Scanning QR codes for object detection based on Yolo-v7 algorithm and deblurring generative adversarial network

Huan Chen[1] Hsin-Yao Hsu[1] Kuan-Ting Lin[1] Jia-You Hsieh [1]

Yi-Feng Chang [1] Bo-Chao Cheng [2]

[1] National Chung Hsing University, Taichung City 40227, Taiwan
[2] National Chung-Cheng University, Chia-Yi 621301, Taiwan
{ huan@nchu.edu.tw, 8109056003@smail.nchu.edu.tw,
Hups900192@gmail.com, roger.hs@smail.nchu.edu.tw,
azure01125@gmail.com, bcheng@ccu.edu.tw}

Abstract. Location-based advertising (LBA) has been popular for several years, and the amount of global investment is increasing year by year. Nowadays, in the vigorous development of vehicle vision systems, many recognition tasks can be completed by combining You Only Look Once version 7 (Yolo-v7) object detection algorithms to apply automotive applications, and also involve a QR codes decoding method with deblurring generative adversarial network version 2(DeblurGAN-v2), which can capture the QR codes set on the route in real-time to obtain the LBA placed by the merchant, the results show that the proposed method outperforms the other object detection model and deblurring model, it obtains more efficient for scanning QR codes.

Keywords: Location-based advertising, QR codes, Object detection, Deblurring generative adversarial network.

1 Introduction

LBA is a targeted advertising approach that delivers ads based on local cultural practices, ranging from static advertising signs on the roadside to mobile devices. According to [1], it shows that LBA traffic has a higher value than the application filed and investment amounts are expanding year by year. Moreover, the global information market research [2] shows that the market size of LBA is based on geographic positioning.

Nowadays, the value of LBA reveals in the area of mobile service. Yu et al. [3] revealed that the roadside for mobile servers from providers by placing targeted advertisements along the itinerary between the point of passenger boarding and their destination, advertisers can increase their revenue by leveraging local cultural characteristics. However, the service model requires pre-setting the driving route and placing advertisements along the road, which limits the advertisement placement to the designated route.

The demand for systems that support artificial intelligence (AI) increases, it revealed the advanced driver assistance systems (ADAS), which are very popular and the trend of the future. The report shows in [4] from the market size was growing at a compound annual growth rate (CAGR) of 13.8%. The literature in [5][6] used computer vision for deep learning architecture for the competition of driving needs.

The problem in ADAS system for LBA communication technology, including global positioning system (GPS), Wi-Fi, cellular tower pings, QR code, and radio frequency identification (RFID), it is important that the driving assistance for received the advertisement to consider the detection of higher accuracy, signal strength, and cost, especially using the QR code scanning are confirm to real conditions. The motivation of this research is the challenge of QR code, which is revealed to get more clearly for image collection, rapid for scanning, and error correction [7]. Li et al. [8] points out that the recognition of QR codes may be affected by some motion or focus blur, which makes the uneven road surface may cause horizontal and vertical motion blur within different speeds and road conditions during the driving process.

To address this problem, this study used a vehicle called a donkey car and combine it with raspberry pi and a high-speed camera to capture video on its own. The process of recording during filming is to capture video footage that involves the automotive application, applying different Yolo-v7 algorithms [9] to split training and testing set, which is compared with various speeds, sizes, and angles. Furthermore, the restoring of real pictures for scanning indeed, using end-to-end generative adversarial network (GAN) [10] for single image motion deblurring, named DeblurGAN-v2 [11] to restore the blurred image, and evaluate the proportion of successful scans that can be restored, and also compares the performance to different models, so that can be achieved the proposed method outperforms the other methods. The training and testing set in the dataset are characterized by their detailed information. The former was acquired through a self-propelled device indoors, while the latter was obtained outdoors using the same device.

The remainder of this paper is organized as follows. Section 2 discusses the related work on deep learning for QR codes scanning based on driving assistance and deblur methods. Section 3 describes the device and the proposed deep learning approaches based on architectures. Section 4 presents the experimental results and discussion for evaluation and comparison with different methods. Finally, Section 5 provides the conclusions and future work.

