RESEARCH ARTICLE

Surface Defect Detection and Size Measurement of Bearings Based on Machine Vision

Liwei Lin[1,*]

[1] School of Electrical Engineering, Yancheng Institute of Technology, Yancheng 224051, China

Abstract

Aiming at the problems of low efficiency, strong subjectivity in traditional bearing surface defect detection and insufficient dimensional measurement accuracy, this paper proposes an integrated detection scheme SimAM-YOLO that combines the improved YOLOv5 algorithm with size measurement technology. Based on YOLOv5, the scheme replaces the original C3 module with the C2F network structure and embeds the SimAM attention mechanism to enhance the model's ability to extract defect features. Combined with OpenCV, it realizes the real-time measurement of the key dimension of bearing radius and constructs a visual system for bearing size measurement. Experimental results show that the improved model achieves an average detection precision of 86.03%, a recall rate of 78%, and an mAP-0.5 of 82.17% for bearing defects such as cracks, scratches, and grooves, which are 14.8%, 8.77%, and 9.2% higher than the original YOLOv5 respectively. The dimensional measurement error is controlled within ±0.000061mm, meeting the requirements of industrial detection. The system has high automation and strong real-time performance, can adapt to the detection needs of bearings of different specifications, and provides an efficient and reliable technical support for bearing quality control.

Keywords: bearing detection, YOLOv5, C2F network, SimAM attention mechanism, machine vision, size measurement.

1 Introduction: The Paradigm Shift from Networking to Intelligence

As the core transmission component of mechanical equipment, the surface quality and dimensional accuracy of bearings directly determine the operational stability and service life of mechanical systems [1–12]. With the advancement of Industry 4.0, bearing production is developing towards high speed and precision, and traditional detection methods [13–15] can no longer meet the needs of quality control. Manual visual inspection is susceptible to subjective experience and fatigue, resulting in high missed detection rate (usually exceeding 2%) and low efficiency (single piece detection time exceeding 30s). Contact mechanical measurement (such as micrometers and vernier calipers) is complex to operate, cannot realize real-time online detection on the production line, and is likely to cause secondary damage to the bearing surface. In this context, developing non-contact detection technology based on machine vision to realize synchronous and accurate detection of bearing surface defects and dimensional parameters has become

Submitted: 05 December 2025
Accepted: 16 December 2025
Published: 09 February 2026

Vol. 1, **No.** 1, 2026.
10.62762/TICPS.2025.819879

***Corresponding author:**
✉ Liwei Lin
linliwei8796@126.com

Citation
Lin, L. (2026). Surface Defect Detection and Size Measurement of Bearings Based on Machine Vision. *ICCK Transactions on Intelligent Cyber-Physical Systems*, 1(1), 10–25.

a key breakthrough direction to improve bearing production quality and efficiency, which has important engineering value for promoting the automation upgrade of the machinery manufacturing industry.

The application of machine vision technology [16–18] in the field of bearing detection has become a research hotspot, and related research has focused on the optimization of defect recognition algorithms and the improvement of size measurement methods [19–27]. Early studies mostly adopted traditional image processing technologies, such as edge detection based on Sobel operator [28] and Canny operator [29], combined with threshold segmentation to achieve defect localization. However, these methods have poor adaptability to complex backgrounds and small-size defects (such as microcracks with width less than 0.1mm), and are prone to false detection due to illumination and noise interference. In recent years, deep learning target detection algorithms have gradually become mainstream, and models such as YOLO series [30] and Faster R-CNN [31] have been widely used. Among them, YOLOv5 [32] has become a common model for bearing defect detection due to its balance of detection speed and accuracy. However, existing research still has bottlenecks: first, the unified recognition accuracy of the model for multiple types of defects (cracks, scratches, pits) is insufficient, and the recall rate of small-size defects is generally lower than 85%; second, defect detection and size measurement are mostly independent systems, requiring separate hardware platforms and data processing modules to be built, resulting in low integration and increased industrial application costs; third, some systems are complex to operate and lack lightweight visual interfaces, making it difficult to meet the needs of production line workers to get started quickly.

To address the above research bottlenecks, this paper conducts the following research work around the integrated demand of bearing surface defect detection and size measurement: first, build a machine vision detection platform, complete the selection and communication connection of industrial cameras, light sources, lenses and other hardware, and determine the three-level system architecture of "image acquisition - data processing - result output"; second, improve the YOLOv5 defect detection model, replace the original C3 module with the C2F network structure to enhance gradient flow, and embed the SimAM [33] attention mechanism to strengthen the attention to defect area features, thereby improving the recognition ability of multiple types and small-size defects; third, design a size measurement algorithm based on Python-OpenCV, and realize the real-time calculation of bearing radius, area and other parameters through image preprocessing, contour extraction and ratio calibration.

