a tool to find bar codes. Both Ohbuchi and Wachenfeld are heavily dependent on the user pointing the camera at the barcode.

In recent papers, the focus has been on offering algorithms for barcode recognition with little reliance on human camera alignment. Some methods that depend on simple geometrical operations such as the paper Katona and Nyul [7] improved from their work in [8]. The improved version adds Euclidean distance to eliminate objects in the distance. This paper is one of the problems related to the recognition of both 1D and 2D barcodes. The data used for testing in this paper are 17,280 composite images and a set of 1000 real-life images with a 1D barcode. This dataset has not been published yet and the author has not tested their algorithm on any of the benchmark datasets yet. However, Sörös *et al.* [9] evaluated Katona's performance plus their own algorithms per 1000 1D images from WWU Muenster Barcode Database [10, 11]. This test shows the Katona algorithm's low score and indicates that though Katona reports high accuracy on their own datasets.

In [12], Creusot *et al.* provided a modern model for 1D barcode detection. The article uses Muenster BarcodeDB and Arte-Lab database to test model performance. Creusot's test results outperformed both Arte-Lab and Muenster BarcodeDB. In [13], the author continued for a follow-up article that improves the results of the previous article by using a method called Parallel Segment Detector (PSD) based on Line Segment Detector (LSD).

3 Proposed Two-Stage Algorithm for Streaming QR Code Decoder

The proposed two-stage algorithm for streaming QR code decoder is shown in Fig. 2. As suggested by the name, the process of extracting the information embedded in the QR code is performed through two stages. In the first stage, the position of the QR code in the frame is located using an object detection algorithm. In the second stage, the QR code is decoded to extract the hidden information, which is the link to the music event in the example shown in the figure.

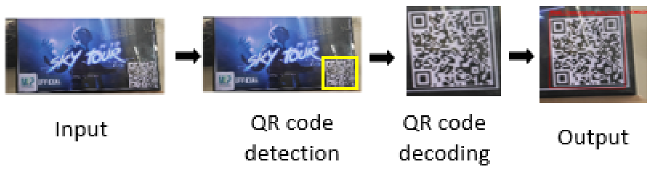


Fig. 2: Procedure of the proposed algorithm.

3.1 QR Code Detection

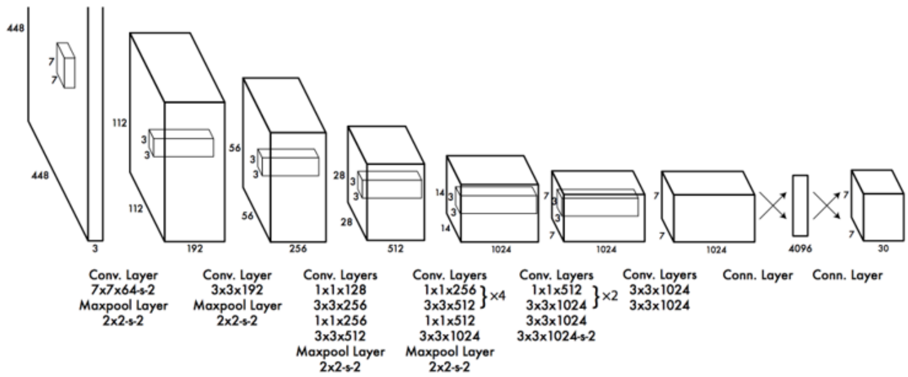


Fig. 3: YOLO architecture.

YOLO was originally introduced as the first object detection model that combined bounding box prediction and objects classification into an end-to-end distinguishable network. It is written and maintained in a framework called Darknet.

In yolov4 [14], the main new features introduced by this new version are:

1. **Bag of Freebies**: BoF is the improvement that only increases training costs to get better performance such as increased images. The authors discuss several methods and choose specific ones like CutMix or the Regular DropBlock, among others.

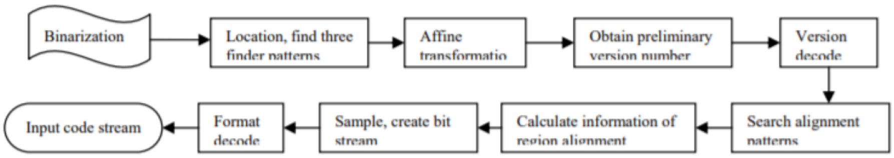


Fig. 4: The decoder flow chart of QR code.

