

Decode Barcode Images with YOLOv8 and REAL-ESRGAN

¹Van Tuan-Tran, ²Mong-Fong Horng, ³Chin-Shiuh Shieh

¹Department of Electronics and Engineering,

¹National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan,

*¹E-mail: f111169109@nkust.edu.tw

²Department of Electronics and Engineering,

²National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan,

*²E-mail: csshie@nkust.edu.tw

³Department of Electronics and Engineering,

³National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan,

*³E-mail: mfhorng@nkust.edu.tw

Abstract. Barcodes have long been an indispensable part of modern trade and logistics, allowing for the effective tracking and identification of products and packages for a long time and have become a popular standard for production management. , but in industrial factories, depending on superior requirements, we can create an integrated barcode reader combined with other functions using the resources available with their cameras. This study explores a new approach to decoding barcode images, leveraging the capabilities of YOLOv8 [18], an advanced object detection model and REAL-ESRGAN [19], a-state-of-the-art image processing method for super-resolution image. The main objective of this study is to demonstrate the feasibility and effectiveness of using YOLOv8 to locate and extract barcodes from complex scenes and the REAL-ESRGAN method to improve barcodes image, increasing successful decoding accuracy and finally conducting a comparative survey of super-resolution methods applied with barcode images. This summary serves as the basis for an evaluation study of super-resolution methods, with potential implications for enhancing barcode-based systems in various real-world scenarios.

Keywords: *OpenCV2, YOLOv8, REAL-ESRGAN, Pyzbar*

1 Introduction

Barcodes have long served as the unsung heroes of modern commerce and logistics, silently orchestrating the seamless flow of products through supply chains, aiding in inventory management, and facilitating swift transactions at the point of sale. From the cashier scanning groceries at the local supermarket to the logistics professional tracking a shipment halfway across the globe, the ubiquitous presence of barcodes has revolutionized the way we interact with the world of goods and services. Yet, despite their ubiquity, decoding barcodes, especially in diverse and complex real-world scenarios, remains a persistent challenge. The traditional methods for decoding barcodes, while reliable in controlled environments, often falter when faced with the unpredictability of the physical world. Lighting conditions, image quality, perspective distortions, and the presence of other objects in the field of view can all conspire to make barcode recognition a formidable task. Furthermore, as we venture into an era increasingly characterized by high-resolution imaging devices, the expectation for barcode decoding accuracy and speed escalates. In response to these challenges, this research paper presents a novel approach to barcode decoding—one that harnesses the power of two cutting-edge technologies: YOLOv8 and REAL-ESRGAN. YOLOv8, an acronym for "You Only Look Once version 8," is a state-of-the-art object detection model renowned for its speed and precision. REAL-ESRGAN, on the other hand, is an advanced image super-resolution network capable of enhancing image clarity and quality. By integrating these two technologies, we embark on a journey to decode barcode images with unprecedented accuracy and reliability, even in the face of adverse conditions. This paper explores the methodology, implementation, and results of our research into the combined use of YOLOv8 and REAL-ESRGAN for barcode decoding. We delve into the technical intricacies of these technologies, explaining how YOLOv8 excels at barcode localization and how REAL-ESRGAN enhances the readability of captured images. Our research goes beyond theory, offering practical insights into the implementation of this innovative approach and showcasing its performance through empirical results. Furthermore, we examine the potential applications of this barcode decoding system across a spectrum of industries, where speed and accuracy in barcode recognition are indispensable. The retail sector, for instance, stands to benefit from faster checkout experiences, while logistics and supply chain management can achieve heightened efficiency and accuracy in inventory tracking. As we progress further into the digital age, where data-driven decision-making is paramount, the ability to decode barcode images swiftly and reliably becomes increasingly critical. This paper stands as a testament to the promise of leveraging YOLOv8 and REAL-ESRGAN in the realm of barcode decoding, offering a glimpse into the future of barcode-based systems and their potential to revolutionize various industries. We evaluate our approach on a barcode image dataset and show that it outperforms state-of-the-art methods in terms of accuracy and noise tolerance. We believe that our method can be used to improve accuracy and reliability not only in barcode decoding but also in many other applications.

