CHEN, R., HUANG, H., YU, Y., REN, J., WANG, P., ZHAO, H. and LU, X. 2023. Rapid detection of multi-QR codes based on multistage stepwise discrimination and a compressed mobilenet. *IEEE internet of things journal* [online], Early Access. Available from: https://doi.org/10.1109/JIOT.2023.3268636

Rapid detection of multi-QR codes based on multistage stepwise discrimination and a compressed mobilenet.

CHEN, R., HUANG, H., YU, Y., REN, J., WANG, P., ZHAO, H. and LU, X.

2023

Rapid Detection of Multi-QR Codes Based on Multistage Stepwise Discrimination and A Compressed MobileNet

Rongjun Chen, Hongxing Huang, Yongxing Yu, Jinchang Ren, *Senior Member, IEEE*, Peixian Wang, Huimin Zhao, Xu Lu

***Abstract*—Poor real-time performance in multi-QR codes detection has been a bottleneck in QR code decoding based Internet-of-Things (IoT) systems. To tackle this issue, we propose in this paper a rapid detection approach, which consists of Multistage Stepwise Discrimination (MSD) and a Compressed MobileNet. Inspired by the object category determination analysis, the preprocessed QR codes are extracted accurately on a small scale using the MSD. Guided by the small scale of the image and the end-to-end detection model, we obtain a lightweight Compressed MobileNet in a deep weight compression manner to realize rapid inference of multi-QR codes. The *Average Detection Precision* (ADP), *Multiple Box Rate* (MBR) and running time are used for quantitative evaluation of the efficacy and efficiency. Compared with a few state-of-the-art methods, our approach has higher detection performance in rapid and accurate extraction of all the QR codes. The approach is conducive to embedded implementation in edge devices along with a bit of overhead computation to further benefit a wide range of real-time IoT applications.**

***Index Terms*—Multi-QR Codes, rapid detection, Internet of Things(IoT), MobileNet, embedded edge devices.**

I. Introduction

With the development of digital twin and deep learning in the Industrial Internet of Things [1], the QR code, as a kind of low-cost reading label, can empower the reliable construction of deep learning-based perception framework [2]. However, fast detection of multiple QR codes in images remains a challenging task, though it is crucial for registering large quantities of sample tubes or commodity goods in various scenes, such as medical facilities for COVID-19 testing, warehousing and logistics, as illustrated in Fig. 1. This requires not only high performance in real-time, namely detection of all QR codes from one image efficiently, but also low computational complexity for embedded implementation using the edge devices.

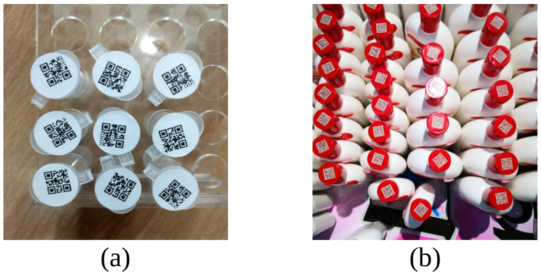

Fig. 1: Multi-QR codes detection scenarios. (a) and (b) present the application of multi-QR codes detection in the medical test tubes for COVID-19 testing and goods warehousing, respectively.

Earlier approaches [3–5] of the QR code detection detected QR codes by calculating the width ratio of the black and white regions. However, they can only be applied to images with a simple background and high contrast. The image resolution, illumination and noise etc can easily influence the detection performance of these approaches [6]. By extracting the contour lines, QR codes can be detected using the morphological processing and Hough Transform [7]. However, this approach has a high computational complexity, yet the overall accuracy in dealing with complex background is still low. Most conventional QR code detection approaches only consider the single QR code scenario, which can not be effectively applied in the multi-QR codes detection task.

Deep learning-based object detection can help to predict the location of multiple QR codes in an image. These object detection approaches can be divided into two categories according to the detection principles, i.e. single-stage and two-stage approaches. The single-stage approach mainly uses the fully convolutional networks to process the image, so that the class probability and position of

The work was supported in part by the National Natural Science Foundation of China under Grants 62072122 and 62176067, in part by the Special Projects in Key Fields of Ordinary Universities of Guangdong Province under Grant 2021ZDZX1087, in part by Guangzhou Science and Technology Plan Project under Grant 2023B03J1327 and 202102020857, in part by the Research Projects of Guangdong Polytechnic Normal University under Grant 22GPNUZDJS17 and in part by the Special Project Enterprise Sci-tech Commissioner of Guangdong Province under Grant GDKTP2021033100 and the GPNU Dazhi Scholarship (*Corresponding authors: J. Ren, H. Zhao, X. Lu, e-mails: jinchang.ren@ieee.org, zhaohuimin@gpnu.edu.cn, bruda@126.com*).

R. Chen, H. Huang, Y. Yu, J. Ren, P. Wang, H. Zhao and X. Lu are with the School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China.

Y. Yu is the School of Automation Science and Engineering, South China University of Technology, Guangdzhou 510641, China.

J. Ren is also with the National Subsea Centre, Robert Gordon University, Aberdeen, AB21 0BH, U.K.

the target can be directly estimated. As a result, the detection speed is fast, but the accuracy rate is lower than that of the two-stage ones. Typical algorithms include YOLOv3 [8], SSD [9], etc. The two-stage target detection approach often needs to determine the target candidate regions first before applying region-based classification. Compared with the single-stage scheme, the two-stage one, for instance, R-CNN [10], Fast R-CNN [11] and Faster R-CNN [11], has a higher accuracy but is more time-consuming.

Many researchers have also introduced various network models in the research field of multi-QR codes detection. For example, [12] proposed an effective one-dimensional barcode detection network. However, the number of model parameters is high, and the detection accuracy still needs to be improved. [13] first proposed a multi-class barcode detection network based on the Faster R-CNN network that is suitable for complex environments, which can locate multiple barcodes and correct the distortions. However, this approach is too time-consuming and does not consider the influence of illumination changes on detection accuracy. Subsequently, the authors compressed and optimized the network model to reduce the parameters in [14], but the computational cost remains high. Based on the EfficientDet [15], [16] locates the four vertices of the QR code with a relatively high training cost. Furthermore, this approach adds an additional key-points regression layer for locating four vertices of the QR code accurately, which increases the algorithm's time complexity. In order to reduce the computation overhead of the learning-based model in the QR code detection task, the batch QR code detection approach [17] is proposed, which combines the conventional QR code detection and the lightweight model, demonstrating improved detection speed. However, the application scenarios of this approach have some limitations. The detection accuracy of QR codes will be poor following the high false detection rate, when the detected image contains complicated background or noise.

In summary, conventional detection approaches encounter difficulties to detect multiple QR codes contained in a single image, while learning-based ones depend on a massive model structure, sufficient labeled data, and high hardware configurations, leading to poor detection speed and difficulty for embedded implementation on edge devices. In order to overcome these shortcomings, we propose an effective approach for rapid detection of multiple QR codes. The major contributions are highlighted as follows.

1) Based on object category determination analysis, we propose a Multistage Stepwise Discrimination (MSD) approach, which is capable of rapidly classifying redundant target candidate regions and accurately subsampling them into small sizes.
2) Inspired by the small-scale and end-to-end detection task, we use the deep learning-based network to capture the convolutional features of QR codes, and propose the lightweight Compressed MobileNet for high accuracy classification.
3) We propose a rapid detection approach for multiple QR codes based on MSD and Compressed MobileNet. The experimental results show that our approach has higher detection accuracy and the faster speed. Our model is conducive to embedded implementation on edge devices, meeting the needs of real-time multi-QR codes detection in IoT applications.

The remainder of this paper is organized as follows. In Section II, we review the related work in detail. The framework of the proposed approach is given in Section III. Section IV presents the experiments and the results, followed by a summary and discussion of future research in Section V. Table I lists all acronyms, along with descriptions.

TABLE I: List of All Acronyms

Acronym	Significance
MSD	Multistage Stepwise Discrimination
SD	Square Degree
SDSM	Square Degree Selection Method
ASSM	Area Statistics Selection Method
DD	Distance Discrimination
CDA	Coincidence Degree Analysis
GSS	Grayscale Statistics
MBR	Multiple Box Rate
ADP	Average Detection Precision

II. RELATED WORK AND MOTIVATIONS

QR code is a low-cost label reading strategy over the perception layer of the IoT. However, accurate and rapid QR code detection on edge sensing devices has been challenging. Related approaches can be divided into two categories, i.e. image processing-based approach and deep learning-based model as detailed below.

A. Image processing-based approach

Early works [18–21] use conventional image processing approaches, including image filtering, binarization operation, morphological processing and the Hough transform, to detect the QR codes, followed by extracting the QR codes as the foreground. This kind of approach is simple and effective, and has also been applied to QR code detection in recent years. [22] proposes a QR code location method based on the areas with high concentration of edge and corner structures which is so-called the structure matrix in the HSV(Hue, Saturation, Value) color system. The edge detection is used to capture the boundary information, before being transmitted to the HSV space to locate the QR codes. [23] proposes an approach for 1D barcode detection and recognition, which utilizes binarization and other operations to preprocess the image, and then performs boundary detection and classification to locate the barcode. Based on the conventional image processing, a more efficient QR code detection approach is proposed in

[24], which can not only locate and detect QR codes under uneven lighting environments, but also be suitable for low-resolution images. After converting the captured image into the gray one followed by binarization, [24] obtains the QR code's bounding-box by locating the Finder Pattern of the QR code. [25] also converts the RGB image into a binary one and then performs morphological processing to detect the QR code. [26] proposes a QR code detection approach based on image integration, which uses the integral image to determine a threshold for binarization, and then applies the Finder Pattern of the QR code to locate the QR code through the extracted connection regions.