2. **Bag of Specials**: BoS are plugin modules and post-processing methods only increase inference cost to achieve better performance, such as the SPP module. The authors, and a few others, chose Mish activation and partially cross-stage connections.
3. **Architecture neck**: The authors introduce a block in the middle of the backbone, CSPDarknet53, and the first part, YOLOv3. She is SPP (Spatial Pyramid Pooling) and PAN (Path Aggregation Network).
4. **Mosaic augmentation**: The authors also introduced a new augmentation method called Mosaic. It combines four images of the training data set into one image.

In this paper, a transfer learning with YOLO, which is illustrated in Fig. 3, is used for QR code detection. The final layer of the network is modified to output the bounding box of the QR code. All the pre-trained parameters of earlier layers are used as the initialized parameters.

3.2 QR Code Decoding

After applying QR code detection, the position of the QR code in the frame has been located. The are of the QR code is cropped and fed into a QR code decoder. In this method, Pyzbar, which is a library available at `https://pypi.org/project/pyzbar/`, is used for this task.

The structure features of QR code symbol must be adequately used in its decoder procedure. The flow chart of QR code recognition is shown as Fig. 4. The main steps in the procedure of the decoding algorithm include:

- Binarization;
- Obtain the approximate region of QR code, and implement coarse positioning for QR image according to the finder patterns;
- Implement accurate positioning according to the alignment patterns;
- Calculate the angle of inclination to rotate QR image, and implement rectification processing;
- Obtain version number and implement self-adaptive sampling;
- Decode based on corrected image and input a standard 2D matrix.

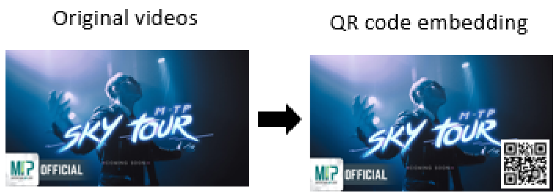


Fig. 5: Active augmented video generation.

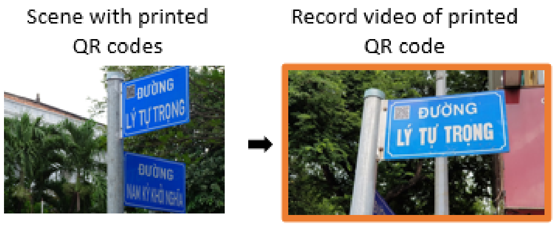
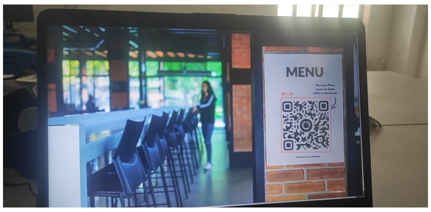
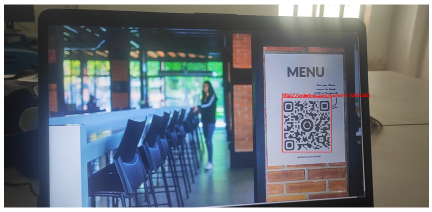


Fig. 6: Passive augmented video generation.



(a) Detect QR code



(b) Decode QR code

Fig. 7: An experimental example of QR detection and decoding.

4 Experimental Results

4.1 Dataset Preparation

To prepare the dataset for the proposed streaming QR code decoder, a collection of augmented videos need to be created. As mentioned earlier, this paper deals with two types of augmented video: active and passive. The active augmented video generation is shown in Fig. 5. Firstly, videos of various topics including entertainment, advertising, education, technology are collected. QR codes are then inserted into frames of these videos to embed information such as links, text, locations. Note that the embedded QR code change through frames of each video to continuously convey different information. The size of the QR code also varied to evaluate the effect of QR code size on the performance of the proposed algorithm. The procedure of making passive augmented videos are shown in Fig. 6. Scenes with printed QR codes were found and recorded. The distance from the camera to the printed QR code varies so that the effect of distance on the performance of the proposed algorithm can be evaluated.Upon obtaining augmented videos, the dataset for training and testing the proposed algorithm was created.

Firstly, the original augmented video was played on displays including computer monitors and projector scenes. Then smartphone cameras were used to record the video of these displays. These videos are cut into frames. Those frames where QR codes appear are labeled to create the dataset for training.