2 Related Work

To improve the QR code scanning rate, the QR code for object detection is blurred to increase the reading rate for more discussion. Yuan et al. [12] proposed that neural networks for training such as linear motion, defocus, and Gaussian blur can be distinguished after training. Schuler et al. [13] demonstrated that the use of a neural network-based approach is superior to traditional methods for non-blind image deblurring, particularly in cases of motion blur. Nah et al. [14] proposed Deep Deblur, which is a multi-scale convolutional neural network (MCNN) that cancels the kernel. Mechanism to remove the limitation by the blur kernel, thereby avoiding artifacts. Inspired by Goodfellow et al. [10] proposed GAN image-to-image translation, Kupyn et al. [15] regarded the blurring problem as an image-to-image translation task, and used conditional GAN (cGAN) network structure and loss function to evaluate the generated clear image. The gap between the image and the ground truth, DeblurGAN was proposed, and the best deblurring effect was obtained. To obtain better image quality, which uses the Feature Pyramid Network (FPN) network for feature fusion, and the discriminator uses the loss function of Least Squares GANs (LSGAN) for the overall training process to get more stable.

To remove noise and perform binarization before it is used as input, which has a square shape and functional symbols to capture obvious features of QR code, [16] [17] [18] believe that the task of identifying QR code has a great relationship with the location of Finder Pattern (FIP). Blanger et al. [18] adjusted the Single Shot MultiBox Detector (SSD) architecture in training, and added FIP as a sub-part feature for training to affect the output results, to improve the accuracy of QR code detection. Wang et al. [19] and others proposed a deblurring method under the condition of motion blur. The GAN-based method was used to obtain the deblurred two-dimensional code image, and compared with the traditional method, it was proved that the recognition accuracy and speed of the convolutional neural network (CNN) method were outperformed.

3 Methodology

3.1 Data collection

In this study, Raspberry Pi was utilized to install vehicle modules for collecting driving QR code images. The following four criteria were used as the basis for collecting and designing the dataset. The first criterion for collecting, labeling, and designing the dataset is the size of the QR code, which determines the amount of space it occupies on the roadside sign. Moreover, it affects the resolution of the in-vehicle camera, especially for smaller QR code sizes, where a higher-resolution lens is required for capturing. The second criterion for collecting and designing the dataset is the driving speed, which can cause shaking and blurring. In this study, the urban speed limit of 50km/hr was used, and the scale was adjusted based on the size of the QR code. Specifically, for a road surface with a length and width of 1.2 meters, and a display driving QR code size of 12cm, the driving speed was reduced to 5km/hr to minimize shaking and blurring. The third criterion is the distance between the QR code and the vehicle. As the camera on the vehicle may have height limitations, images in the inner lane may be blocked by vehicles in the outer lane. To address this, the environment assumes that the vehicle is driving in the outer lane or a single lane, and the position of the QR code is scaled according to the actual road shoulder width plus the width of the sidewalk. The fourth criterion pertains to the shooting distance, as it is crucial to capture a clear and detailed image of the QR Code. During the data collection process, the initial detection of the QR Code is set at 10 meters from the vehicle on the road surface. However, driving speed becomes faster, the QR Code will be early detected at 20 meters. The zoom distance of the vehicle is adjusted based on the scale of the QR code. The following are designed is shown as in **Fig. 1**.

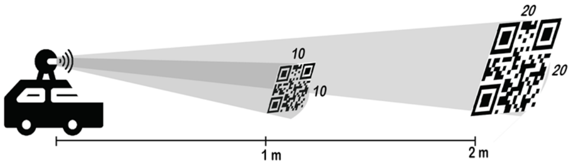
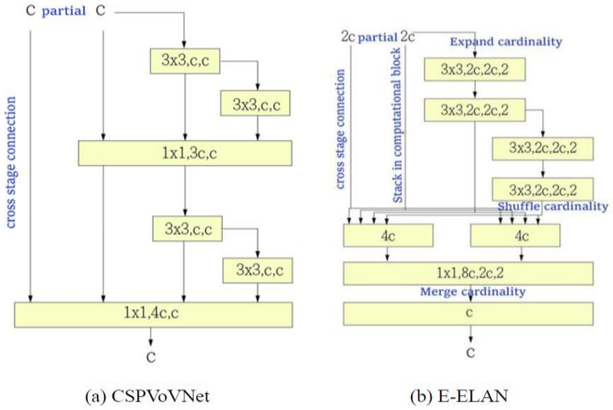


Fig. 1. An illustration of QR code size and distance ratio

The data collection was in both indoor and outdoor locations, with images captured every 2 seconds. To analyze the impact of blur restoration at different speeds, it is range from 15 km/hr to 50km/hr based on the hourly speed.