The primary purpose of the binarization operation above is to enhance the edges whilst removing the unwanted background. Image binarization can be divided into local and global thresholding based approaches [27]. The local thresholding determines the binarization threshold using the pixel distribution with a neighborhood region. Although the image can carry out this binarization operation adaptively, it needs to calculate multiple thresholds, leading to low efficiency. The global threshold approach achieves the binarization by setting one threshold, thus it is fast. OTSU [28] is a kind of global thresholding approach with simple and rapid calculation, unaffected by image brightness and contrast. It divides the image into the background and the foreground according to the histogram distribution. Because of the rich black and white block features of the QR code, there is an apparent distinction between the foreground and the background. We can obtain the edge information of the QR code contours by using the OTSU to get the maximal inter-class variance of the QR code and the background. Therefore, in our work, we use the OTSU to calculate the binarization threshold and convert the RGB image into a binary one.

B. *Deep learning-based model*

In recent years, with the rapid development of Deep learning and computer vision techniques, Deep Neural Network (DNN) models have been applied to detect multiple QR code patterns in the image. [29] uses YOLOv2 [30] to detect the positions of the multiple QR codes in the image. Simultaneously, the rotation angles of the QR codes are regressed by DarkNet19. This approach achieves favorable results on the Muenster barcode dataset. [31] uses the edge detection operator to process images, mapping them to the Hough space. The rotation angles of the 1D barcode are obtained through a trained Multi-Layer Perceptron Network, which achieves satisfactory results in the experiments. A geometric approach in [32] is introduced to the deep learning-based model, which extracts the bounding box regions and uses a line detection algorithm to detect corresponding barcodes after utilizing the YOLOv2 to locate multiple 1D barcodes in the image. In [33], an end-to-end model is proposed to detect multiple QR codes in the image. This approach contains a Quadrilateral Regression Layer that can accurately locate barcodes and a Multi-scale Spatial Pyramid Pooling Layer to detect small-scale barcodes.

In addition, many DNN models have achieved high accuracy in object detection tasks in recent years. These deep learning-based models can not only be divided into the two-stage and the single-stage, but also anchor-based and anchor-free model where anchor is the pregenerated bounding box. Fast R-CNN, Faster R-CNN, YOLOv2 and SSD belong to anchor-based methods, while DenseBox [34], CornerNet [35], ExtremeNet [36], FSAF [37], FCOS [38] and FoveaBox [39] are anchor-free ones. However, it is for the multi-QR codes detection that the practicability of these approaches is considered necessarily in actual scenes, where whether they can meet the requirements of the perceptive devices with scarce computing resources is particularly challenging. Although anchor-free models proposed in recent years simplify the structure of anchor-based models and reduce the computational complexity, they still have a large number of parameters and require high-performed hardware, which prevents them from being applied directly to directly the multi-QR codes detection task. In order to deploy these DNN models in real scenarios, they need to be compressed through knowledge distillation, low-rank decomposition, network quantization and so on. Moreover, SqueezeNet [40], ShuffleNet [41], MobileNet [42–44] and other lightweight networks are proposed to reduce the complexity of the DNN models and the parameters. Inspired by network compression technology and the lightweight network model, we introduce a lightweight model in the last part of our approach to improve the recognition accuracy.

C. *Our motivations*

Nowadays, the Internet of Things (IoT) has elicited great attention from both industrial and scholarly circles, such as big data, energy management, and commerce, to name a few. However, in the perception layer of IoT, QR codes, as the most widely used labeling technology, still present challenges. These challenges are primarily composed of two aspects. The first is the complicated industrial environments [45], where the intricacy of real-world settings renders detection and decoding more arduous. This complexity encompasses multiple scales, occlusion between QR codes and backgrounds, geometric distortion, and orientation, commonly seen in industrial environments. The second is the slow detection speed for multi-QR codes, which may also affect the security of QR code information transmission [46, 47] due to the poor detection speed, as well as for ensuring the production line efficiency. Therefore, all of these necessitate that IoT devices are able to quickly acquire as much QR code information as possible.

To tackle these challenges, we aim to develop a rapid multi-QR codes detection approach under industrial environments. Although attempts have been made to address some of these issues, as summarized in Table II, they are far from fully solved. In addition, existing approaches can be categorized into image processing-based and Deep learning-based. The former features poor detection accuracy but fast speed even on the central processing units

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3268636

(CPUs). The latter requests more computational power hence has difficulty for implementation on a resource constrained edge devices yet it tends to have high detection accuracy. How to combine them for both high detection accuracy and fast speed will be the major purpose of the proposed approach.

III. THE PROPOSED APPROACH

In the QR code image detection task, a large number of predicted bounding boxes are generated. Through utilizing the category determination principle and IoU [48] to analyze these bounding boxes, they can be divided into independent ones, intersecting ones and containing ones. In this section, we propose an efficient box-screening strategy, namely the Multistage Stepwise Discrimination(MSD)-based approach, to improve the screening performance. Finally, we present our multi-QR codes rapid detection approach in detail.

A. Multistage Stepwise Discrimination

1) Analysis of predicted bounding boxes based on category determination principle: In the multi-QR codes detection task, there will be many generated bounding boxes that may not contain the full targets. Thus these need be examined to remove the false alarms. The classification of these boxes is a kind of object category determination problem. As shown in Fig. 2, there are four types for the overlap of bounding boxes in an image: 1) The four corners of a bounding box are all contained in the interior of the another. 2) A corner of a bounding box is contained inside the another. 3) Two corners of a bounding box are contained inside the another. 4) The bounding box does not contain any corners of the others. As for a further comment, these types can fall into two cases, one is the intersection relation which is shown in Fig. 2(a), and the another is the inclusion one which is presented in Fig. 2(b)(c)(d). These bounding boxes $\mathcal{X}$ can be divided into independent boxes c_{ind}, intersecting ones c_{int}, and containing ones c_{con}. So the set $\mathcal{Y}$ of category is defined as

$$\mathcal{Y} = \{c_{ind}, c_{int}, c_{con}\} \tag{1}$$

$\boldsymbol{x} \in \mathcal{X}$ represents a bounding box, and the risk of misplacing it into category c is $R(c|\boldsymbol{x})$, which is defined as

$$R(c|\boldsymbol{x}) = R(c_{ind}|\boldsymbol{x}) \times (R(c_{int}|\boldsymbol{x}) + R(c_{con}|\boldsymbol{x})) \tag{2}$$

This risk can be minimized by finding a decision rule $h : \mathcal{X} \mapsto \mathcal{Y}$, which h is defined as

$$R(h) = \mathbb{E}[R(h(\boldsymbol{x}) \mid \boldsymbol{x})] \tag{3}$$

where $\mathbb{E}$ stands for the expected function. For each sample $\boldsymbol{x}$, if h can minimize the risk value $R(h(\boldsymbol{x}) \mid \boldsymbol{x})$, the risk $R(h)$ will also be minimized, i.e

$$h(\boldsymbol{x}) = \underset{c \in \mathcal{Y}}{\arg\min}\, R(c \mid \boldsymbol{x}) \tag{4}$$

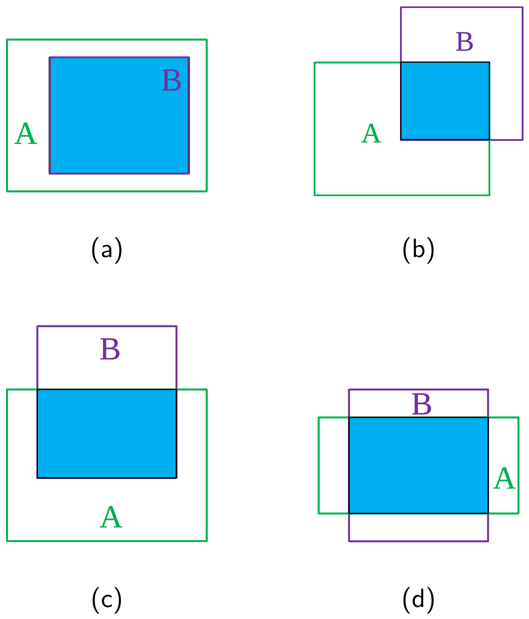

Fig. 2: The overlapping relations of Box A and Box B. The interior of Box A in (a)-(c) contains 4, 1, and 2 corners of Box B respectively. The interior of Box A dose not contain the corners of Box B in (d).

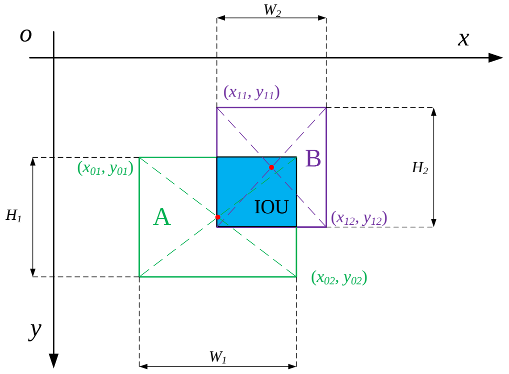

Fig. 3: The intersection over union of Box A and Box B.