A total of 68 videos has been recorded using the rear camera on iPhone. We then separate the frames to proceed with labeling the dataset. The total number of photos we extracted from the videos for labeling was 3,388. We also divided the total number of tagged photos into two pieces of training and validations with the number of photos for each episode of 2779 and 609 respectively. Finally, we created a test set with 574 images to evaluate the model.

4.2 Training

Model parameters to train for QR code recognition: class = 1, max batches = 6000, steps = 4800, 5400 and width = 416, height = 416. The transfer learning model based on yolov4 was trained on google colab pro. Initially, the yolov4.conv.137 weight file, which was created by the authors of the darknet is trained with the dataset created in this paper. At the iteration of 1600, the loss of the model was shown to be stable and thus the training was thus decided to stop. The weight file at this iteration is used for the second training. In the second training, the model was trained with 6000 iterations in 16 hours.

4.3 Experimental Results

Figure 7a show the result of the detection step. It can be seen that the QR code in the streaming video can be detected at high accuracy. A test set with 574 images is used to evaluate the proposed streaming QR code decoding algorithm. The accuracy of the detection stage is evaluated through IOU (Intersection over union) and Average precision (AP). Usually, with object detection problems, mAP is the main index used for evaluation. However, since there is only one type of object, that is QR code, in this problem, the detection accuracy is evaluated through AP. The AP of the QR code detection step is displayed in Fig. 8. It can be seen that the detection part works well at 99.93 % AP with threshold IoU $>$ 0.5. This is because since QR codes all have a reasonable size to have a decent possibility of being successfully decoded.

An example of the result of the decoding stage is shown in Fig. 7b. As mentioned earlier, the performance of the proposed algorithm is affected by the size of the QR code and the distance from the camera to the QR code. When the QR code is too small or the distance is too far, the QR code cannot be decoded even though it still can be detected. Figure 9 show the limitation of the proposed algorithm. More specifically, the figure shows the detectable distance corresponding to the size of the QR code to achieve a decoding rate of at least 90%.

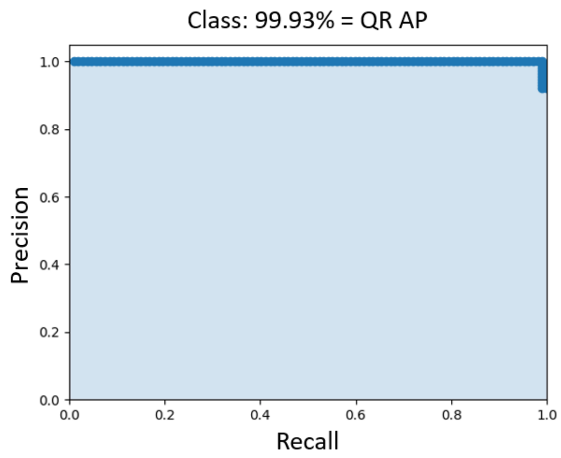


Fig. 8: Average precision.

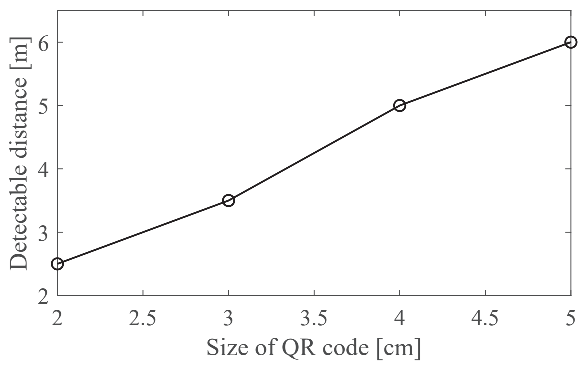
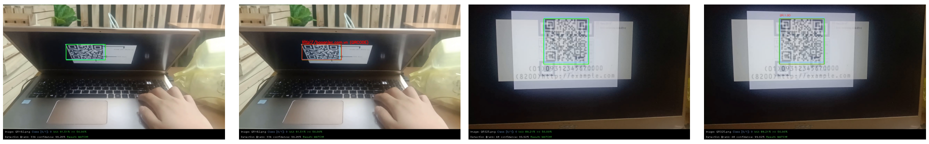


Fig. 9: Decoding efficiency.