2 Related works

There is a large number of publications covering the localization and decoding of barcodes. In this context, the restoration of blurred images has already been discussed in detail in the literature. In the following, we give a brief overview of existing work on barcode localization and image restoration with classical methods as well as the more modern methods of deep learning. Traditional methods for locating barcodes in camera

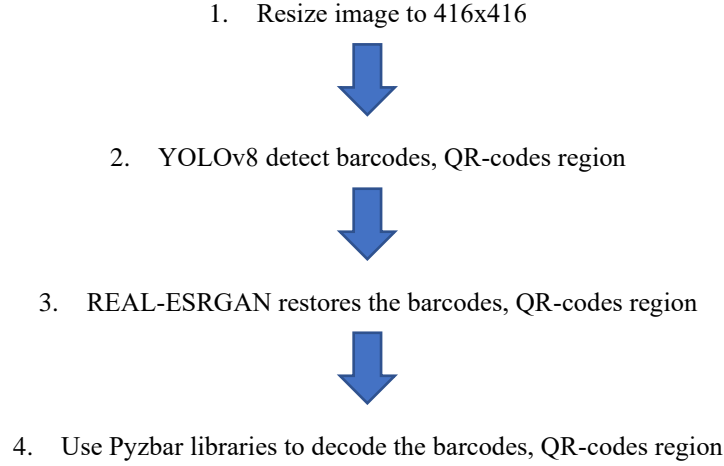
images typically use low-level image features. For example, Gallo et al. [1] used horizontal and vertical gradients to position barcodes in images where the camera's optical axis was perpendicular to the plane containing the barcode. However, this method fails if the barcode is almost vertically aligned. Gabor algorithm [2] extracted edge and corner maps from camera images to construct a barcode saliency map to independently locate the orientation of 1D and 2D barcodes. However, this method is time-consuming and the input image is blurred, especially with 1D barcodes. Neural networks have been successfully used to locate barcodes for about a decade now. Zamberletti et al. [3] proposed an angle-invariant method for barcode detection based on the Hough transform and multi-layer perceptron (MLP). However, the barcode must be clearly present in the image to be recognized. Hansen et al. [4] used the YOLO [5] detector to locate barcodes and predict their orientation. However, their approach does not segment the barcode. In general, a blurred image can be described as a convolution of an exact representation of an object and a Point Spread Function (PSF), which describes the system's impulse response and represents, for example, loss motion blur or sharpness. The inverse process, which produces a blurred image and obtains an unblurred image, is called decoding. Many classical methods exist for decoding, such as the Lucy-Richardson algorithm [6] and the Wiener or Tikhonov filter [7], [8]. Esedoglu [9] presented a decoding technique for 1D barcodes that takes into account the interactions of neighboring bars as well as the overall information contained in the observed signal. Yahyanejad et al. [10] presented an iterative barcode deblurring method based on the bimodal characteristics of the barcode image histogram. Lou et al. [11] presented a partial deblurring method for out-of-focus barcode images. However, results for real data show that this method reaches its limit at high blur levels. With the success of deep learning, convolutional neural networks are setting new standards in image restoration [12] [13] [14]. Most recently, the use of generative adversarial networks (GANs) [15] for image restoration has yielded promising results [16] [17]. However, the execution time of the proposed method is not applicable to real industrial scenarios.

3 Methods

We design the system's workflow in 4 steps as follows, including the following technologies and libraries:

- 1 Image resized: Before passing the image through the YOLOv8 model, we use the Opencv2 open-source library to resize the image, aiming to make the input image the same size as the size of the data set used to train the YOLOv8 model. At the same time, it is also to ensure processing speed when sending images through the model, specifically here we set the image to a size of 416x416.
- 2 Barcode localization: We use YOLOv8 to identify barcode regions in images. YOLOv8's real-time object detection enables accurate identification of barcode regions in images, even in challenging environments.
- 3 Barcode restoration: We use REAL-ESRGAN to enhance the quality and readability of barcode images. The image super-resolution ability of REAL-ESRGAN can greatly improve the clarity of barcode images, It can be said that it almost restores the barcode image to its original state, this is to maximize accuracy when decoding barcode images.
- 4 Decode barcode images: We use Pyzbar. This is an open source library for reading one-dimensional barcodes and QR codes. Its advantages are ease of use, high accuracy and fast response time. Its

disadvantage is that it cannot work with blurry, noisy images and changing environmental brightness, which we overcome by integrating the two methods YOLOv8 and REAL-ESRGAN.