We study the bounding box categories to improve the detection efficiency and propose a decision rule h, namely Multistage Stepwise Discrimination(MSD), which can eliminate extra boxes rapidly. Thus all bounding boxes are quickly divided into independent boxes, intersecting ones, containing ones. Finally, the QR code within each bounding box is detected and extracted.

2) Design of MSD approach: For two bounding boxes containing the same detection target, as shown in Fig. 3, the *IoU* score can measure their intersection degree. In order to judge whether two boxes are in the inclusion relation, we propose another metric based on the *IoU* score, namely the coincidence score C. A and B in Fig. 3 are two intersected boxes. The width iou_w, height iou_h

TABLE II: Summarized Comparison of Related Works

Works		Features
Image processing-based	[18–21]	Mainly using the Hough transform to detectiong QR codes
	[22]	Applying a kind of structure matrix in HSV color space to locate the QR code
	[23]	Detecting the boundary of QR code after binarization
	[24]	Having a better performance in uneven lighting environments
	[24]	Locating the QR code via the Finder Pattern
	[25]	Applying morphological processing to images to find QR codes
	[26]	Finding the QR code by the Finder Pattern after image integration
Deep learning-based	[29]	Using YOLOv2 to detect the positions of QR codes
	[31]	Obtaining the rotation angle of QR code by a Multi-Layer Perceptron Network
	[32]	Locating QR code by YOLOv2 following the line detection algorithm
	[33]	Proposing an end-to-end model to detect QR codes

and IoU score of their intersected region can be defined as

$$\begin{cases} iou_w = min(x_{01}, x_{02}, x_{11}, x_{12}) + w_1 + w_2 \\ \qquad - max(x_{01}, x_{02}, x_{11}, x_{12}) \\ iou_h = min(y_{01}, y_{02}, y_{11}, y_{12}) + h_1 + h_2 \\ \qquad - max(y_{01}, y_{02}, y_{11}, y_{12}) \\ iou_{AB} = \dfrac{iou_w \times iou_h}{S_A + S_B - iou_w \times iou_h} \end{cases} \tag{5}$$

where (x_{01}, y_{01}), (x_{02}, y_{02}), w_1 and h_1 are the upper-left corner, the lower-right corner, width and height of A. (x_{11}, y_{11}), (x_{12}, y_{12}), w_2 and h_2 are also the upper-left corner, the lower-right corner, width and height of B. S_A and S_B denote the area of A and B respectively. The coincidence score C_{AB} is calculated as

$$C_{AB} = \frac{iou_w \times iou_h}{min(S_A, S_B)} \tag{6}$$

The Coincidence Degree Analysis (CDA) can be conducted on A and B, i.e. $C_{AB} = 1$ when the relation of A and B is inclusion. Consider $\mathcal{X}_{ind}$, $\mathcal{X}_{int}$ and $\mathcal{X}_{con}$ are the boxes sets of c_{ind}, c_{int} and c_{con} respectively. n_{ind}, n_{int} and n_{con} are the number of samples of $\mathcal{X}_{ind}$, $\mathcal{X}_{int}$ and $\mathcal{X}_{con}$. When a sample $\boldsymbol{x}$ is classified into c_{ind}, the risk value $R(c_{ind}|\boldsymbol{x})$ is defined as

$$R(c_{ind}|\boldsymbol{x}) = sign(\sum_{j=1}^{n_{ind}} iou_{\boldsymbol{x}\boldsymbol{x_j}}),\ \boldsymbol{x_j} \in \mathcal{X}_{ind} \tag{7}$$

where $sign(x)$ is a sign function defined as

$$sign(x) = \begin{cases} 1,\ x > 0 \\ 0,\ x = 0 \\ -1,\ x < 0 \end{cases} \tag{8}$$

When $\boldsymbol{x}$ is assigned to c_{ind} correctly, $\boldsymbol{x}$ does not intersect any other bounding box $\boldsymbol{x_j}$ in $\mathcal{X}_{ind}$, whilst $R(c_{ind}|\boldsymbol{x}) = 0$. Otherwise, $R(c_{ind}|\boldsymbol{x}) = 1$. When $\boldsymbol{x}$ is classified into c_{int}, the risk value $R(c_{int} \mid \boldsymbol{x})$ is defined as

$$\begin{aligned} R(c_{int}|\boldsymbol{x}) = \frac{1}{2} sign \left(\frac{1}{\sum_{j=1}^{n_{int}} e^{C_{\boldsymbol{x}\boldsymbol{x_j}}}} - \frac{1}{n_{int}} \right) \\ + 0.5,\ \boldsymbol{x_j} \in \mathcal{X}_{int} \end{aligned} \tag{9}$$

where $R(c_{int}|\boldsymbol{x}) = 0$ when there is an intersection relation between $\boldsymbol{x}$ and the bounding box $\boldsymbol{x_j}$ of $\mathcal{X}_{int}$, namely $\boldsymbol{x} \in \mathcal{X}_{int}$. Conversely, $R(c_{int}|\boldsymbol{x}) \in \{0.5, 1\}$. When $\boldsymbol{x}$ is classified into c_{con}, the risk value $R(c_{con} \mid \boldsymbol{x})$ is defined as

$$R(c_{con}|\boldsymbol{x}) = \sum_{j=1}^{n_{con}} C_{\boldsymbol{x}\boldsymbol{x_j}} - \left[\sum_{j=1}^{n_{con}} C_{\boldsymbol{x}\boldsymbol{x_j}} \right],\ \boldsymbol{x_j} \in \mathcal{X}_{con} \tag{10}$$

where $R(c_{con}|\boldsymbol{x}) = 0$ when there is inclusion relation between $\boldsymbol{x}$ and the bounding box $\boldsymbol{x_j}$ in $\mathcal{X}_{con}$, namely $\boldsymbol{x} \in \mathcal{X}_{con}$. On the contrary, the risk value is $0 < R(c_{con}|\boldsymbol{x}) < 1$.

So far, according to the IoU score and coincidence score, we have completed the category determination of independent bounding boxes, intersecting ones and containing ones by minimizing the risk value $R(c \mid \boldsymbol{x})$. After obtaining the category of each box by the MSD in this section, we propose a filtering algorithm to eliminate the redundant candidate boxes in $\mathcal{X}_{int}$ and $\mathcal{X}_{con}$.

3) Box filtering algorithm based on MSD: Given the upper-left corner (x_{lt}, y_{lt}), width w and height h of a bounding box, the center point (cx, cy) can be calculated as

$$\begin{cases} cx = x_{lt} + \dfrac{w}{2} \\ cy = y_{lt} + \dfrac{h}{2} \end{cases} \tag{11}$$

where the center points (cx_A, cy_A) and (cx_B, cy_B) in Fig. 3 can be found. The horizontal distance d_x and vertical distance d_y between the two center points, namely $|cx_A - cx_B|$ and $|cy_A - cy_B|$, can also be calculated as

$$\begin{cases} d_x = |x_{11} - x_{01} + \dfrac{w_2}{2} - \dfrac{w_1}{2}| \\ d_y = |y_{11} - y_{01} + \dfrac{h_2}{2} - \dfrac{h_1}{2}| \end{cases} \tag{12}$$

The Distance Discrimination (DD) can be conducted for A and B, when they have the critical intersection distance. The width t_x and the height t_y of the intersected region can be calculated as

$$\begin{cases} t_x = \dfrac{w_1}{2} + \dfrac{w_2}{2} \\ t_y = \dfrac{h_1}{2} + \dfrac{h_2}{2} \end{cases} \tag{13}$$

where if two bounding boxes intersect each other, $d_x \leq t_x$ and $d_y \leq t_y$. Otherwise, there is no a intersection relation between them.

Fig. 4 gives the flowchart how to eliminate the extra boxes and only retain those smaller ones in Fig. 3, where the intersecting bounding boxes that do not contain QR code should be discarded. Different from the containing bounding boxes, intersecting ones cannot quickly determine whether the boxes contain real QR codes. Therefore, we design a rapid filtering approach by combining both the size and the edge features of the QR code, which has two stages in Fig. 4. First, all the independent bounding boxes are obtained as the prior knowledge, using their size features to filter the intersecting ones preliminarily. Then, the edge feature is utilized to further filter the remaining boxes. When detecting the multi-QR codes, their sizes should be within a range of $[min\_area, max\_area]$, which can be determined using the areas of the detected independent bounding boxes. Actually, we introduce the area redundancy parameter $\beta = 0.2$ and set the size range as $[(1-\beta)min\_area, (1+\beta)max\_area]$, aiming to prevent the valid boxes from missing or being eliminated. The bounding boxes are retained when their areas are within this range. Furthermore, for the bounding boxes that contain the QR code, another significant feature is its edge grayscale distribution, which comes from the stage of binarization and morphological processing. The Grayscale Statistics (GSS) is utilized to determine the edge grayscale distribution in each intersecting bounding box of the same target, which is for finding the box with the maximal area proportion of the QR code.