3.2 Yolo-v7 architecture

Real-time object detection requires a faster training speed. To meet this requirement, it is necessary to adopt a model that can achieve a frame rate of over 30 frames per second(fps). The YOLO [20] series offers several versions of object detection algorithms that have demonstrated high execution speeds and accuracy. YOLOv7 [9] employs advanced optimization methods to enhance the model architecture. It combines the original VoVNet [21] architecture with Cross Stage Partial Network (CSPNet) and also improves the gradient path to CSPVoVNet [22] so that the model can learn the weights of different layers more effectively, speed up training and improve accuracy. The model has improved and extended based on Extended Efficient Layer Aggregation Networks (E-ELAN) [23], which stabilizes learning and convergence through methods such as shuffling cardinality, expand cardinality, and merge cardinality, and avoids excessive computational blocks leading to unstable states. A structure is shown in **Fig. 2**.



(a) CSPVoVNet (b) E-ELAN

Fig. 2. Extended efficient layer aggregation networks[9]

To address the issue of transition layer width in deep neural networks is through a compound model scaling method. The approach involves scaling the concatenation-based model while maintaining the original nature of the model when the depth of the computational block is scaled [24]. The scaling process also takes into consideration the scaling of the transition layer, which ensures that the width of the transition layer is adjusted proportionally to the scaling of the computational block. The depth of the computational block should be scaled, while the corresponding width can be adjusted through the transition. It allows to maintain the inherent characteristics of the initial model during the design phase and preserve the optimal structure, while also avoiding the issue of reduced computational utilization. Overall, this method aims to optimize the performance of deep neural networks by preserving their original architecture and characteristics while scaling them to handle more complex tasks. A structure is shown in **Fig. 3**.

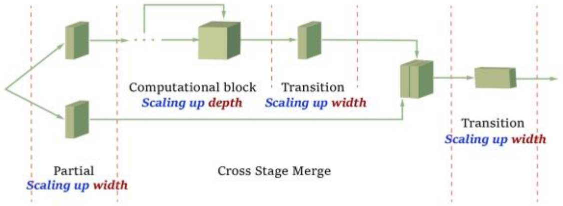


Fig. 3. A compound scaling up depth and width for concatenation-based model[24]

A structural re-parameterization technique with Visual Geometry Group called RepVGG[25], optimizes various indicators such as floating point operations per second (FLOPS), accuracy, and speed through re-parameterization. The RepVGG technique destroys the residual in Residual Neural Networks (ResNet) [26]and connection in Densely Connected Convolutional Networks (DenseNet) [27], to improve their performance of more diversity of gradients for different feature maps. The RepVGG identity connection (ResConvN)is combined with the ResNet shortcut connection. As a solution, YOLOv7 proposes to use RepVGG in ResNet (RepResNet) to remove the identity connection and improve the new architecture. Ultimately, the architecture aims to optimize the accuracy of deep neural networks while reducing the complexity of the architecture [9], it used the planned re-parameterized model and auxiliary head with independent label assignment strategy.

3.3 DeblurGAN-v2 architecture

GAN [10] is a type of deep learning architecture that consists of two models: a generator and a discriminator. The generator produces fake samples, which are then compared to real input samples during the training process. The discriminator is trained to distinguish between the real and fake samples and adjust its weights, the difference is used to update the weights of the discriminator and improve its ability to distinguish.

During the training process, the generator is used to create synthetic data, and the discriminator is used to identify whether the generated data is real or fake. The process is designed to optimize the generator's ability to create realistic data that can fool the discriminator.