4 Experiments

In this section, we detail the experimental setup, data collection, and results of our novel approach for decoding barcode images using the workflow involving YOLOv8, REAL-ESRGAN, and Pyzbar. The experiments were conducted to assess the accuracy, speed, and robustness of our system in decoding barcode images across various scenarios and conditions.

Dataset: For our experiments, we assembled a diverse dataset comprising a wide range of barcode types, including one-dimensional barcodes and QR codes. This dataset encompassed images captured under varying conditions, such as different lighting environments, image resolutions, and degrees of distortion. Additionally, we introduced intentional image degradation, including blurriness and noise, to simulate challenging real-world scenarios.

Hardware: The training part were conducted on a standard workstation with two NVIDIA GPU, it is RTX 3060 (Graphics Processing Unit), CUDA version 11.7 to accelerate the deep learning computations. The specifications of the hardware ensured efficient processing of images and real-time performance during barcode localization and restoration.

Software and Libraries: The software stack consisted of Python as the primary programming language, along with the following libraries and frameworks:

- OpenCV2: Utilized for image resizing and preprocessing.
- YOLOv8: Employed for barcode localization.
- REAL-ESRGAN: Utilized for barcode image restoration.
- Pyzbar: Applied for the final barcode decoding step.

4.1 Barcode localization accuracy

We aimed to evaluate the accuracy of barcode localization using YOLOv8. We measured the precision, recall, and F1-score of the barcode detection process on our diverse dataset. The metrics were calculated by comparing the ground truth barcode regions with the regions identified by YOLOv8. The experiment assessed the model's capability to accurately locate barcode regions within complex and varied images.



Fig. 2. Barcode localization accuracy.

4.2 Restoration quality and decoding accuracy

Focus on evaluating the effectiveness of REAL-ESRGAN in improving the quality and readability of barcode images. The results show that the REAL-ESRGAN model has removed noise and blur and returned the restored image to its original state. We then measured the image super-resolution ability of REAL-ESRGAN by comparing the quality of the recovered barcode images with the original high-resolution images from our dataset. Next aims to evaluate the overall accuracy and reliability of our barcode decoding system. We used Pyzbar to decode barcode images and measured decoding accuracy on images subjected to a variety of challenges, including blur, noise, and environmental brightness changes. This experiment determined the extent to which our integrated approach mitigates the limitations commonly associated with barcode decoding under adverse conditions.