B. The Proposed Approach for Rapid Detection

This section introduces our detection algorithm for multi-QR codes using MSD presented in the previous section. Different from approaches based on conventional image processing and QR code's finder pattern, we eliminate false bounding boxes without splitting an image into several blocks. These blocks' quantity and size are uncertain and may lead to incomplete QR code region. Hence actually those approaches have the delicate adaptive ability. Inspired by DNN-based detection approaches, we obtain all bounding boxes that may contain QR code in an image according to the QR code's dense edge information after preprocessing and contours extraction of RGB image. And then the boxes filtering algorithm based on MSD is applied to these bounding boxes, which has the low computational complexity and the high precision. Finally, we train a lightweight model, i.e. Compressed MobileNet, to improve the classification precision of bounding boxes.

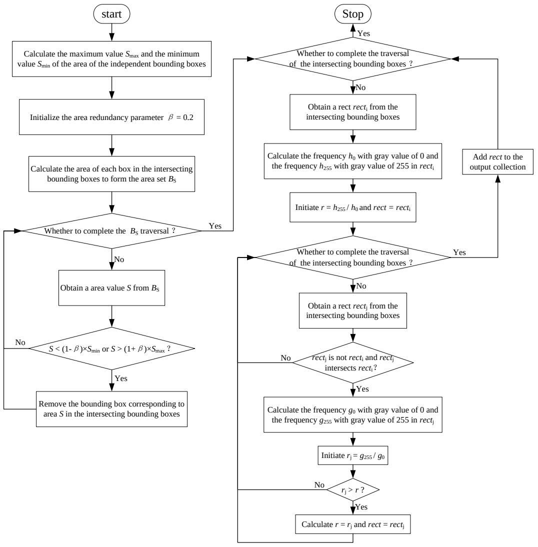
start
Calculate the maximum value S_{max} and the minimum value S_{min} of the area of the independent bounding boxes
Initialize the area redundancy parameter $\beta = 0.2$
Calculate the area of each box in the intersecting bounding boxes to form the area set B_S
Whether to complete the B_S traversal ?
Yes
No
Obtain a area value S from B_S
$S < (1-\beta)\times S_{min}$ or $S > (1+\beta)\times S_{max}$?
No
Yes
Remove the bounding box corresponding to area S in the intersecting bounding boxes
Stop
Yes
Whether to complete the traversal of the intersecting bounding boxes ?
No
Obtain a rect $rect_i$ from the intersecting bounding boxes
Calculate the frequency h_0 with gray value of 0 and the frequency h_{255} with gray value of 255 in $rect_i$
Add $rect$ to the output collection
Initiate $r = h_{255} / h_0$ and $rect = rect_i$
Whether to complete the traversal of the intersecting bounding boxes ?
Yes
No
Obtain a rect $rect_j$ from the intersecting bounding boxes
$rect_j$ is not $rect_i$ and $rect_j$ intersects $rect_i$?
No
Yes
Calculate the frequency g_0 with gray value of 0 and the frequency g_{255} with gray value of 255 in $rect_j$
Initiate $r_j = g_{255} / g_0$
$r_j > r$?
No
Yes
Calculate $r = r_j$ and $rect = rect_j$

Fig. 4: The diagram of box filtering based on MSD.

Fig. 5 is the pipeline of the approach we proposed, which is described in detail in the following.

1) Preprocessing: At the beginning, the multi-QR codes' detection accepts the inputs, high resolution images. These images are usually down sampled to reduce the computational burden of the subsequent steps. However, the number of QR codes is uncertain in the image captured by the actual camera. To make sure that as many QR codes as possible can be detected, we only carry out the gray scale transformation on the RGB images instead of the downsampling. Actually, whether the proportion of finder pattern which is unique to QR code or the orthogonal distribution of QR code's edge grayscale, some complex environments will make both of them insignificant and lead to detection failure directly. In the study of the saliency feature of the QR code, we find that QR code' edge has a dense distribution of the black and white blocks which can be applied to detection as rich edge information. Moreover, a standard QR code is surrounded by a white border to segregate out of the background. This edge information and the white border are less sensitive to the environment with robust saliency. Thus we utilize them as the basis feature to rapidly find out all the regions that may contain QR code and mainly carry out edge feature extraction of gray image during preprocessing.

The edge detection operator is applied to QR code's edge extraction. There are two types of common edge detection operators. One is first-order differential operator, for instance, Roberts, Sobel and Scharr. Another is second-order differential operators such as Laplacian and Log/Marr. The Roberts operator is conductive to detecting steep edges but sensitive to noise as same as the Laplacian operator, which leads to inaccurate location. The Log/Marr operator consists of Gaussian smoothing and Laplacian filter, which requires more computation. The Sobel operator and the Scharr operator both can

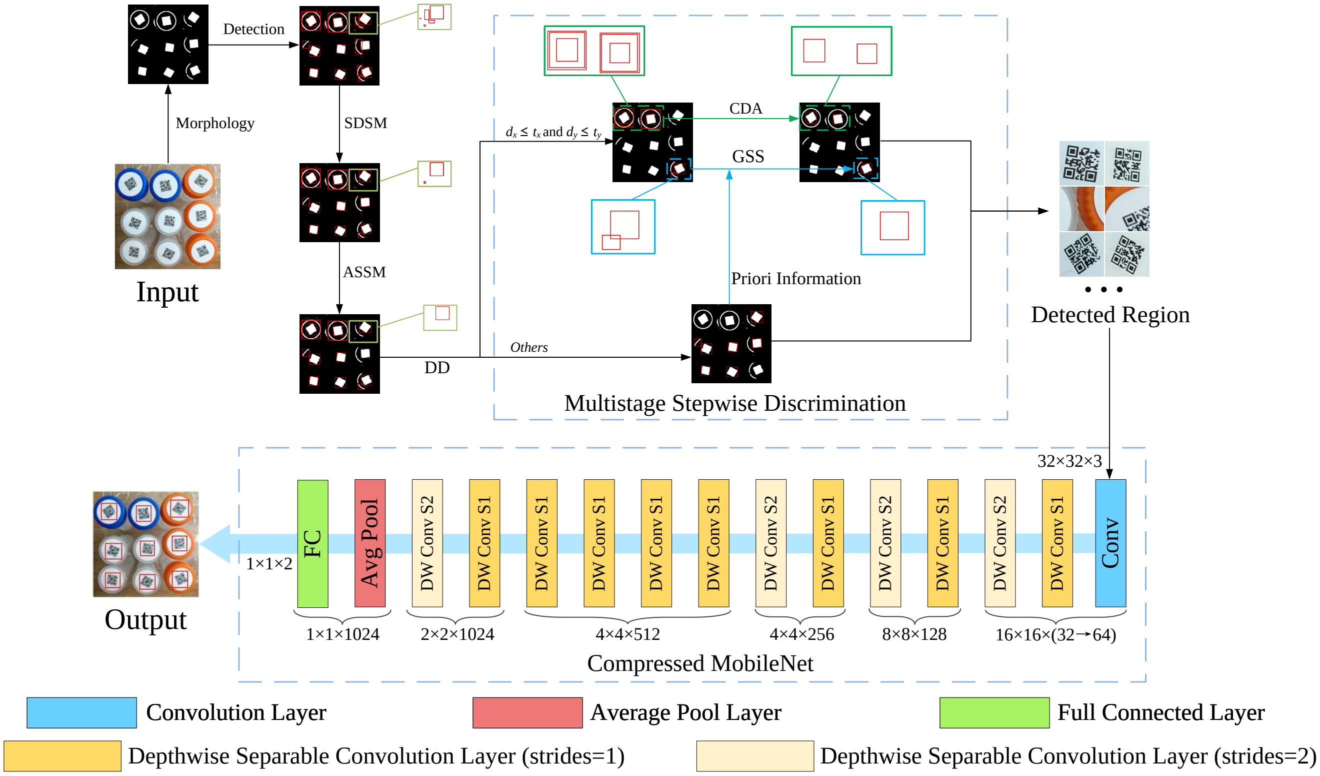

Fig. 5: Overview of MSD-based approach for rapid detection of multi-QR codes. DD indicates distance discrimination after ASSM, CDA represents coincidence degree analysis, and GSS denotes grayscale statistics.

detect edges accurately. But the sensitivity of the Sobel operator is different in all directions. And the Scharr operator can calculate grayscale gradient changes more accurately via weighted neighborhood. Thus we use the Scharr operator to process the gray image, obtaining the gradient image. Furthermore, we apply binarization to the gradient image to enhance the visual quality of the edge features. Since the gradient image where the range of the grayscale distribution is [0, 255] does not have complex background, we utilize OTSU [28] to obtain the binary image. Finally, we dilate the binary image several times after eroding it to eliminate the slight noise. The intermediate results in the preprocessing stage are shown in Fig. 6.

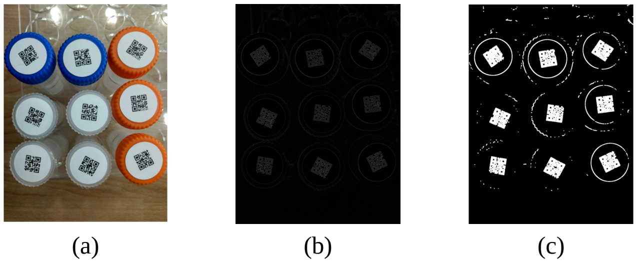

Fig. 6: The results of the preprocessing. (a) is original image. (b) and (c) are images processed by edge detection and dilation.