The main contributions of this paper are as follows:

1. Propose the SimAM-YOLO improved model, which increases the average detection precision of small-size defects to 86.03%, the recall rate to 78%, and the mAP-0.5 to 82.17%.

2. Design a size measurement algorithm based on Python-OpenCV, which can realize real-time calculation of bearing radius through image preprocessing, contour extraction and ratio calibration.

2 Related Work

2.1 Experimental Platform Construction

To realize the synchronous development of bearing surface defect detection and size measurement, this paper completes the overall platform scheme design, core hardware selection and communication connection configuration based on system requirements, and constructs a stable and reliable basic architecture for machine vision detection.

Combined with the actual requirements of bearing detection in industrial production scenarios, we summarize four core requirements of the system. First, real-time performance: it is necessary to realize real-time acquisition and processing of bearing images, the detection time of a single frame image should not exceed 20ms, support continuous detection of 1-2 bearings per second on the production line, and can real-time mark and reject defective bearings. Second, accuracy: the false detection rate of defect detection should be less than 1% and the missed detection rate should be less than 0.5%, and the dimensional measurement error should be controlled within ±0.03mm to meet the bearing industry precision standards. Third, flexibility: it can adapt to bearings of different specifications with inner diameter of 20-50mm and outer diameter of 30-80mm, and the specification switching can be completed by replacing the lens or adjusting the working distance without reconstructing the system. Fourth, usability: the operation interface should be intuitive and simple, supporting functions such as image upload, real-time preview and result export, and operators can use it independently after simple training.

The system adopts a three-level architecture of "image acquisition - data processing - result output", with clear functions and collaborative linkage of each module: the image acquisition module is responsible for obtaining high-quality bearing surface images, capturing the front and side surface information of bearings through industrial cameras, and transmitting them to the computer via USB 3.0 interface with a transmission rate of 5Gbps to ensure no delay and no distortion of images; the data processing module, as the core module, includes two sub-units: defect detection and size measurement. The former identifies defects such as cracks, scratches and pits on the bearing surface through the improved YOLOv5 model, and the latter completes the calculation of parameters such as radius and area based on the OpenCV algorithm. Both share image acquisition data to achieve parallel processing; the result output module displays the detection results through a visual interface, including defect type, confidence, position coordinates and dimensional parameters, and also supports abnormal data alarm (such as excessive defects and out-of-tolerance size) and detection record storage (Excel format export) to facilitate quality traceability.

As the core component of image acquisition, the industrial camera needs to balance the requirements of resolution and frame rate. As shown in Figure 1, this paper comprehensively compares the performance parameters of different models and selects the Daheng Mercury II MER-500-7UC-L industrial camera. It adopts a 1/2.5-inch CMOS sensor with an effective pixel of 5 million (2592×1944) and a maximum frame rate of 30FPS, which can meet the clear capture of bearing detailed features; it supports global shutter to avoid motion blur and adapt to the bearing transmission speed (0.5m/s) of the production line; the dynamic range is 68dB, which can maintain the gray level of images under different illumination conditions, providing a high-quality data foundation for subsequent defect recognition.

The selection of optical lens needs to match the camera sensor size and the field of view (FOV) of detection. According to the imaging optical principle, combined with the requirements of the maximum outer diameter (80mm) and working distance (30cm) of the bearing to be detected, a 16mm focal length fixed-focus industrial lens is selected. The distortion rate of the lens is less than 1%, ensuring no distortion of the bearing edge contour; the aperture range is F1.4 - F16, which can adjust the light intake according to the light source intensity to further optimize the image clarity. The light source adopts an annular LED light source with an inner diameter of 50mm and an outer diameter of 100mm, and a light-emitting angle of 45°. It provides uniform illumination for the bearing surface through diffuse reflection, effectively eliminating the shadow interference caused by the bearing surface and highlighting the gray difference between defects and the background; the light source controller supports 0-100% brightness adjustment, which can adapt to the light reflection characteristics of bearing materials (such as stainless steel and bearing steel). In addition, a high-precision stage is configured to support X/Y axis fine-tuning (precision 0.01mm), ensuring that the bearing is placed in the center during detection and avoiding detection errors caused by position offset.