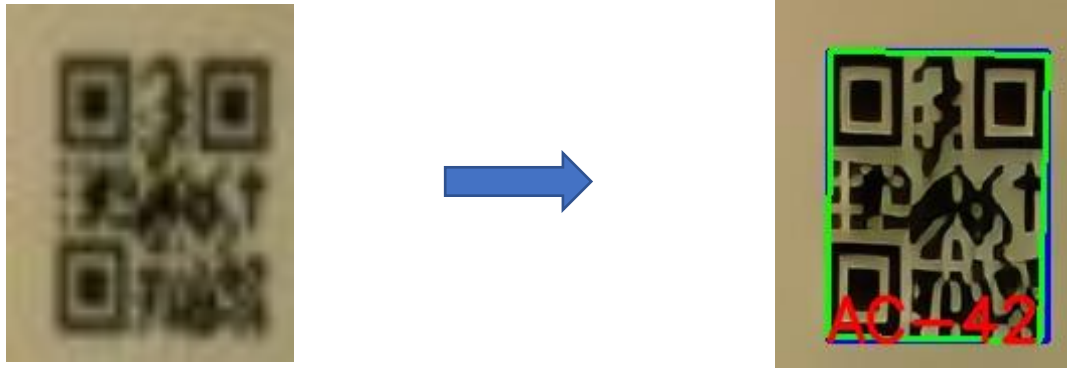


Fig.3. The region of QR-code was restored using REAL-ESRGAN

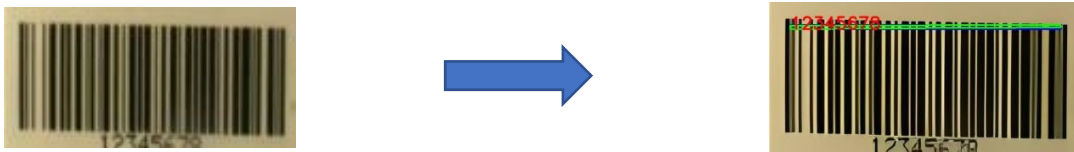


Fig.4. The region of barcode UPC was restored using REAL-ESRGAN



Fig.5. The region of barcode CODE-128 was restored using REAL-ESRGAN

4.3 Proposed methods

Experiment 4.1 shows that YOLOv8 achieves high accuracy in barcode positioning, even in difficult situations. Experiment 4.2 highlights the robustness of our system, the results show that REAL-ESRGAN improved the quality of barcode images, effectively restoring them to their original state. Experiment 4.3 demonstrates the feasibility of REAL-ESRGAN's image recovery ability. Pyzbar has successfully decoded the barcode. These findings highlight the potential of our integrated approach to transform barcode decoding across diverse industries. Furthermore, the tests confirm the synergistic effect of combining YOLOv8, REAL-ESRGAN, and Pyzbar in solving the challenges associated with barcode image processing, ensuring speed and accuracy in these applications changing environmental conditions.

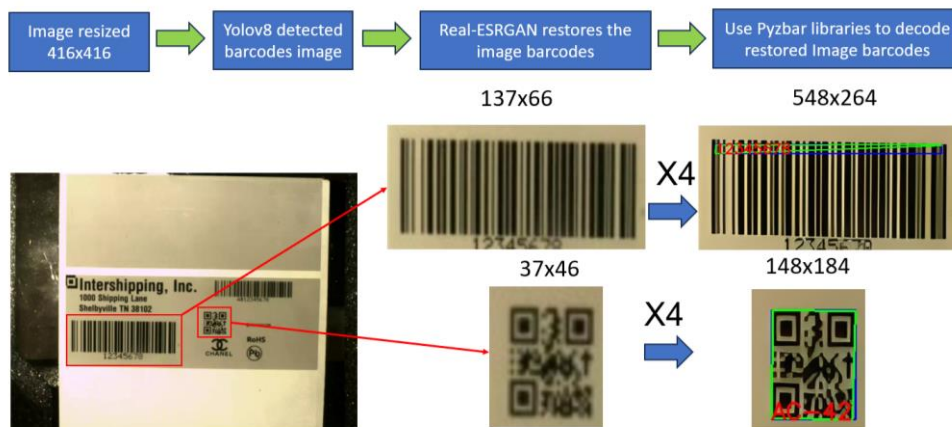


















Fig.6. Deploy system workflow on practical data

4.4 Compare with previous works

Works	Inputs	Outputs
SwinIR [20]	  	  
ESRGAN [21]	  	  
RealSR [22]	 	 















		
BSRGAN [23]		
		
		
Real-ESRGAN [19]		
		
		

Table. 1. Comparison REAL-ESRGAN with previous methods, on data objects specified as image barcodes and QR codes






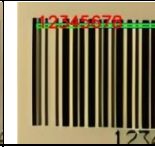















	Input	SwinIR [20]	ESRGAN [21]	RealSR [22]	BSRGAN [23]	Real-ESRGAN [19]	Ground-truth
Barcode UPC							
Width and Height	137x66	548x264	548x264	548x264	548x264	548x264	
PSNR (dB)		6.99	6.70	6.55	6.37	6.28	
SSIM [0,...1]		0.0734	0.0733	0.0781	0.0584	0.0783	
Time reference (ms)		200	188	235	218	244	
Barcode EAN							
Width and Height	103x40	412x156	412x156	412x156	412x156	412x156	
PSNR (dB)		6.94	6.79	6.69	6.55	6.56	
SSIM [0,...1]		0.0956	0.0867	0.0887	0.0664	0.0758	
Time reference (ms)		119	87	25	22	206	
QR-Code							
Width and Height	37x46	144x180	144x180	144x180	144x180	144x180	
PSNR (dB)		6.99	6.89	6.65	6.77	6.86	
SSIM [0,...1]		0.1055	0.1174	0.1202	0.1129	0.1198	
Time reference (ms)		101	139	26	24	188	

Table. 2. Compare the calculated metrics with the ground truth image

Explanation of metrics:

- Width and Height: Width and height of image.