2) The Detection of QR Code Regions: Since the QR code's edge feature can not distinguish completely the QR codes from background, this section captures these ones' bounding box through contour extraction and further filtering. After applying contour extraction [49] to bounding regions that contain the QR code's edge feature, we represent these ones as Λ_0, i.e, $\Lambda_0 = ([x_0, y_0, w_0, h_0], ...[x_n, y_n, w_n, h_n])$, where $[x_i, y_i, w_i, h_i](i = 0, ..., n)$ denotes the i_{th} bounding region and n is their count. Most bounding regions in Λ_0 are negative samples, which contain only the edge background and no the QR code's edge feature. To gain positive ones, inspired by the characteristic that bounding shape is close to a square, we define the Square Degree SD of the bounding shape as

$$SD = \frac{min(w, h)}{max(w, h)} \tag{14}$$

where w and h are the bounding shape's width and height. We eliminate bounding shapes that are not like a square after setting SD-based filter condition, which we call Square Degree Selection Method (SDSM). SDSM is introduced in Algorithm 1.

We filter out bounding boxes that contain only background after SDSM, but there are still negative samples which have a large or small one. Thus we first calculate all the area size of the bounding box obtained by SDSM following figuring out the critical area size via statistical

Algorithm 1 Square Degree Selection Method

```
Input: The bounding regions $\Lambda_0$
Ouput: The collection $\Lambda_{out}$ of the bounding regions containing QR codes
Initialize: $t \leftarrow 0.65$
for $rect \in \Lambda_0$ do
    The $rect$'s $SD$ was calculated according to Eq. (14)
    if $SD \geq t$ then
        Keep $rect$ and add it to $\Lambda_{out}$
    end if
end for
```

Algorithm 2 Area Statistics Selection Method

```
Input: The collection $\Lambda_1$ of the bounding boxes obtained by SDSM
Ouput: The collection $\Lambda_{out}$ of the bounding boxes containing QR codes
Count the area size set $A_S$ and the number $num$ of the bounding box
Obtain the area size histogram of $A_S$ by statistical analysis
Create a dictionary $D$ which key is the value range corresponding to the area size frequency in the histogram
Sort $D$ in descending order according to the frequency
Get the upper bound value $v$ of the area size range in the second highest frequency
Set the parameter: $\alpha \leftarrow 0.5$
Calculate the segmentation threshold: $t \leftarrow \alpha \times v$
for $rect \in \Lambda_1$ do
    Calculate $rect$'s area size $S$
    if $S > t$ then
        Keep $rect$ and add it to $\Lambda_{out}$
    end if
end for
```

analysis for the distribution of the area size histogram, and finally we regard it as segmentation threshold to further eliminate negative samples. These steps which we call Area Statistics Selection Method (ASSM) are shown in Algorithm 2.

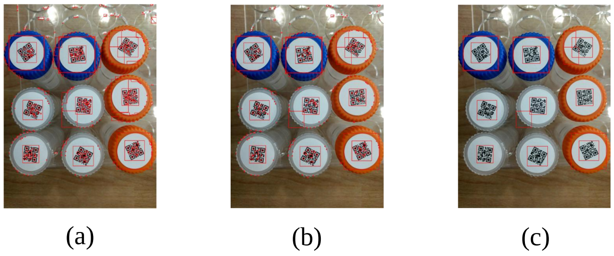

Fig. 7: The results of SDSM and ASSM. (a) shows the bounding boxes. (b) and (c) show the results of the bounding boxes processed by SDSM and ASSM.

Fig. 7 shows the QR codes' bounding boxes after preliminarily processing in SDSM and ASSM. Obviously we filter out most negative samples after ASSM but still retain several bounding boxes which overlap each other. These boxes also overlap positive samples, where it is hard to eliminate them. This phenomenon also exists in other multi-QR codes detection approaches. In our study, we use MSD-based process to weed out these hard-to-eliminate samples.

3) Bounding Boxes Filtering Based on MSD: We rapidly divided the previous stage's overlapping bounding boxes into independent boxes, intersecting ones and containing ones through MSD. In addition, these independent bounding boxes that we utilize as priori knowledge are combined with the QR code's edge feature to process intersecting bounding boxes and containing boxes, which is shown in Fig. 4. Finally, multi-QR codes as same as their bounding boxes are detected rapidly through *Preprocessing*, *The Detection of QR Code Regions* and *Bounding Boxes Filtering Based on MSD*. However, there is still low accuracy in bounding box locating. We introduce lightweight DNN-based boxes selection to solve this problem. Unlike the MSD which need to classify a great number of overlapping bounding boxes rapidly, the inputs to this lightweight DNN model are non-overlapping bounding regions that may contain QR codes. What's more, compared with other DNN models for multi-QR codes detection, the lightweight DNN model that we call Compressed MobileNet has a simple structure which requires a bit of overhand computation, whilist most of its inputs are positive samples.

4) Boxes Classification Based on Compressed MobileNet: Deep convolutional network has the ability of feature extraction that can map RGB image to convolutional latent space and obtain the target's higher-layer features. Therefore many object detection models design deep convolutional network as backbone, which can merge contextual feature and capture the deep representation of the inputs. But this method will increase the model's complexity, requiring more train cost to update parameters. The QR code itself consists of many black and white blocks that are obvious and abundant. This feature conduces to differentiate the QR code and other objects. Thus we only need to train a lightweight model that can learn this feature, which contributes to not only reducing the network parameters and the computation complexity, but also greatly improving the runtime speed. Whilist by doing this it is unnecessary to update a large number of parameters like other common object detection models.

In our work, we introduce the modifiable width multiplier and the modifiable resolution multiplier to compress a small trained MobileNet model, i.e, the Compressed MobileNet. The role of the modifiable width multiplier $\alpha \in (0, 1]$ is to thin a model uniformly at each layer. The number of a given layer's input channels M becomes αM and the number of its output channels N becomes αN. α is modifiable statistically during training through a series of experiments. The modifiable resolution multiplier $\rho \in (0, 1]$ we used is to reduce the inputs and the internal representation at each layer, which can reduce computational cost by ρ^2. ρ is also modifiable likehood of α. The Compressed MobileNet is used to category discriminant for the output to the stage of *Bounding Boxes Filtering Based on MSD*. The related experiment is conducted and discussed in section IV-A.

IV. Experiment Results and Analysis

This section mainly compares our approach and other approaches following analyzing them. In Section IV-A, we conduct the comparison of the learned-based model performance. The time cost and the overhand computation of our approach are analyzed in detail in Section IV-B. Section IV-C shows the quantitative results of the multi-QR codes detection approaches.

Our experiment is conducted on a server equipped with an Intel Core i7-5930k 3.50GHz CPU and an NVIDIA RTX1080Ti GPU. The server system is Ubuntu 20.04, and it has 32GB of memory. What' s more, we use the medical test tube images in [17] as our dataset. The image resolution is 3096 × 4048, and each image may contain 2/9/40/80/120 or 160 QR codes. Each QR code region' s area accounts for 0.25%-10% of the image. Some examples in our dataset are shown in Fig. 8.

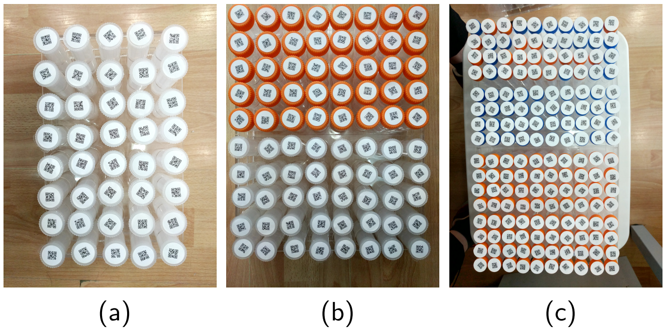
(a) (b) (c)

Fig. 8: Some examples of our dataset. (a) 40 QR codes. (b) 80 QR codes. (c) 160 QR codes.

A. Quantitative Comparison for Lightweight Models

This section mainly compares the classification performance of lightweight models. We first obtain images from *Preprocessing* and *The Detection of QR Code Regions*. These images may contain background or the QR code and are split into two groups manually. One group contains 30,000 positive samples and the another has 17,000 negative ones. We use these images as sub-dataset to train all the lightweight models and to evaluate their performance. Before training, we randomly divided the training set and test set according to 9:1. Fig. 9 shows some examples in our sub-dataset.

We first train a small MobileNet model, whilist adjusting our modifiable width multiplier α and modifiable resolution multiplier ρ step by step to compress this model. By applying α and ρ, our Compressed MobileNet accepts RGB images as inputs with size 32 × 32. Its parameters was reduced to 0.26 million, which is about 1/10 of the standard MobileNet's. What's more, in order to enhance our model's robustness, we extended training samples via data augmentation and increased the number of training epochs. The learning rate is set to 10^{-3}, and we employed a variety of tricks to obtain a model with the best performance during training. Our model converged approximately between the 45th and 50th iterations. We exclusively used this model for validation in succeeding trials. Table III shows the accuracy comparison of our approach. In this comparison, SqueezeNet, ShuffleNet, MobileNetV1, MobileNetV2 and MobileNetV3 all adopt standard specifications.