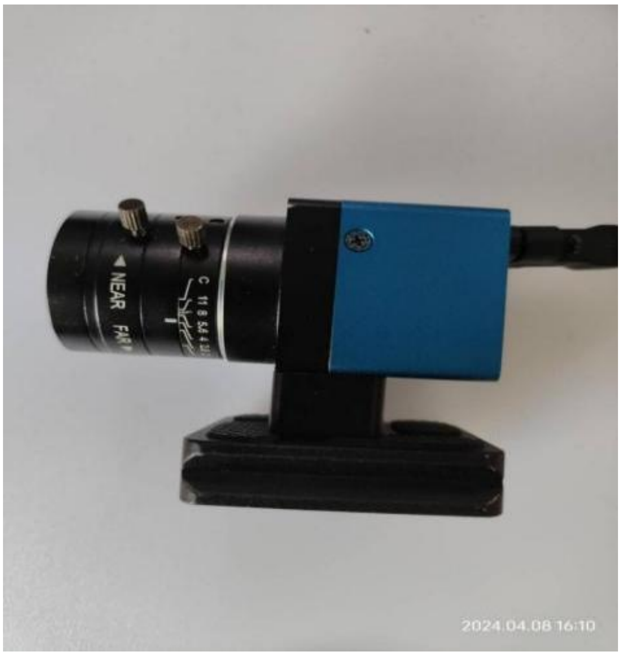


Figure 1. Daheng Mercury II MER-500-7UC-L industrial camera.

2.2 Attention Mechanism

In the technological iteration of Convolutional Neural Networks (CNN), the research on attention mechanism has always focused on breaking through the inherent constraints of the local receptive field of CNN. Its core idea is to dynamically adjust the feature weight distribution, strengthen the representation efficiency of key visual information, and thus become a core technical direction to improve the adaptability of CNN in complex scenarios. From the development context, the Spatial Transformer Network (STN) proposed in 2015 can be regarded as the early exploration form in this field. Although the model did not explicitly adopt the term "attention", it can realize adaptive focusing and alignment of key image regions through end-to-end learning of deformable spatial transformation matrices, providing the core idea of "selective attention" for the design of subsequent attention mechanisms and becoming an important theoretical cornerstone for attention research in the visual field.

It was not until 2017 that the proposal of Squeeze-and-Excitation Networks (SENet) [34] marked the formal formation of channel attention mechanism. The model innovatively designed a two-stage operation of "global average pooling (squeeze stage) - fully connected layer modeling (excitation stage)", which dynamically learned the dependency relationship between feature channels to generate weight coefficients, significantly improving the model's attention to highly discriminative feature channels. Relying on this design innovation, SENet showed superior performance over traditional CNN in the ImageNet image classification competition, which not only verified the technical feasibility of integrating attention mechanism with CNN, but also promoted the large-scale application of attention technology in various visual tasks.

Since 2018, the attention mechanism has entered a stage of rapid development with multiple dimensions and diversification, and the research focus has shifted from single-channel optimization to global dependency modeling and multi-dimensional collaborative optimization. Among them, Non-Local Neural Networks [35] first introduced the self-attention mechanism into the visual field. By calculating the correlation similarity between any two points in the feature map, the model effectively captures long-distance contextual information, making up for the lack of global semantic modeling ability of CNN due to local convolution operations, and providing a new technical solution for tasks that require global information support such as target detection and video analysis. The Convolutional Block Attention Module (CBAM) [36] proposed in 2019 further achieved technical breakthroughs, innovatively constructing a "channel-space" dual-branch attention structure: first, the channel attention branch quantifies and weights the importance of different feature channels to screen out channels carrying key information; then, the spatial attention branch locates and enhances the salient regions in the channel feature map, forming a collaborative optimization mechanism of "channel screening - spatial focusing". This hybrid attention module has the characteristics of light weight and easy embedding, and can be flexibly integrated into various CNN architectures, showing excellent generalization performance in various tasks such as image classification, target detection and semantic segmentation. At the same time, the Efficient Channel Attention Network (ECA-Net) [37] optimized the information redundancy problem caused by the fully connected layer in SENet, replacing the fully connected operation with 1D convolution, which greatly reduces the computational complexity while retaining local channel interaction information, further improving the application efficiency of the channel attention mechanism in resource-constrained scenarios.

In recent years, the research on attention mechanism has shown two core evolution trends: "lightweight design" and "task customization". In resource-constrained scenarios such as mobile terminals and embedded systems, models such as ShuffleAttention [38] adopt the technical strategy of "channel grouping - shuffle interaction - local attention calculation", which maintains the feature representation ability while reducing redundant parameters and computational load, realizing the "lightweight" and efficient deployment of the attention module; in specific visual tasks, the attention mechanism begins to deeply integrate with task characteristics. For example, in target detection tasks, the attention module is designed to have a weight distribution logic that suppresses background noise and interfering targets, and in semantic segmentation tasks, cross-scale attention is used to fuse multi-level contextual information to adapt to the differentiated needs of different tasks for feature representation. At the same time, inspired by the global modeling ability of Vision Transformer (ViT), a new generation of visual models such as CoAtNet and ConvNeXt [39, 40] further break the technical barrier between CNN and self-attention, adapting to image multi-scale features through a hierarchical attention mechanism: retaining the advantage of CNN in local detail extraction in the bottom network, and introducing self-attention in the top network to model global semantic associations, forming a "local-global" collaborative feature learning mode, and promoting the continuous improvement of the performance of visual models in complex scenarios.