In the training process of a GAN model, two models are used for evaluation. The first is the generator network, which takes in noise data represented as $\boldsymbol{z}$ and generates synthesized data represented as $G(\boldsymbol{z})$, through the training process. The second model is the discriminator network, which evaluates the difference between the synthesized data and the real data represented as x and computes a probability score. The function $V(D, G)$ [10] is used to model the interaction between the generator and discriminator networks and is defined in the formula as follows:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_{data}(x)}[log\ D(x)] + E_{x\sim p_z(x)}[log(1-D(G(\boldsymbol{z})))] \qquad (1)$$

Assume p_z be the data distribution of noise input $\boldsymbol{z}$ and p_{data} be the data distribution of real samples. The function E represents the empirical estimation of the joint probability distribution. In the decision process of the discriminative network, the maximum value of $E_{x\sim p_{data}(x)}[log\ D(x)]$ to get the fake data $G(\boldsymbol{z})$ and the expected probability of output is $p_z(x)$close to the real sample x, which is obtained by $D(G(\boldsymbol{z}))$. Then, the minimization of the fake data probability is obtained by $E_{x\sim p_z(x)}[log(1-D(G(\boldsymbol{z})))$.

To fix the vanishing gradients and stabilize the training, using the L2-regularized called Least Squares GANs discriminator (LSGAN) [28] to introduce a loss function can fix the vanishing gradients and stabilize training, and provide smoother and unsaturated gradients. The further away the fake samples are from the boundary, receive the greater penalties. By minimizing the Pearson $\mathcal{X}^2$ divergence in the loss function that leads to the better training stability can be achieved. The formula is shown as follows:

$$\min_{D} V(D) = \frac{1}{2}E_{x\sim p_{data}(x)}[\,(D(x)-1)^2] + \frac{1}{2}E_{x\sim p_z(x)}[D(G(x))^2] \qquad (2)$$

$$\min_{G} V(G) = \frac{1}{2}E_{x\sim p_z(x)}[D(G(x) - 1)^2] \qquad (3)$$

The relevant GAN called Double-Scale Relativistic GAN Least Square (RaGAN-LS) for upgrades in DeblurGAN-v2, will be getting the image to become more qualifier and clear, it adopted the relativistic wrapping [29] on the LSGAN cost function, the formula is shown as follows:

$$\begin{aligned} L_D^{RaLSGAN} &= E_{x\sim p_{data}(x)}[D(x) - E_{z\sim p_z(x)}D\big(G(z)\big) - 1)^2] \\ &\quad + E_{z\sim p_z(x)}[D(G(z)) - E_{x\sim p_{data}(x)}D(x) + 1)^2] \qquad (4) \end{aligned}$$

A relativistic discriminator to estimate the probability that a given real data is more realistic than randomly sampled fake data shows more stable and computationally efficient training.

4 Experimental results and discussion

4.1 Experimental setup

The experimental setup for training was conducted on Google Colab, utilizing the NVIDIA T4 Tensor Core GPU as the primary computing device. In the software implementation, the Python programming language was utilized, with the deep learning framework being implemented through the PyTorch package, and also involve the Pyzbar package to make a barcode reader for decoding the QR codes. In addition, the captured QR codes images were recorded based on the homemade donkey car [30] with data collector and raspberry pi 4 Model B as shown in **Fig. 4**. A Raspberry Pi Camera Module v2 is used in the camera module part in the hardware implementation.

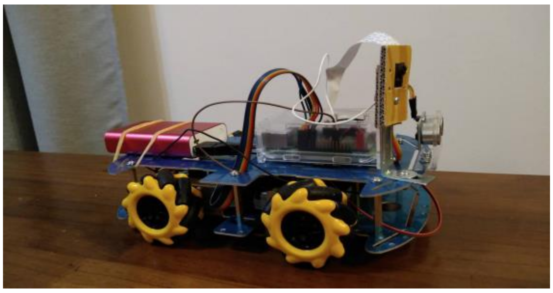
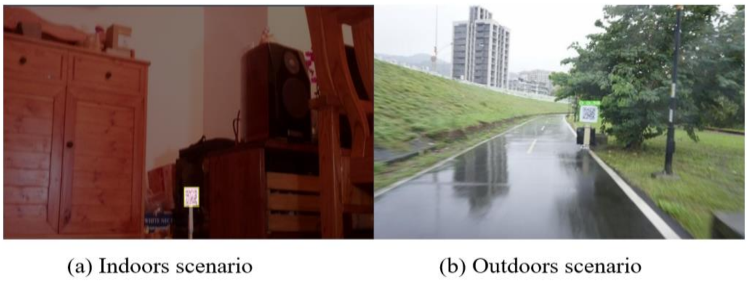