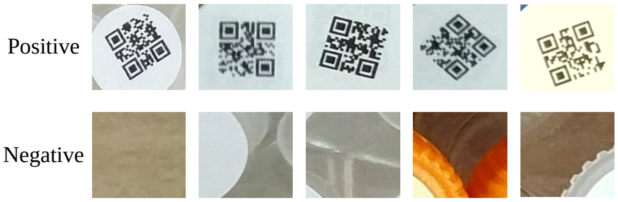

Fig. 9: Some example of our sub-dataset.

TABLE III: Quantitative Results of Different Lightweight models

Lightweight Model	Million Parameters	Accuracy(%)
SqueezeNet [40]	1.13	96.66
ShuffleNet [41]	1.91	95.73
MobileNetV1 [42]	4.17	95.51
MobileNetV2 [43]	3.36	97.37
MobileNetV3 [44]	5.40	**98.11**
Compressed MobileNet	**0.26**	97.60

As seen in Table III, the accuracy of all lightweight models exceeds 95% in our binary classification task, in which our Compressed MobileNet has the least number of parameters, i.e. less than 0.3 million, compared to over 1 million in other models. Although the MobileNetV3 has a classification accuracy of 98.1%, it has more than 5 million parameters thus has difficulty for implementation on IoT devices. The SqueezeNet and ShuffleNet have a small number of parameters, yet their classification accuracies are only 96.6% and 95.7%, respectively. The Compressed MobileNet has a comparable classification accuracy up to 97.6%, though much less parameters. By training a small lightweight model, we can not only achieve higher classification accuracy, but also ensure the embedded implementation on edge devices for real-time multi-QR codes detection.

B. Performance Analysis of The Proposed Approach

In this section, we present the implementation details and relevant experiment results of the proposed approach, while using detection accuracy and time cost to evaluate the detection performance of multi-QR codes. The detection accuracy consists of two metrics, $Detection\ Precision(DP)$, the detection accuracy for an image, and $Average\ Detection\ Precision_c(ADP_c)$, the average detection accuracy on a specific number c of QR codes. DP and ADP_c in our work are defined as

$$DP_i = (1.0 - \frac{n_i^e + n_i^m}{n_i^{gt}}) \times 100\% \tag{15}$$

$$\begin{cases} S = \{2, 9, 40, 80, 120, 160\} \\ ADP_c = \dfrac{1}{n_c} \displaystyle\sum_{i=1}^{n_c} DP_i, \ c \in S \end{cases} \tag{16}$$

$$MADP = \frac{1}{6} \sum_{i=1}^{6} ADP_{S_i} \tag{17}$$

where

n_i^e is the number of false QR codes in the i_{th} image,

n_i^m and n_i^{gt} are the number of missed ones and the number of real ones, respectively,

c is the number of QR codes in an image,

n_c is the number of images with the number of QR codes c in the test dataset,

DP_i denotes the detection accuracy of the i_{th} image,

ADP_c represents the average detection accuracy of all images with the number of QR code c,

$MADP$ represents the mean average detection precision of the approach.

For the actual multi-QR codes detection, false detection is allowed and missed detection should not be tolerated. Because we can filter bounding boxes via post process where redundant or repeated bounding boxes are eliminated. It is necessary that the multi-QR codes detection approach should detect QR codes in an image as many as possible. Thus the ability of generating redundant bounding boxes is also important. The $Multiple\ \ Box\ \ Rate(MBR_c)$ and the $Mean\ Multiple\ Box\ Rate(MMBR)$ of an image on a specific number of QR codes are defined as

$$MBR_c = \frac{n_c^r}{n_c} \times 100\%, \quad c \in S \tag{18}$$

$$MMBR = \frac{1}{6} \sum_{i=1}^{6} TAP_{S_i} \tag{19}$$

For all images with the number of QR codes c, n_c^r in Eq. (18) represents the number of images where the number of generated bounding boxes is not less than c. The MBR_c reflects the ability of generating redundant bounding boxes of the approach. In Eq. (19), the $MMBR$ represents the mean MBR_c in the test dataset. In order to compare the detection effect of different approaches, we divided the test dataset into 6 groups and each group contains 50 images. $Group_c(c = 2, 9, 40, 80, 120, 160)$ indicates that each image in this group contains c QR codes.

Table IV shows the detection accuracy of different groups. It can be seen that for the detection of an single image DP exceeds 97% in $Group_{40}$, $Group_{80}$, $Group_{120}$ and $Group_{160}$. Most of the ADP_c remain at 99.8%. However, the DP drops slightly on $Group_2$ and $Group_9$ while being at least 50% on $Group_2$. The reason is that each image in $Group_2$ contains only two QR codes. If one QR code is missed during detection, DP will decrease. Table V shows the quantitative results of $Mutltiple\ Box\ Rate$ in different groups. It can be seen that the MBR of our approach remains above 94% with an average value of 97.6%, showing an efficient detection performance. Additionally, for the false detection and missed detection that are shown in Fig. 10, the cause where our approach fails is the influence of extreme blur and extreme noise.

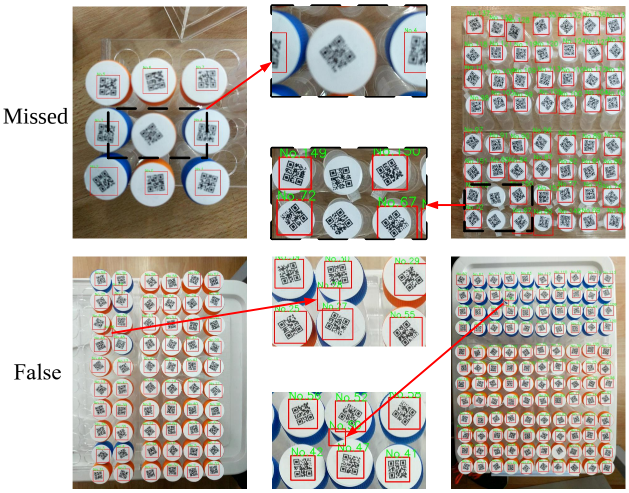

Fig. 10: Some examples of our results. The first row shows the missed examples and the second row shows the false examples.

Table VI compares the running time, in seconds, of several key modules in our approach, where obviously our approach has a fast processing speed. It requires about 0.6s to accurately detect all 160 QR codes within an image, and more time is needed if the image contains more QR codes. In Fig. 11, we further visualize the running time in each module as a histogram, where it clearly shows that the Compressed MobileNet takes about 2/3 of the total time. Regarding the mean square error of each stage's detection time across 6 groups, the error is 0.5139 for the overall detection time, yet it reduces to 0.3160 for the Compressed MobileNet. This indicates that the detection stability of our approach is largely benefitted from the Compressed MobileNet. Actually, the number of parameters in the Compressed MobileNet can be further reduced by adjusting the modifiable width multiplier and the resolution multiplier. In addition, through MSD-based *Bounding Boxes Filtering*, most of the inputs to the Compressed MobileNet are positive samples, which can further reduce the running time for the classification of negative samples and improve the detection efficiency.

TABLE IV: Results of DP And ADP_c on Different Groups

Group	$N_{false}/N_{miss}/DN_{total}/N_{total}$	$DP(\%)$		$ADP_c(\%)$
		min value	max value	
$Group_2$	2/1/101/100	50	100	97
$Group_9$	2/2/450/450	88.9	100	99.1
$Group_{40}$	1/1/2000/2000	97.5	100	99.9
$Group_{80}$	3/0/4003/4000	98.8	100	99.9
$Group_{120}$	14/0/6014/6000	98.3	100	99.8
$Group_{160}$	5/10/7995/8000	97.5	100	99.8

TABLE V: Results of MBR_c on Different Groups

Group	$Group_2$	$Group_{40}$	$Group_{80}$	$Group_{120}$	$Group_{160}$	Mean
MBR_c	98%(49/50)	98%(49/50)	98%(49/50)	100%(50/50)	94%(47/50)	97.6%

TABLE VI: Running Time in Seconds of Key Modules in Our Approach

Group	Preprocessing	QR Regions Detection	MSD-Based Box Filter	Compressed MobileNet	Total
$Group_2$	0.1617	0.0085	0.010	0.2187	0.3989
$Group_9$	0.1697	0.0244	0.0066	0.2492	0.4499
$Group_{40}$	0.1644	0.0121	0.0056	0.2639	0.4460
$Group_{80}$	0.1694	0.0170	0.0159	0.3331	0.5354
$Group_{120}$	0.1738	0.0181	0.0188	0.3882	0.5989
$Group_{160}$	0.1722	0.0190	0.0201	0.4428	0.6541
Mean	0.1685	0.0165	0.0106	0.3160	0.5139
Proportion	32.79%	3.21%	2.06%	61.49%	100%

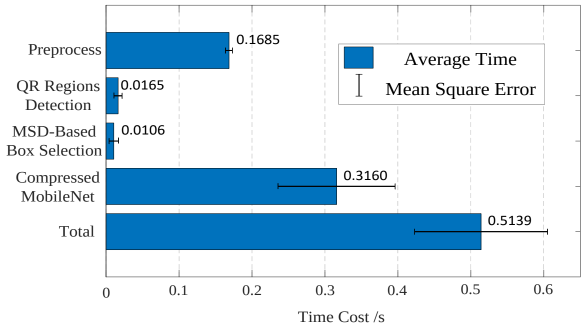

Fig. 11: Histogram of the module based running time in our approach.