At present, the attention mechanism has developed from an early auxiliary optimization method to a core component of modern convolutional networks. Its technical evolution path clearly presents the logical main line of "from single-dimensional optimization to multi-dimensional collaboration, from generalized design to task customization, and from performance priority to efficiency-performance balance". Looking forward to the future, with the continuous deepening of the demand for edge computing, real-time vision and other scenarios, the attention mechanism will further develop towards "high efficiency (low parameter quantity, low computational overhead)" and "scenarioization (adapting to specific tasks and hardware environments)", continuing to provide core support for technological breakthroughs in the field of convolutional images.

2.3 YOLOv5

Since the proposal of the YOLO (You Only Look Once) series algorithms in 2016, they have attracted wide attention in the field of target detection. The core idea of this series of algorithms is to transform the target detection task into a regression problem, and obtain the bounding box and category probability of the target through a single forward propagation, thereby realizing high-speed and accurate detection. With the continuous development of the YOLO series algorithms, their performance has been significantly improved. From the initial YOLOv1 to the subsequent YOLOv2, YOLOv3, and now YOLOv4 and YOLOv5, each generation of algorithms has been optimized and improved in terms of structure, loss function, training strategy and other aspects. Due to the excellent performance of the YOLOv5 algorithm in terms of speed and accuracy, it has a wide range of application advantages in the field of target detection, so this paper adopts the YOLOv5 algorithm as the main module for defect detection. Especially in the task of bearing surface defect detection, the YOLOv5 algorithm can give full play to its advantages and realize efficient and accurate defect detection.

The high speed of YOLOv5 enables it to complete the detection task of a large number of bearing images in a short time, meeting the application scenarios with high real-time requirements. This is of great significance for real-time detection of bearing surface defects on the production line. Secondly, the accuracy of YOLOv5 ensures the reliability of the detection results. By optimizing the algorithm structure and parameter settings, YOLOv5 can accurately identify various defect types on the bearing surface and provide accurate defect position and size information. This provides strong support for subsequent defect analysis and processing. In addition, the usability of YOLOv5 allows users to easily apply it to practical projects. Through simple configuration and training, users can use the YOLOv5 algorithm to realize the bearing surface defect detection task without complex algorithm development and debugging work.

2.4 Edge Detection Algorithm

Edge detection is a core technology in the fields of image processing and computer vision, aiming to identify regions in images with significant brightness changes, i.e., edges. These edges often carry key features or important event information of the image. Edges can be detected by calculating the gradient change of the image. Because the Canny edge detection algorithm has good noise suppression and edge detection performance, this paper uses it as the edge detection algorithm for bearings.

Edge detection based on the Canny operator mainly relies on image filtering and pixel gradient calculation. Image filtering is usually used to eliminate or reduce noise, blur or other unwanted details in the image, while retaining or enhancing important information in the image. In digital image processing, filtering operations usually involve applying a filter to each pixel in the image. This filter can be predefined or dynamically generated according to the image content. The application method of the filter usually involves convolving the filter with a small region (such as a 3x3 or 5x5 pixel block) in the image, and then assigning the result of the operation to the central pixel of the region. Figure 2 shows the comparison between the median filtered image and the original image, and we adopt median filtering as the filter for our images. The main advantage of median filtering is that it can effectively filter out impulse noise (such as salt and pepper noise) and other random noise, while keeping the edge information of the image from being blurred. Compared with mean filtering, median filtering is more excellent in protecting image details. In addition, median filtering is not sensitive to the statistical characteristics of noise in the input signal, making it show good robustness in processing various noises.

We adopt the Sobel operator as the core algorithm for pixel gradient calculation. The Sobel operator occupies an important position in image processing, especially in the field of edge detection. It realizes edge detection by calculating the approximate value of the gradient of the image brightness function. Applying the Sobel operator to any point in the image can obtain