- PSNR (dB): Computes the peak signal-to-noise ratio, in decibels, between two images. This ratio is used as a quality measurement between the original and a compressed image. The higher the PSNR, the better the quality of the compressed, or reconstructed image.

$$PSNR = 10 * \log_{10} \left(\frac{R^2}{MSE} \right) (dB)$$

In the previous equation, R is the maximum fluctuation in the input image data type. For example, if the input image has a double-precision floating-point data type, then R is 1. If it has an 8-bit unsigned integer data type, R is 255, etc.

The MSE represents the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error. The lower the value of MSE, the lower the error.

$$MSE = \frac{\sum_{M,N} [l_1(m,n) - l_2(m,n)]^2}{M * N}$$

- SSIM: Computed for the image with respect to the reference image. The reference image is usually needs to be of perfect quality. This quantitative measure considers three parameters namely luminance, contrast and structural information between the two images to computed the SSIM value.
- Time reference (ms): Model time for restoring photos is measured in milliseconds.

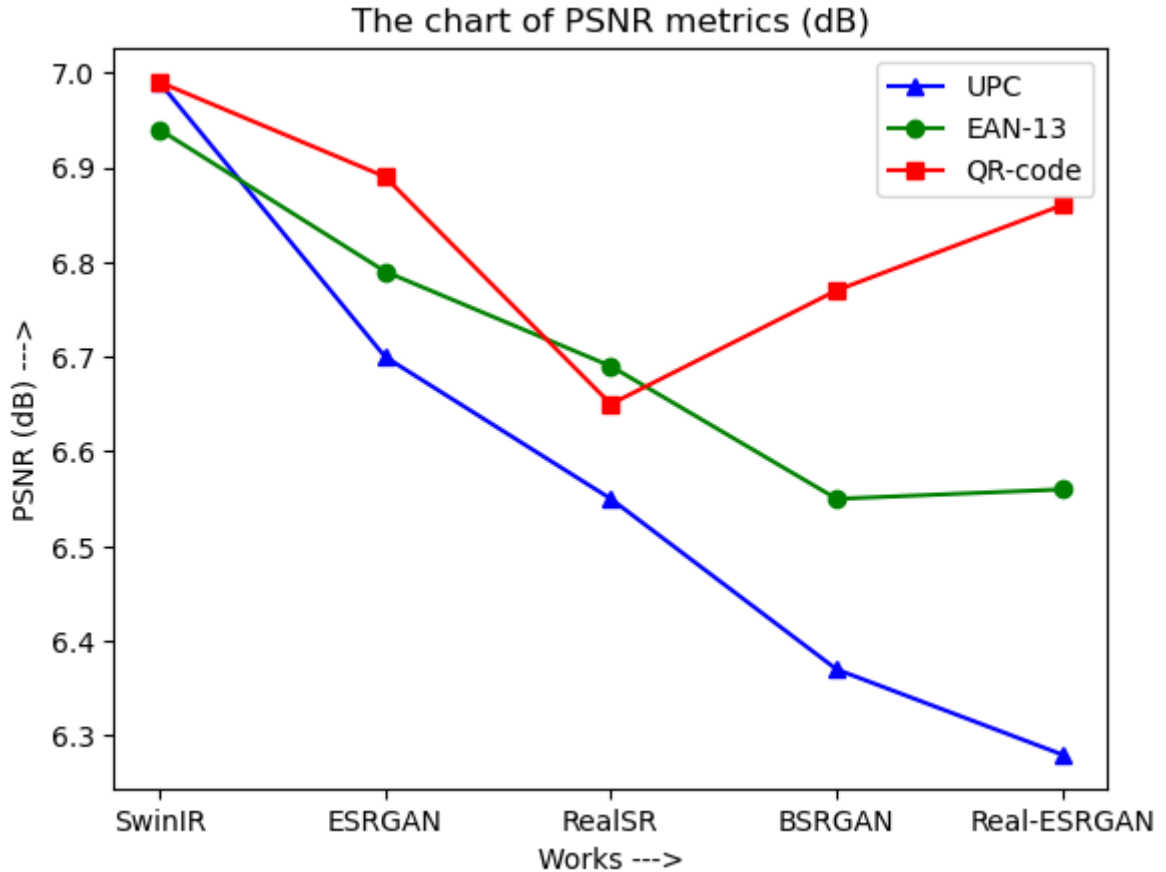


Fig.7. The chart of PSNR metrics

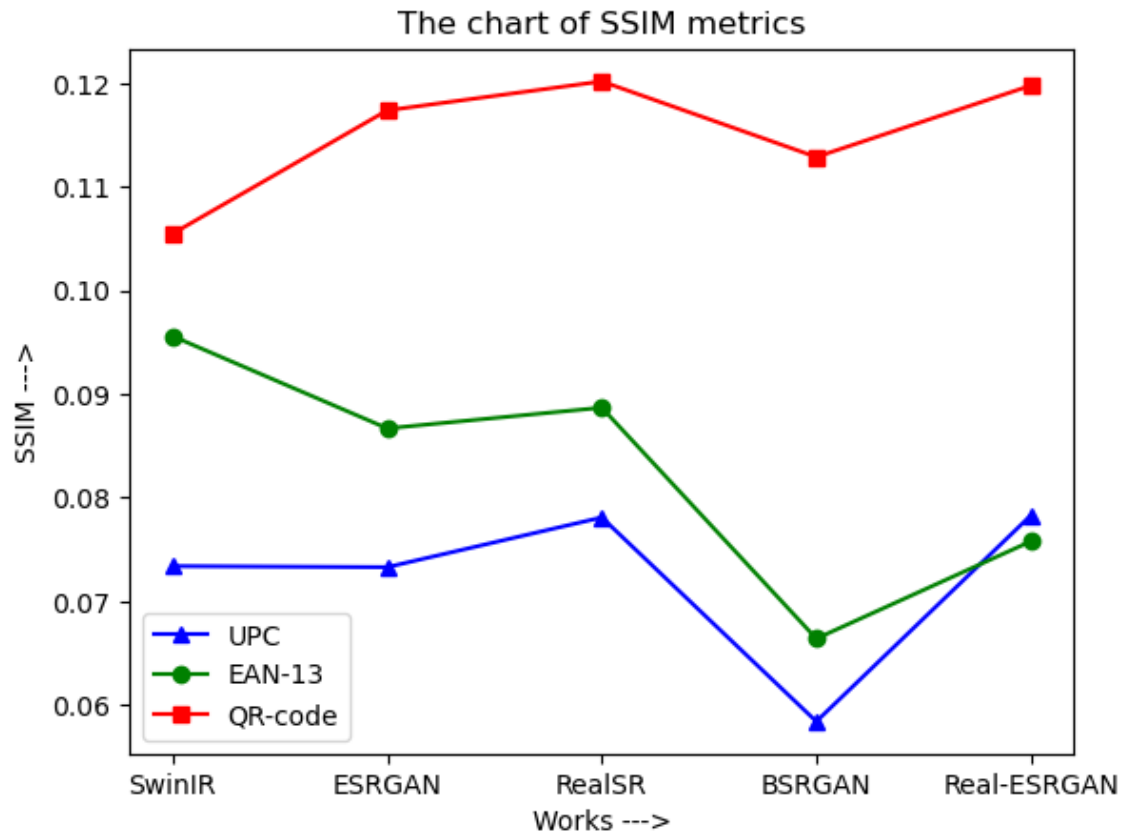


Fig.8. The chart of SSIM metrics

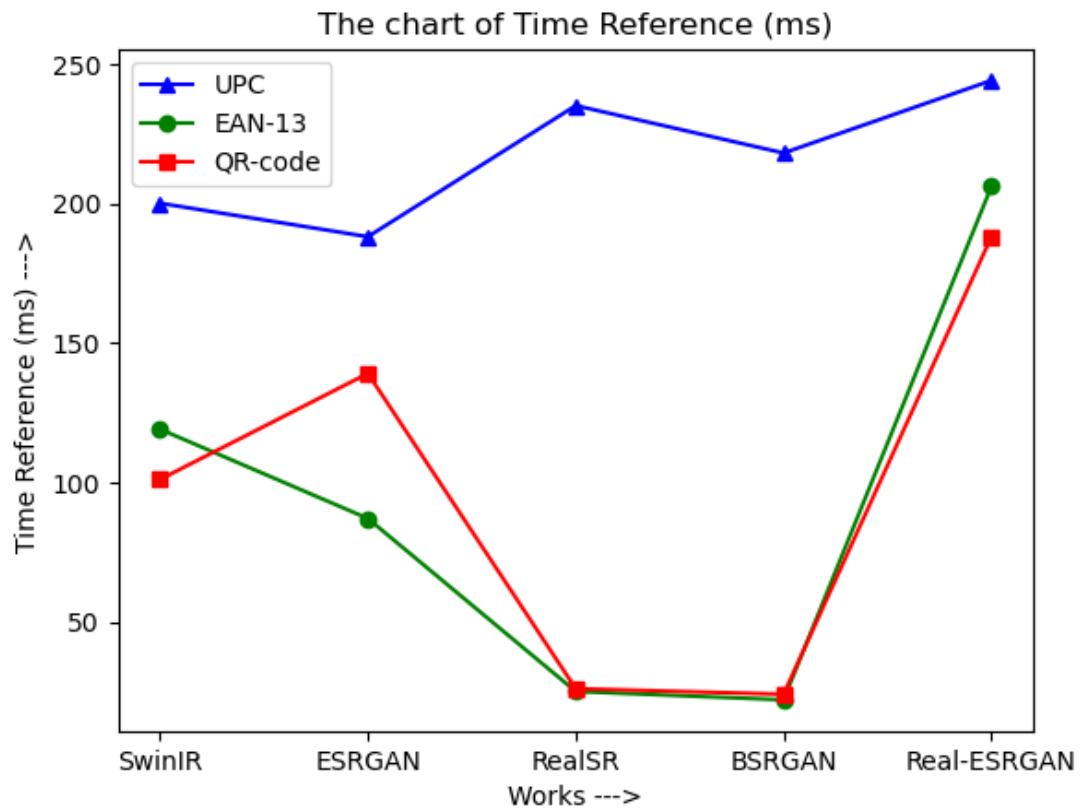


Fig.9. The chart of Time Reference