C. Quantitative Comparison for Multi-QR Codes Detection Approaches

In this section, we use three kinds of metrics, i.e, ADP, MBR, and the time cost, to quantitatively compare the performance of our approach and other up-to-date approaches. How we used the test dataset is the same as that in the previous section. Table VII, Table VIII and Table IX are three representative results, corresponding to $Group_2$, $Group_{80}$ and $Group_{160}$ respectively. We applied the same train dataset to all the learning-based approaches in this section. The images in the train dataset are not repeated in the test dataset and each of them contains the number of QR codes ranging from 2 to 160. What's more, we did not stop the training of the network models until the training loss convergence and the hyper-parameters of these models were set uniformly according to their articles.

TABLE VII: The Detection Results of Different Approaches on $Group_2$

Approach	$ADP_2(\%)$	$MBR_2(\%)$
YOLOv3 [8]	45.86	75.00
Faster R-CNN [11]	68.68	87.00
Jia et al. [13]	75.55	79.00
Sharif et al. [16]	**98.00**	90.00
The Proposed Approach	97.00	**98.00**

As seen in Table VII, for $Group_2$ where each image contains 2 QR codes, the MBR_2 of our approach is the highest compared with other approaches, which means that our approach can generate more redundant bounding boxes. The ADP_2 of [16] is 1% higher than our approach, because the number of QR codes contained is fewer, leading to the false detection of our approach. In Table VIII, we can find that the ADP_{80} of our approach and [16] both have more than 99% on $Group_{80}$ where each image contains 80 QR codes, and the MBR_{80} of both is also over 98%. Compared with YOLOv3, Faster R-CNN and [13], our approach has the best detection accuracy on $Group_{80}$. With the increasing number of QR codes in an image, Table IX further shows an excellent detection performance of our approach on $Group_{160}$. On $Group_{160}$, except our approach which achieves over 99%, all other approaches have their performance declined apparently, i.e. 71.45% for Faster R-CNN, 37.21% for ADP_{80} and 34.24% for ADP_{160}, respectively.

As visually compared in Fig. 12 where different multi-QR codes detection approaches were evaluated in different groups, it can find that with the increase of the number of QR codes in an image, the *Average Detection Precision* curve and the *Multiple Box Rate* curve of YOLOv3, Faster R-CNN, [13] and [16] ascend gradually and subsequently decline. The reason is that these network models

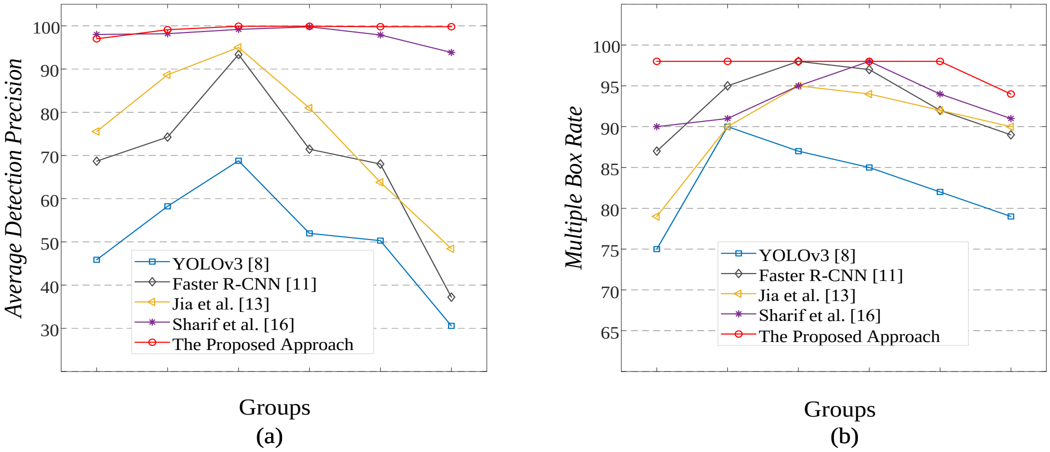

Fig. 12: Comparison of detection effect of different approaches on different groups. (a) Comparison of average detection precision. (b) Comparison of multiple box rate.

TABLE VIII: The Detection Results of Different Approaches on $Group_{80}$

Approach	$ADP_{80}(\%)$	$MBR_{80}(\%)$
YOLOv3 [8]	51.97	85.00
Faster R-CNN [11]	71.45	97.00
Jia et al. [13]	80.98	94.00
Sharif et al. [16]	99.77	98.00
The Proposed Approach	**99.90**	**98.00**

TABLE IX: The Detection Results of Different Approaches on $Group_{160}$

Approach	$ADP_{160}(\%)$	$MBR_{160}(\%)$
YOLOv3 [8]	30.57	79.00
Faster R-CNN [11]	37.21	89.00
Jia et al. [13]	48.41	90.00
Sharif et al. [16]	93.84	91.00
The Proposed Approach	**99.80**	**94.00**

[8, 11, 13, 16] require manually-setted anchor box sizes, which limits the size of the targets they can detect. If the size of a image is fixed, as the number of QR codes in this image changes, the size of these targets changes accordingly. Only when the QR codes are kept within an appropriate size range can these network models achieve optimal performance. In contrast, our approach does not lie in the anchor size setting, thus we can maintain the robust detection performance. Moreover, the more QR codes an image contains, the better detection effect our approach achieves. In addition, it can be seen from Fig.12(b) that the *Multiple Box Rate* of our approach is the highest in all groups of the test dataset. We attempt to explain such an excellent synthesis effect that our approach tends to generate more bounding boxes, namely being more tolerant towards the redundancy of bounding boxes. Table X shows the quantitative results of the average detection effect of each approach, where we use $MADP$, $MMBR$ and the time cost as evaluated metrics.

TABLE X: The Detection Results and The Running Time of Different Approaches

Approach	$MADP(\%)$	$MMBR(\%)$	Time(s)
YOLOv3 [8]	50.95	83.00	**0.1127**
Faster R-CNN [11]	68.83	93.00	1.3169
Jia et al. [13]	75.40	90.00	1.2852
Sharif et al. [16]	97.81	93.00	0.8112
The Proposed Approach	**98.34**	**95.00**	0.5522

As seen in Table X, YOLOv3 has a rapid detection speed, with an average detection time of 113 ms for a multi-QR codes image. However, both the $MADP$ and $MMBR$ of YOLOv3 is poor, which denotes that YOLOv3 has a low performance for multi-QR codes detection. In contrast, Faster R-CNN, as a two-stage detection model, has a higher $MADP$ and $MMBR$ than YOLOv3, but it is quite time-consuming. For [13], an improved Faster R-CNN, with rotation-invariant detection and key points regression for QR codes detection, also belonging to the dual phase detection approach, has achieved 90% of $MMBR$. However, when the QR code is large, the $MADP$ is poor. The QR code detection in [16] is improved using the EfficientDet, by adding the key point regression header network, which can capture the key points of the QR codes for regression training. The relatively high $MADP$ is achieved at up to 97.81%. Its $MMBR$ is also close to our approach, but the structure of the model is complex and results in a much higher computational cost. The overall detection accuracy of our approach is the best, which has the lowest computational cost to benefit real-time industrial applications.

V. CONCLUSION

In this paper, we have proposed a fast multi-QR codes detection approach, which consists of four stages: *Preprocessing*, *Detection of QR Code Regions*, MSD-based *Bounding Boxes Filtering*, and *Compressed MobileNet based Classification of the detected bounding boxes*. The proposed MSD strategy for box-screening has indeed improved the accuracy for rapid classification of the multi-QR codes. In addition, the Compressed MobileNet has resulted in reduced parameters for efficiency but higher recognition accuracy. In comparative experiments, we validate the superiority of our approach on the publicly available dataset. Finally, due to lower requirements on hardware resources than the existing deep learning-based approaches, our approach is conducive to the embedded implementation on edge devices to further benefit multi-QR codes detection in real-time IoT scenarios.

In order to further improve the accuracy and efficiency in multi-QR codes detection, especially with more lightweight local spatial representation, we will make reference to deep learning-based image segmentation approaches [50, 51] and local feature extraction approaches [52]. Also, to improve the reliability and explainability of the DL-based models, we will also explore multi-QR codes detection with the Explainable Artificial Intelligence [53] in trustable IoT systems, as well as to test on new version of YOLO, such as YOLOv5.

REFERENCES

[1] Zongmin Jiang, Yangming Guo, and Zhuqing Wang. Digital twin to improve the virtual-real integration of industrial IoT. *Journal of Industrial Information Integration*, 22:100196, 2021.

[2] Hossein Darvishi, Domenico Ciuonzo, Eivind Roson Eide, and Pierluigi Salvo Rossi. Sensor-fault detection, isolation and accommodation for digital twins via modular data-driven architecture. *IEEE Sensors Journal*, 21(4):4827–4838, 2020.