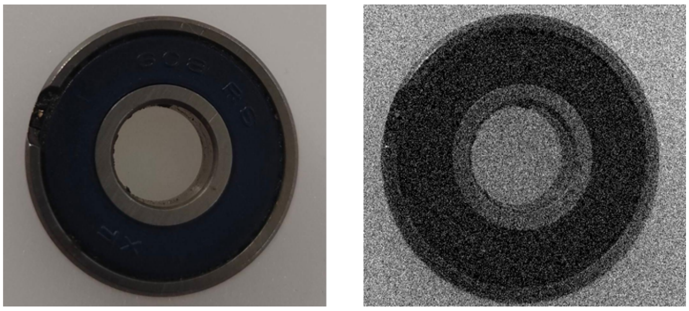
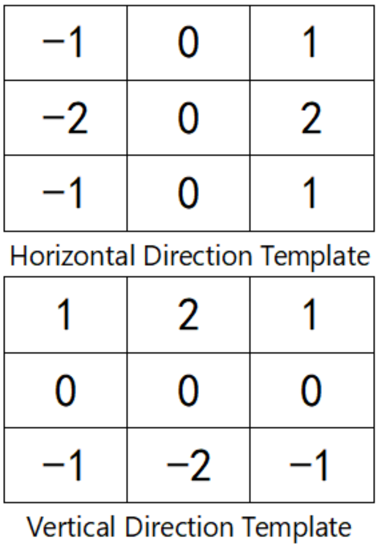


Figure 2. Comparison between the original image and the image after median filtering.

the gradient vector or its norm at that point. This operator mainly performs first-order or second-order differential operations on each pixel of the image based on various possible reasons for edge formation, so as to accurately identify points with significant brightness changes, i.e., edges.

The Sobel operator has two directional templates, namely horizontal and vertical direction templates, as shown in Figure 3.



−1	0	1
−2	0	2
−1	0	1

Horizontal Direction Template

1	2	1
0	0	0
−1	−2	−1

Vertical Direction Template

Figure 3. Direction templates of the Sobel operator.

Calculation process of the Sobel operator: first, use the above two templates to calculate the gradients of the original image I in the horizontal and vertical directions by Equation (1):

$$G_X = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I, \quad G_Y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * I \tag{1}$$

(1) Its expanded calculation formula is as Equation (2):

$$\begin{aligned} \Delta_x f(x,y) = {} & [f(x-1,y+1) + 2f(x,y+1) + f(x+1,y+1)] \\ & - [f(x-1,y-1) + 2f(x,y-1) + f(x+1,y-1)] \end{aligned} \tag{2}$$

$$\begin{aligned} \Delta_y f(x,y) = {} & [f(x-1,y-1) + 2f(x-1,y) + f(x-1,y+1)] \\ & - [f(x+1,y-1) + 2f(x+1,y) + f(x+1,y+1)] \end{aligned} \tag{3}$$

(2) Finally, the gradient of the entire image is calculated by Equation (3):

$$G = \sqrt{G_x^2 + G_y^2} \tag{4}$$

The Sobel operator skillfully combines the advantages of the Prewitt operator and the Roberts operator, which can not only effectively suppress noise, but also provide relatively accurate edge positioning information. It determines the position and direction of edges by calculating the approximate value of the gradient of the image brightness function and combining the gray weighted difference of the upper, lower, left and right neighboring pixels of the pixel. This strategy makes the Sobel operator perform well in practical applications, especially when the accuracy requirement is not very high. The Sobel operator can handle noise well and obtain relatively accurate edge information, so it has been widely used in the field of image processing.

3 Proposed Methods

3.1 Network Structure

As one of the core tasks in the field of computer vision, target detection has a wide range of applications in intelligent monitoring, autonomous driving and many other fields. The YOLO series algorithms have become the mainstream choice for real-time target detection tasks due to their end-to-end detection process and efficient inference speed. To further improve its detection performance, this paper proposes an improved method based on the YOLO architecture, which mainly realizes the improvement of feature expression ability and detection accuracy through two key improvements: introducing the SimAM attention mechanism in the neck and replacing the C3 module with the C2F module. The final network structure is shown in Figure 4.

The improved YOLO network still follows the classic three-stage architecture of Backbone-Neck-Head. The Backbone is responsible for extracting multi-scale