Fig. 4. Homemade donkey car with data collector and Raspberry Pi

The dataset used in this research is derived from the QR codes image dataset, as described in Section 3. It comprises a total of 1099 QR codes being annotated and captured by mounted cameras on donkey carts, which was divided into 779 samples of training set and 320 samples of testing set. The shooting scenes for the study were captured in both indoor and outdoor settings. The indoor shots were taken in a room, while the outdoor shots were taken beside the bike lane in Dahan River Riverside Park, as shown in **Fig.5**, with QR codes being available in three sizes: 10cm x 10cm, 15cm x 15cm, and 20cm x 20cm.



(a) Indoors scenario

(b) Outdoors scenario

Fig.5. Indoors and outdoors scenario

4.2 Parameters settings and evaluation metrics

The input of each YOLO model for parameters settings are shown in **Table1**. Backpropagation algorithm was employed in conjunction with Adam optimizer [34] for gradient descent during the training process. The models were trained for a total of 50 epochs, with a batch size of 8 samples used for each epoch. The learning rate was set to 0.01 to facilitate the convergence of the training process. The only exception is for the You Only Look Once version 4 tiny (YOLOv4-tiny) [31] model, which was trained for 60 epochs to ensure optimal performance. Some of the models used Complete-IOU [35], and the others used Focal loss [36] for loss function settings. This study utilizes an evaluation framework to assess the performance of the model, which was divided into two stages. The first stage involves QR code label tracking, with evaluation metrics including Precision, Recall, F1 score, and Intersection over Union (IOU). The second stage consists in deblurring QR code labels, which used the pretrained model of Deblur-GAN-v2 with evaluation including success and failure for scanning the number of QR code images.

Table 1. Parameters settings for each Yolo model

Model	Epochs	Optimizer	Learning rate	Batch size	Loss function
YOLOv4	50	Adam	0.01	8	Complete-IoU
YOLOv4-tiny	60	Adam	0.01	8	Complete-IoU
YOLOv7	50	Adam	0.01	8	Complete-IoU
YOLOv7-tiny	50	Adam	0.01	8	Focal loss
YOLOv7-W6	50	Adam	0.01	8	Focal loss

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{7}$$

$$IOU = \frac{\text{Object} \cap \text{Detected box}}{\text{Object} \cup \text{Detected box}} \tag{8}$$

where TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative, and the object refers to the area of the actual object, while the detected box refers to the predicted candidate area by dividing the overlap area by the union area [32].

4.3 Compared approaches

In this study, several metrics were compared to the results obtained that the training model and verify the training dataset on indoor data. The data collected from outdoor scenes were used as a testing dataset to evaluate the performance of various YOLO models as shown in **Table 2**, including YOLOv4, YOLOv4-tiny, YOLOv7, YOLOv7-tiny [9], and a larger YOLOv7 model called YOLOv7-W6 [9] [33] were evaluated. The main comparison criteria were the detection accuracy and efficiency of models. The results showed that the YOLOv4 model exhibited good detection performance. The YOLOv4-tiny model decreased the Precision by 0.03, F1-score by 0.02 and IOU by 0.021. The YOLOv7 model shows the results that the performance has a significant increase, which is better than YOLOv4 and YOLOv4-tiny. YOLOv7-tiny model obtains a decrease by only Precision by 0.01 and IOU by 0.021. The YOLOv7-W6 model achieved the best performance among all models, with a high value for a Precision of 0.97 and an IOU of 0.8611. Additionally, all properties of different YOLOv7 models are better than different YOLOv4 models, and the value for F1-score is almost 0.99.