[3] Luiz Belussi and Nina Hirata. Fast QR code detection in arbitrarily acquired images. In *Proc. 24th SIBGRAPI Conf on Graphics, Patterns and Images*, pages 281–288. IEEE, 2011.

[4] István Szentandrási, Adam Herout, and Markéta Dubská. Fast detection and recognition of QR codes in high-resolution images. In *Proc. 28th spring conference on computer graphics*, pages 129–136, 2012.

[5] R. Zhu, Y. Lu, and D. Sun. Location and correction arithmetic of QR code based on symbol characteristic. *China Measurement & Test*, 41(2):88–90, 2015.

[6] X. Cheng H. Ren and H. Li. Research on QR code location method based on sub-pixel. *Computer Applications and Software*, 32(12):214–217, 2015.

[7] Euisung Kang. A rectification method for quick response code image. In *The 18th IEEE International Symposium on Consumer Electronics (ISCE 2014)*, pages 1–2. IEEE, 2014.

[8] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint: 1804.02767*, 2018.

[9] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *European conf. on computer vision*, pages 21–37. Springer, 2016.

[10] Xiaolong Wang, Abhinav Shrivastava, and Abhinav Gupta. A-fast-rcnn: Hard positive generation via adversary for object detection. In *Proc. IEEE conference on computer vision and pattern recognition*, pages 2606–2615, 2017.

[11] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, 39(6):1137–1149, 2017.

[12] Adnan Sharif, Guangtao Zhai, Jun Jia, Xiongkuo Min, Xiangyang Zhu, and Jiahe Zhang. An Accurate and Efficient 1-D Barcode Detector for Medium of Deployment in IoT Systems. *IEEE Internet of Things Journal*, 8(2):889–900, 2020.

[13] Jun Jia, Guangtao Zhai, Jiahe Zhang, Zhongpai Gao, Zehao Zhu, Xiongkuo Min, Xiaokang Yang, and Guodong Guo. EMBDN: An efficient multiclass barcode detection network for complicated environments. *IEEE Internet of Things Journal*, 6(6):9919–9933, 2019.

[14] Jun Jia, Guangtao Zhai, Ping Ren, Jiahe Zhang, Zhongpai Gao, Xiongkuo Min, and Xiaokang Yang. Tiny-BDN: An efficient and compact barcode detection network. *IEEE Journal of Selected Topics in Signal Processing*, 14(4):688–699, 2020.

[15] Mingxing Tan, Ruoming Pang, and Quoc V Le. EfficientDet: Scalable and efficient object detection. In *Proc. IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[16] A. Sharif, G. Zhai, X. Min, J. Jia, and K. Munir. Enhancing Decoding rate of Barcode Decoders in Complex Scenes for IoT Systems. *IEEE Internet of Things Journal*, PP(99):1–1, 2021.

[17] Binyao Jiang, Yisheng Ji, Xiaohua Tian, and Xinbing Wang. Batch Reading densely arranged QR codes. In *IEEE INFOCOM 2019-IEEE Conf. on Computer Communications*, pages 1216–1224. IEEE, 2019.

[18] Douglas Chai and Florian Hock. Locating and decoding EAN-13 barcodes from images captured by digital cameras. In *2005 5th International Conf. on Information Communications & Signal Processing*, pages 1595–1599. IEEE, 2005.

[19] Ruben Muniz, Luis Junco, and Adolfo Otero. A robust software barcode reader using the hough transform. In *Proc. 1999 International Conference on Information Intelligence and Systems (Cat. No. PR00446)*, pages 313–319. IEEE, 1999.

[20] Orazio Gallo and Roberto Manduchi. Reading challenging barcodes with cameras. In *2009 Workshop on applications of computer vision (WACV)*, pages 1–6. IEEE, 2009.

[21] Romulus Terebes, Otmane Lahlou, and Romania Enseirb. Camera phone based barcode decoding system. *ACTA Technica Napocensis*, 49(3):57–62, 2008.

[22] Gábor Sörös and Christian Flörkemeier. Blur-resistant joint 1D and 2D barcode localization for smartphones. In *Proc. 12th International Conference on Mobile and Ubiquitous Multimedia*, pages 1–8, 2013.

[23] S. Wachenfeld, S. Terlunen, and X. Jiang. Robust recognition of 1-d barcodes using camera phones. In *International Conf. on Pattern Recognition*, 2008.

[24] Ladislav Karrach, Elena Pivarčiová, and Pavol Bozek. Recognition of perspective distorted QR codes with a partially damaged finder pattern in real scene images. *Applied Sciences*, 10(21):7814, 2020.

[25] ML Hakim, S Yatmono, AC Nugraha, and M Khairudin. The Effects of Vertical and Horizontal Distance on The Performance of QR Code Detection System. In *Journal of Physics: Conference Series*, volume 1737, page 012029. IOP Publishing, 2021.

[26] Omar Lopez-Rincon, Oleg Starostenko, Vicente Alarcon-Aquino, and Juan C Galan-Hernandez. Binary large object-based approach for QR code detection in uncontrolled environments. *Journal of Electrical and Computer Engineering*, 2017, 2017.

[27] M. W. Azani and Kmmm Abdul. Binarization of Document Images: A Comprehensive Review. *Journal of Physics Conference Series*, 1019:012023, 2018.

[28] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[29] Markéta Dubská, Adam Herout, and Jiří Havel. Real-time precise detection of regular grids and matrix codes. *Journal of Real-Time Image Processing*, 11(1):193–200, 2016.

[30] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *Proc. IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[31] Alessandro Zamberletti, Ignazio Gallo, and Simone Albertini. Robust angle invariant 1d barcode detection. In *2013 2nd IAPR Asian Conf. on Pattern Recognition*, pages 160–164. IEEE, 2013.

[32] Yunzhe Xiao and Zhong Ming. 1d barcode detection via integrated deep-learning and geometric approach. *Applied Sciences*, 9(16):3268, 2019.

[33] Jiahe Zhang, Xiongkuo Min, Jun Jia, Zehao Zhu, Jia Wang, and Guangtao Zhai. Fine localization and distortion resistant detection of multi-class barcode in complex environments. *Multimedia Tools and Applications*, 80(11):16153–16172, 2021.

[34] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. Densebox: Unifying landmark localization with end to end object detection. *arXiv preprint arXiv:1509.04874*, 2015.

[35] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proc. European conference on computer vision (ECCV)*, pages 734–750, 2018.

[36] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *Proc. IEEE/CVF conference on computer vision and pattern recognition*, pages 850–859, 2019.

[37] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In *Proc. IEEE/CVF conference on computer vision and pattern recognition*, pages 840–849, 2019.

[38] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proc. IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

[39] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, Lei Li, and Jianbo Shi. Foveabox: Beyound anchor-based object detection. *IEEE Transactions on Image Processing*, 29:7389–7398, 2020.

[40] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv preprint arXiv:1602.07360*, 2016.

[41] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *Proc. IEEE conference on computer vision and pattern recognition*,

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3268636

pages 6848–6856, 2018.

[42] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[43] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetv2: Inverted residuals and linear bottlenecks. In *Proc. IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[44] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proc. IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019.

[45] Yinqiu Liu, Kun Wang, Kai Qian, Miao Du, and Song Guo. Tornado: Enabling blockchain in heterogeneous Internet of Things through a space-structured approach. *IEEE Internet of Things Journal*, 7(2):1273–1286, 2019.

[46] Lizhi Xiong, Xinwei Zhong, Neal N Xiong, and Ryan Wen Liu. QR-3S: A high payload QR code secret sharing system for industrial Internet of Things in 6G networks. *IEEE Transactions on Industrial Informatics*, 17(10):7213–7222, 2020.

[47] Peng-Cheng Huang, Chin-Chen Chang, Yung-Hui Li, and Yanjun Liu. Efficient QR code secret embedding mechanism based on hamming code. *IEEE Access*, 8:86706–86714, 2020.

[48] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proc. IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019.

[49] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985.

[50] Huimin Zhao, Zhenyu Fang, Jinchang Ren, Calum MacLellan, Yong Xia, Shuo Li, Meijun Sun, and Kevin Ren. SC2Net: a novel segmentation-based classification network for detection of Covid-19 in chest X-ray images. *IEEE Journal of Biomedical and Health Informatics*, 26(8):4032–4043, 2022.

[51] Jinchang Ren, He Sun, Huimin Zhao, Hao Gao, Calum Maclellan, Sophia Zhao, and Xiaoyu Luo. Effective extraction of ventricles and myocardium objects from cardiac magnetic resonance images with a multi-task learning U-Net. *Pattern Recognition Letters*, 155:165–170, 2022.

[52] Genyun Sun, Hang Fu, Jinchang Ren, Aizhu Zhang, Jaime Zabalza, Xiuping Jia, and Huimin Zhao. SpaSSA: Superpixelwise adaptive SSA for unsupervised spatial-spectral feature extraction in hyperspectral image. *IEEE Transactions on Cybernetics*, 2021.

[53] Alfredo Nascita, Antonio Montieri, Giuseppe Aceto, Domenico Ciuonzo, Valerio Persico, and Antonio Pescapé. XAI meets mobile traffic classification: Understanding and improving multimodal deep learning architectures. *IEEE Transactions on Network and Service Management*, 18(4):4225–4246, 2021.