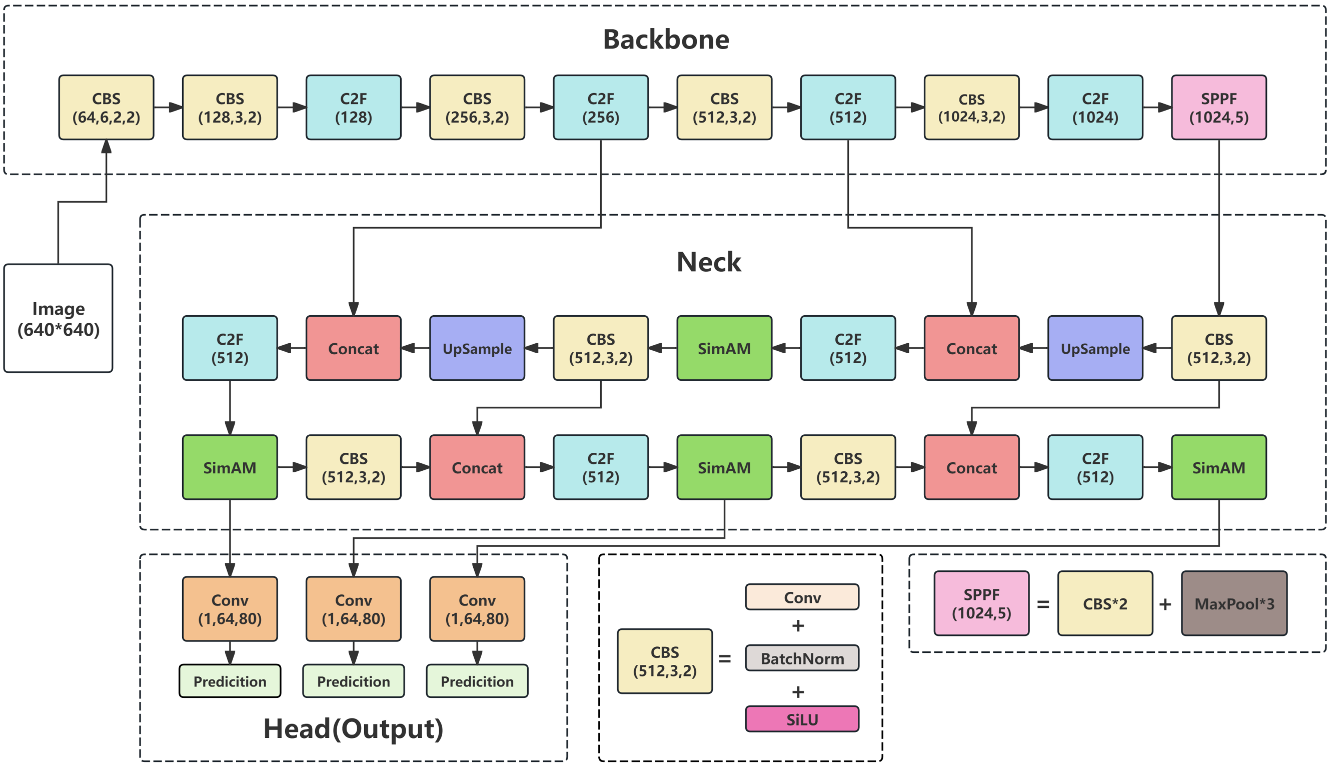


Figure 4. Network structure of SimAM-YOLO.

features from the input image. The C2F network structure is a special network design mainly used for computer vision tasks, especially target detection tasks. Its core idea is to convert the output of the convolutional layer into the input of the fully connected layer, so as to better integrate and utilize feature information at different levels. Specifically, the C2F structure usually includes a series of convolutional layers, which are used to extract features from the input image. Then, these features are converted into a format suitable for processing by the fully connected layer through a certain method (such as pooling or other feature integration methods). The fully connected layer further processes and classifies these features to generate the final detection results. In this paper, the original C3 module in YOLO is replaced with the C2F module, and the comparison of their network structures is shown in Figure 5. Compared with the C3 network structure, the Channel of the input Tensor entering the Bottleneck calculation sequence is only 0.5 times that of the input channel of C2F, so the computational complexity is significantly reduced. On the other hand, the increase in gradient flow can also significantly improve the convergence speed and effect, so the effect is better. In the construction process of feature maps of different scales (such as 128, 256, 512, 1024 channel dimensions), the alternate combination of CBS (Conv+BatchNorm+SiLU) units and C2F modules realizes the gradual extraction of multi-scale features from the input image (640×640). In addition, the SPPF (Spatial Pyramid Pooling - Fast) module is introduced at the end of the Backbone to fuse features of different receptive fields through multi-scale pooling operations, further improving the global perception ability of features.

As a key link of feature fusion, the core improvement of the Neck in this paper lies in introducing the SimAM attention mechanism and combining the feature fusion ability of the C2F module. As a parameter-free attention method, SimAM models the importance of neurons in the feature map, adaptively strengthens key features and suppresses redundant information. In the multi-scale feature fusion path of the Neck, SimAM is inserted into the key feature layers, so that after operations such as upsampling and concatenation of features, the network can more accurately focus on the effective features of the target region.