In general, distance and size are the deciding factors for QR code decoding. As shown in Fig. 10 (a) and (b), even when the augmented video is viewed at a small angle, the QR code still can be detected and decoded successfully. Besides distance and size, the quality of the QR code images that appear on the smartphone camera also greatly affects the possibility of decoding. The QR code in the video shown in Fig. 10 (c) is large and viewed at a straight angle. However, because of the light reflecting on the monitor, the QR code does not appear clearly in the smartphone camera. Therefore, even though being detectable, it cannot be decoded as shown in Fig. 10(d).



(a) Detect QR code (b) Decode QR code (c) Detect QR code (d) Decode QR code

Fig. 10: Difficult cases with successful results.

5 Conclusion

In augmented video streaming, QR codes are embedded on frames of videos streamed over the Internet. Thanks to the use of QR codes, various types of information can be intentionally embedded into videos for advertisement, education, and other applications. Since QR codes are inserted continuously into frames in the video, existing QR code decoders, which are developed for decoding static QR code, cannot be used. This paper deal with the problem of detecting and extracting information of QR code in augmented video streaming. More specifically, this paper proposed a two-stage algorithm for streaming QR code decoding. In the first stage, transfer learning with Yolov4 is adopted to detect the QR code in the frame. In the second stage, the Pyzbar library is used for decoding the QR code in the area detected in the previous stage. Experimental results on active and passive augmented videos show a good performance of the proposed algorithm. More specifically, QR codes in the tested videos were detected with an average precision (AP) of up to 99.93 %. QR code is shown to be detected and decoded successfully even when the displays which present the augmented videos are viewed at small angles. QR codes in approximately 90 % of tested videos were successfully detected. The QR codes in the remaining 10 % cannot be decoded due to the blurriness or small size of the QR code in the augmented videos. Overall, experimental results show that the proposed algorithm works well on live augmented video. At the same time, the proposed algorithm is also very lightweight, which ensures the implementation on smartphones.

References

1. A.-T. Tran, N.-N. Dao, and S. Cho, "Bitrate adaptation for video streaming services in edge caching systems," *IEEE Access*, vol. 8, pp. 135 844–135 852, 2020.
2. N.-N. Dao, D. T. Ngo, N.-T. Dinh, T. V. Phan, N. D. Vo, S. Cho, and T. Braun, "Hit ratio and content quality tradeoff for adaptive bitrate streaming in edge caching systems," *IEEE Systems Journal*, 2020.
3. A.-T. Tran, T.-V. Nguyen, V.-D. Tuong, N.-N. Dao, and S. Cho, "On stalling minimization of adaptive bitrate video services in edge caching systems," in *IEEE International Conference on Information Networking (ICOIN)*. IEEE, 2020, pp. 115–116.
4. R. Muniz, L. Junco, and A. Otero, "A robust software barcode reader using the Hough transform," in *IEEE International Conference on Information Intelligence and Systems (Cat. No. PR00446)*, 1999, pp. 313–319.

5. E. Ohbuchi, H. Hanaizumi, and L. A. Hock, “Barcode readers using the camera device in mobile phones,” in *IEEE International conference on cyberworlds*, 2004, pp. 260–265.
6. S. Wachenfeld, S. Terlunen, and X. Jiang, “Robust recognition of 1-D barcodes using camera phones,” in *IEEE International Conference on Pattern Recognition*, 2008, pp. 1–4.
7. M. Katona and L. G. Nyúl, “Efficient 1D and 2D barcode detection using mathematical morphology,” in *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, 2013, pp. 464–475.
8. ——, “A novel method for accurate and efficient barcode detection with morphological operations,” in *IEEE International Conference on Signal Image Technology and Internet Based Systems*, 2012, pp. 307–314.
9. G. Sörös and C. Flörkemeier, “Blur-resistant joint 1D and 2D barcode localization for smartphones,” in *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*, 2013, pp. 1–8.
10. O. Gallo and R. Manduchi, “Reading 1D barcodes with mobile phones using deformable templates,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1834–1843, 2010.
11. E. Tekin and J. Coughlan, “BLaDE: Barcode localization and decoding engine,” *Tech. Rep. 2012-RERC. 01*, 2012.
12. C. Creusot and A. Munawar, “Real-time barcode detection in the wild,” in *IEEE winter conference on applications of computer vision*, 2015, pp. 239–245.
13. ——, “Low-computation egocentric barcode detector for the blind,” in *IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 2856–2860.
14. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020.