➔ When looking at the evaluation metrics, we see that the PSNR index of the models when compared to the ground-truth image is all below 20. The SSIM index is all below 0.2, although these indices are really low compared to the ground-truth image, so these methods still need to improve in terms of algorithms as well as training data in the future. But with the naked eye, we can see that the Real-ESRGAN [19] method has the best image recovery ability among all the remaining methods.

5. Conclusion

As we conclude this study, we reflect on the significance of our findings and the broader implications of this integrated system. Unlocking barcode localization accuracy: The first step in our workflow involves resizing images using the OpenCV2 library, ensuring compatibility with the YOLOv8 model. This preparation not only helps optimize processing speed but also lays the foundation for accurate barcode positioning. Even in challenging environments due to varying lighting conditions and image distortions, YOLOv8 demonstrated outstanding accuracy, emphasizing its effectiveness in locating barcodes. The test shows REAL-ESRGAN's excellent ability to improve the quality and readability of barcode images, REAL-ESRGAN's image super-resolution capabilities, such as restoring barcode images to their original state. pristine, dramatically improving the clarity and fidelity of barcode presentation. This step is important in maximizing the accuracy of subsequent barcode decoding. The final step is to decode the barcode using Pyzbar, which will create an integrated system capable of decoding barcode images with outstanding accuracy. While Pyzbar may falter in such situations, our integrated approach exploits the strengths of both YOLOv8 and REAL-ESRGAN to overcome these challenges and achieve reliability. Implications and future directions: The implications of this research extend across industries where barcode technology is indispensable. As we conclude this study, we acknowledge that this integrated approach is only a stepping stone toward a more complex and flexible barcode decoding system. Future research directions could explore further improvements to the YOLOv8 model, fine-tune REAL-ESRGAN for specific barcode types, and develop Pyzbar enhancements to handle challenging imaging conditions alike. In a broader context, the combination of computer vision, super-resolution imaging and barcode decoding illustrates the profound potential of interdisciplinary approaches in solving the world's complex challenges. Finally, we compare the REAL-ESRGAN method with old super-resolution methods and the visual evaluation results show that REAL-ESRGAN recovers barcode images in the best state.