First, SimAM extracts the feature map of the input image through a Convolutional Neural Network (CNN). Each position in the feature map represents specific information of a part of the image. In a convolutional neural network, these feature maps usually contain rich spatial information, which is

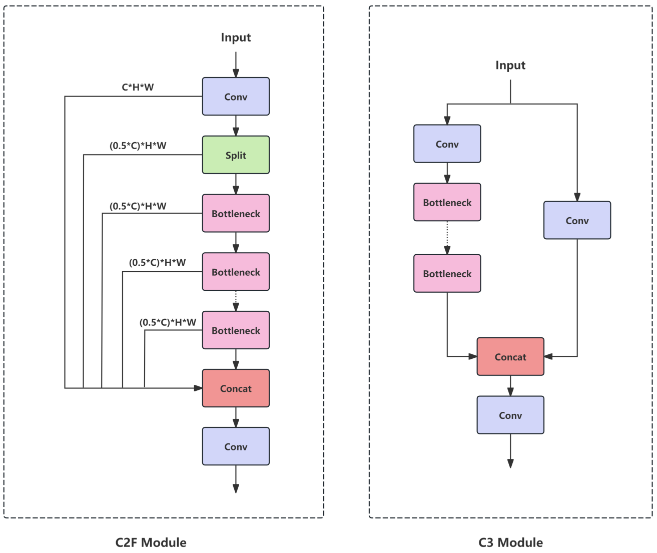


Figure 5. Structural comparison diagram of C2F and C3 modules.

crucial for tasks such as image recognition and classification. Next, SimAM calculates the attention weight by using the local self-similarity of the feature map. In an image, adjacent pixels usually have strong similarity, while pixels at a long distance have weak similarity. Based on this characteristic, SimAM calculates the similarity between each pixel and its adjacent pixels to generate attention weights. Specifically, SimAM first calculates the square of the difference between each pixel and the average value of its channel, which helps to capture the difference between the pixel and its surrounding area. Then, these difference values are divided by a regularization term (i.e., the sum of squares of each channel plus a small constant e_lambda), and 0.5 is added to obtain a new tensor y. This operation aims to enhance the weight of pixels that are significantly different from the surrounding pixels, while suppressing the weight of pixels that are similar to the surrounding pixels. Finally, SimAM multiplies the original feature map by y processed by the Sigmoid activation function to obtain the weighted feature map. This process actually applies an attention weight to each pixel in the original feature map, making the model pay more attention to the key features that are significantly different from the surrounding pixels. In general, the SimAM model generates attention weights by calculating the local self-similarity of the feature map, thereby guiding the model to focus on the key regions in the image. This lightweight and parameter-free attention mechanism can effectively improve the performance of the convolutional neural network, making the model perform better in tasks such as target detection and image classification.

In the feature fusion process, the Neck realizes cross-scale feature fusion through UpSample operation, and combines the feature integration ability of the C2F module and the attention guidance of SimAM to repeatedly fuse the multi-scale features output by the Backbone, and finally generates three enhanced feature maps of different scales, providing more discriminative feature input for the detection task of the Head.

The Head part adopts a fully convolutional structure, and maps the feature maps output by the Neck to detection results through Conv layers. Each detection branch includes a convolutional layer with 64 channels, and finally outputs three types of prediction results (corresponding to target detection requirements of different scales), realizing the prediction of the category, position and confidence of the target.

3.2 Real-time Bearing Size Measurement

3.2.1 Overall Measurement Scheme Design

The flow chart of the real-time bearing size measurement designed in this paper is shown in Figure 6: first, set the type of camera to be called. Before starting the measurement, it is necessary to select an appropriate camera type according to the actual application scenario, such as a USB camera or an industrial camera, and configure corresponding parameters such as resolution and frame rate. The main function of this step is to facilitate the program to switch between the built-in camera and the external camera. Next, call the camera. Use functions or classes in the Python-OpenCV library to initialize and start the camera so that it can capture real-time video streams. This step is the basis of real-time measurement, ensuring that continuous image data of the bearing can be obtained.

In the image processing stage, preprocess and extract features from the captured bearing images. This usually includes steps such as grayscale conversion, filtering and denoising, and edge detection to more accurately identify the contour and feature points of the bearing. In contour endpoint detection, contour detection algorithms provided by OpenCV can be used to find the endpoints of the bearing contour, which are crucial for subsequent size measurement.

Then, perform border drawing (data calculation). According to the found contour endpoints, draw the corresponding border on the image and calculate the