The experiment results for the deblurring models which are pre-trained models for prediction of testing set, and scanning from a barcode reader with comparing as shown in **Table 3**, the donkey car for driving is about 0-25km/h speed range, 212 images can be compared with different models, including without deblurring, Deep Deblur [14], DeblurGAN [15], DeblurGAN-v2 with the plugin of sophisticated backbones are MobileNet and Inception-ResNet-v2 [11]. It is shown that the QR code image without deblurring has a higher rate of failed scanning, it shows that more blurring image it cannot scan more clearly, and the use Deep Deblur model shows that the performance of successful scanning has more QR code image than without scanning. Unfortunately, it has failed to scan and it is worst. Moreover, the DeblurGAN model shows that it is equal to successful and fails to scan to Deep Deblur. Otherwise, the DeblurGAN-v2(Inception-ResNet-v2) returns to the source and read QR code successful scanning is grow up to 89 images, it has a significant increase and is better than others. Nevertheless, it has to be improved for more successful scanning and less failure.

Table 2. A comparison of QR code detection for the results.

Model	Precision	Recall	F1-score	IOU
YOLOv4	0.95	1	0.93	0.836
YOLOv4-tiny	0.92	1	0.91	0.815
YOLOv7	0.95	1	**0.99**	0.857
YOLOv7-tiny	0.94	1	**0.99**	0.836
YOLOv7-W6	**0.97**	1	**0.99**	**0.861**

Table 3. A comparison of QR code deblurring and scanning for the results.

Model	Success	Fail
None	10	202
Deep Deblur	21	191
DeblurGAN	21	191
DeblurGAN-v2 (MobileNet)	31	181
DeblurGAN-v2 (Inception-ResNet-v2)	**89**	**123**

5 Conclusion

This research presents a base for an in-depth learning method combined with Donkey Car for high-speed vehicle reading QR codes efficient structure. The purpose of the research using the model for object detection relies on the detection of QR codes that are strategically placed in low-density areas of the image. These codes are then processed using an image enhancement technique to reduce reconciliation errors and missing QR code images. As a result, the QR code reading success rate is significantly increased, leading to more accurate object detection. In the limitation of this research have the challenge of collecting data for each size, lightness, number of roadside and roadside width, etc. In future works, it can be tried out to add data argumentation and attention mechanism into the training process of detecting QR codes and develop corresponding image processing techniques to enhance the scanning rate of these codes. By addressing these challenges, the accuracy and efficiency of the object detection model can be improved.

Acknowledgements

This work was supported in part by the National Science and Technology Council (NSTC) of Taiwan, R.O.C., under Contract NSTC-110-2221-E-005-032-MY3, NSTC-111-2218-E-005-007-MBK and Qualcomm’s UR program support.

References

1. Dhar, Subhankar, and Upkar Varshney.: Challenges and business models for mobile location-based services and advertising. Communications of the ACM 54(5), 121-128(2011).
2. Global Location-Based Advertising Market 2021-2025. https://www.grandviewresearch.com/industry-analysis/laboratory-informatics-market, last accessed 2023/06/08