REFERENCES

- [1] 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, 2011.
- [2] G. Sörös and C. Flörkemeier, "Blur-resistant joint 1d and 2d barcode localization for smartphones", *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*, pp. 1-8, 12 2013.
- [3] A. Zamberletti, I. Gallo, M. Carullo and E. Binaghi, "Neural image restoration for decoding 1-d barcodes using common camera phones", *VISAPP*, vol. 1, pp. 5-11, 01 2010.
- [3] H. Zhang, G. Shi, L. Liu, M. Zhao and Z. Liang, "Detection and identification method of medical label barcode based on deep learning", *2018 Eighth International Conference on Image Processing Theory Tools and Applications (IPTA)*, pp. 1-6, 11 2018.
- [4] D. K. Hansen, K. Nasrollahi, C. B. Rasmusen and T. B. Moeslund, "Real-time barcode detection and classification using deep learning", *IJCCI*, pp. 321-327, 2017.
- [5] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement", *CoRR*, vol. abs/1804.02767, 2018, [online] Available: <http://arxiv.org/abs/1804.02767>.
- [6] D. A. Fish, A. M. Brinicombe, E. R. Pike and J. G. Walker, "Blind deconvolution by means of the richardson-lucy algorithm", *J. Opt. Soc. Am. A*, vol. 12, no. 1, pp. 58-65, 01 1995.
- [7] A. D. Hillery and R. T. Chin, "Iterative wiener filters for image restoration", *IEEE Transactions on Signal Processing*, vol. 39, no. 8, pp. 1892-1899, 08 1991.
- [8] D. Trong, C. Phuong, T. Tuyen and D. Thanh, "Tikhonovs regularization to the deconvolution problem", *Communication in Statistics Theory and Methods*, vol. 43, pp. 4384-4400, 10 2014.
- [9] S. Esedoglu, "Blind deconvolution of bar code signals", *Inverse Problems*, vol. 20, pp. 121-135, 07 2004.
- [10] S. Yahyanejad and J. Strm, "Removing motion blur from barcode images", *CVPRW*, pp. 41-46, 7 2010.
- [11] Y. Lou, E. Esser, H. Zhao and J. Xin, "Partially blind deblurring of barcode from out-of-focus blur", *SIAM Journal on Imaging Sciences [electronic only]*, vol. 7, pp. 740-760, 4 2014.
- [12] S. Nah, T. H. Kim and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring", *IEEE CVPR*, pp. 257-265, 7 2017.
- [13] M. Noroozi, C. Paramanand and P. Favaro, "Motion deblurring in the wild", *Lecture Notes in Computer Science*, pp. 65-77, 01 2017.
- [14] X. Tao, H. Gao, X. Shen, J. Wang and J. Jia, "Scale-recurrent network for deep image deblurring", *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8174-8182, 6 2018.
- [15] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. WardeFarley, S. Ozair, et al., "Generative adversarial networks", *ArXiv*, vol. Abs/1406.2661, 2014.
- [16] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin and J. Matas, "Deblurgan: Blind motion deblurring using conditional adversarial networks", *IEEE/CVF CVPR*, pp. 8183-8192, 8 2018.
- [17] O. Kupyn, T. Martyniuk, J. Wu and Z. Wang, "Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better", *IEEE ICCV*, pp. 8877-8886, 2019.
- [18] Jocher, G., Chaurasia, A., & Qiu, J. (2023). YOLO by Ultralytics (Version 8.0.0) [Computer software]. <https://github.com/ultralytics/ultralytics>

- [19] X. Wang, L. Xie, C. Dong and Y. Shan, "Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data," 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 2021, pp. 1905-1914, doi: 10.1109/ICCVW54120.2021.00217.
- [20] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool and R. Timofte, "SwinIR: Image Restoration Using Swin Transformer," 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 2021, pp. 1833-1844, doi: 10.1109/ICCVW54120.2021.00210.
- [21] Wang, X. et al. (2019). ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In: Leal-Taixé, L., Roth, S. (eds) Computer Vision – ECCV 2018 Workshops. ECCV 2018. Lecture Notes in Computer Science(), vol 11133. Springer, Cham. https://doi.org/10.1007/978-3-030-11021-5_5.
- [22] X. Ji, Y. Cao, Y. Tai, C. Wang, J. Li and F. Huang, "Real-World Super-Resolution via Kernel Estimation and Noise Injection," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 2020, pp. 1914-1923, doi: 10.1109/CVPRW50498.2020.00241.
- [23] Zhang, K. et al. "Designing a Practical Degradation Model for Deep Blind Image Super-Resolution." 2021 IEEE/CVF International Conference on Computer Vision (ICCV) (2021): 4771-4780.