3. LIU, Chun; WANG, Zhong-qi. The research on advertising model of self-driving car platform. In: 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC). IEEE, pp. 95-99(2017).
4. Advanced Driver Assistance Systems (ADAS) Market - Global Industry Analysis, Size, Share, Growth, Trends, Regional Outlook, and Forecast from 2023 to 2032. https://www.precedenceresearch.com/advanced-driver-assistance-systems-market, last accessed 2023/06/08
5. Boukerche, Azzedine, and Zhijun Hou.: Object detection using deep learning methods in traffic scenarios. ACM Computing Surveys (CSUR) 54(2), 1-35(2021).
6. Arcos-García, Álvaro, Juan A. Álvarez-García, and Luis M. Soria-Morillo.: Evaluation of deep neural networks for traffic sign detection systems. Neurocomputing 316, 332-344(2018).
7. Hakimpour, Farshad, and A. Zare Zardiny.: Location based service in indoor environment using quick response code technology. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 40(2), 137(2014).
8. Li, Junnian, et al.: A motion blur QR code identification algorithm based on feature extracting and improved adaptive thresholding. Neurocomputing 493, 351-361(2022).
9. WANG, Chien-Yao; BOCHKOVSKIY, Alexey; LIAO, Hong-Yuan Mark. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7464-7475(2023).
10. Goodfellow, Ian, et al.: Generative adversarial networks. Communications of the ACM 63(11), 139-144(2020).
11. KUPYN, Orest, et al. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 8878-8887(2019).
12. Yuan, Quan, et al.: Blind motion deblurring with cycle generative adversarial networks. The Visual Computer 36, 1591-1601(2020).
13. Schuler, Christian J., et al.: Learning to deblur. IEEE transactions on pattern analysis and machine intelligence 38(7), 1439-1451(2015).
14. NAH, Seungjun; HYUN KIM, Tae; MU LEE, Kyoung. Deep multi-scale convolutional neural network for dynamic scene deblurring. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3883-3891(2017).
15. KUPYN, Orest, et al. Deblurgan: Blind motion deblurring using conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8183-8192(2018).
16. HUSSAIN, Nouredine; FINELLI, Christopher. KP-YOLO: a modification of YOLO algorithm for the keypoint-based detection of QR codes. In: Artificial Neural Networks in Pattern Recognition: 9th IAPR TC3 Workshop, ANNPR 2020, Winterthur, Switzerland, September 2–4, 2020, Proceedings 9, pp. 211-222. Springer International Publishing (2020).
17. PENG, Jinbo; YUAN, Song; YUAN, Xin. QR Code Detection with Faster-RCNN Based on FPN. In: *Artificial Intelligence and Security:* 6th International Conference, ICAIS 2020, Hohhot, China, July 17–20, 2020, Proceedings, Part I 6, pp. 434-443. Springer International Publishing (2020).
18. BLANGER, Leonardo; HIRATA, Nina ST. An evaluation of deep learning techniques for QR code detection. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 1625-1629. IEEE (2019).

19. WANG, Benzhi, et al. Motion deblur of QR code based on generative adversative network. In: Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence, pp. 166-170(2019).
20. REDMON, Joseph, et al. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788(2016).
21. LEE, Youngwan, et al. An energy and GPU-computation efficient backbone network for real-time object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pp. 0-0(2019).
22. WANG, Chien-Yao; BOCHKOVSKIY, Alexey; LIAO, Hong-Yuan Mark. Scaled-yolov4: Scaling cross stage partial network. In: Proceedings of the IEEE/cvf conference on computer vision and pattern recognition, pp. 13029-13038(2021).
23. Wang, Chien-Yao, Hong-Yuan Mark Liao, and I-Hau Yeh.: Designing Network Design Strategies Through Gradient Path Analysis. arXiv preprint arXiv:2211.04800(2022).
24. WANG, Chien-Yao, et al. CSPNet: A new backbone that can enhance learning capability of CNN. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pp. 390-391(2020).
25. DING, Xiaohan, et al. Repvgg: Making vgg-style convnets great again. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 13733-13742 (2021).
26. HE, Kaiming, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778(2016).
27. HUANG, Gao, et al. Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700-4708(2017).
28. MAO, Xudong, et al. Least squares generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision, pp. 2794-2802(2017).
29. Jolicoeur-Martineau, Alexia. The relativistic discriminator: a key element missing from standard GAN. arXiv preprint arXiv:1807.00734 (2018).
30. Donkeycar: a python self-driving library. https://github.com/topics/donkeycar, last accessed 2023/05/11.
31. JIANG, Zicong, et al. Real-time object detection method based on improved YOLOv4-tiny. arXiv preprint arXiv:2011.04244 (2020).
32. Kocakanat, Kaan, and Tacha Serif.: Turkish Traffic Sign Recognition: Comparison of Training Step Numbers and Lighting Conditions. Avrupa Bilim ve Teknoloji Dergisi .28, 1469-1475(2021).
33. Jernbäcker, Axel.: Kalman filters as an enhancement to object tracking using YOLOv7(2022).
34. Kingma, Diederik P., and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).
35. ZHENG, Zhaohui, et al. Distance-IoU loss: Faster and better learning for bounding box regression. In: Proceedings of the AAAI conference on artificial intelligence, pp. 12993-13000(2020).
36. LIN, Tsung-Yi, et al. Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp. 2980-2988(2017).