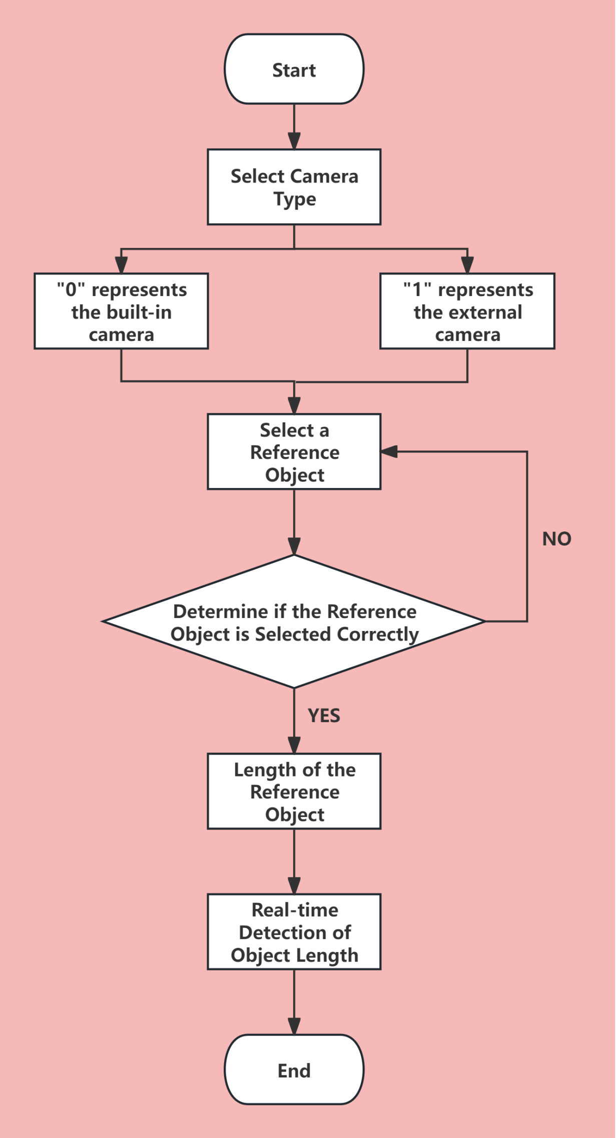

Start
Select Camera Type
"0" represents the built-in camera
"1" represents the external camera
Select a Reference Object
NO
Determine if the Reference Object is Selected Correctly
YES
Length of the Reference Object
Real-time Detection of Object Length
End

Figure 6. Flow chart of real-time bearing size measurement.

size data of the border. These size data can be pixel values, but usually need to be converted into actual physical sizes according to the calibration parameters of the camera. The main purpose of this part is to draw the frame and calculate the data. Next is ratio calculation. In real-time measurement, it may also be necessary to calculate the ratio according to a reference object of known size to convert pixel size into actual size. This can be achieved by introducing a reference object of known size (such as a standard bearing or a calibration plate) during the measurement process.

Reference object selection (photography) is an important part of ratio calculation. Before the measurement starts, it is necessary to select a suitable reference object and take its image with a camera. Then, through image processing and analysis, find the size information of the reference object and use it as the benchmark for subsequent size measurement.

Finally, perform real-time measurement. In the real-time video stream captured by the camera, perform the above processing and analysis on each frame of image, calculate and output the real-time size data of the bearing. This can be achieved by presenting the size data to the user in the form of numbers, charts or real-time displays.

Through the above process, the real-time bearing size measurement system based on Python-OpenCV can efficiently and accurately complete the size measurement task, providing strong support for fields such as industrial automation and quality control.

3.3 Image Processing

This paper adopts the Canny edge detection algorithm to realize the accurate positioning of image edges, effectively reducing noise interference and the false detection rate of non-edge pixels. The specific implementation process is as follows: first, convert the input RGB color image into a grayscale image through the cv2.cvtColor function to lay the foundation for subsequent edge detection; second, use a Gaussian kernel of size $(5, 5)$ to filter the grayscale image (where the standard deviation of the Gaussian function is set to 0, and OpenCV will automatically calculate this parameter according to the kernel size) to further smooth the image and suppress noise; then, perform Canny edge detection based on double thresholds (min_val and max_val). Pixels with values lower than min_val are judged as non-edge pixels, those higher than max_val are judged as edge pixels, and pixels between the two thresholds are judged as edge pixels only when they are connected to pixels higher than max_val, otherwise they are classified as non-edge pixels; considering the requirement of edge continuity, this study only performs cv2.dilate dilation operation on the detected image (without corrosion operation) to achieve edge thickening and connection of discontinuous edges; then, call the cv2.findContours function to extract contours from the dilated binary image, and only retain the outermost contours, which are represented in the form of compressed horizontal, vertical and diagonal line segments (only the endpoints of the line segments are retained); since there are differences in the return value formats of the cv2.findContours function between OpenCV 3 and OpenCV 4, the return results of the two versions need to be unified through the contour extraction step; finally, the function outputs a list of detected contours, where each contour is presented in the form of a set of points, corresponding to an edge in the image or the boundary of a target object.

3.3.1 Border Drawing

The specific flow of this part is shown in Figure 7. The core functions are frame drawing and data calculation: in the initial stage of the program, when taking a photo of the reference object, the selected object is clearly identified by drawing a frame; as the program progresses, on the basis of continuously drawing this frame, the length and area in the real world are determined through a series of ratio calculations; finally, while completing the frame drawing, the program intuitively presents the calculated results such as length and area to the user. Considering that bearings are mainly circular structures, it is only necessary to extract the coordinates of the leftmost and rightmost points of the contour to meet the calculation needs of relevant parameters.

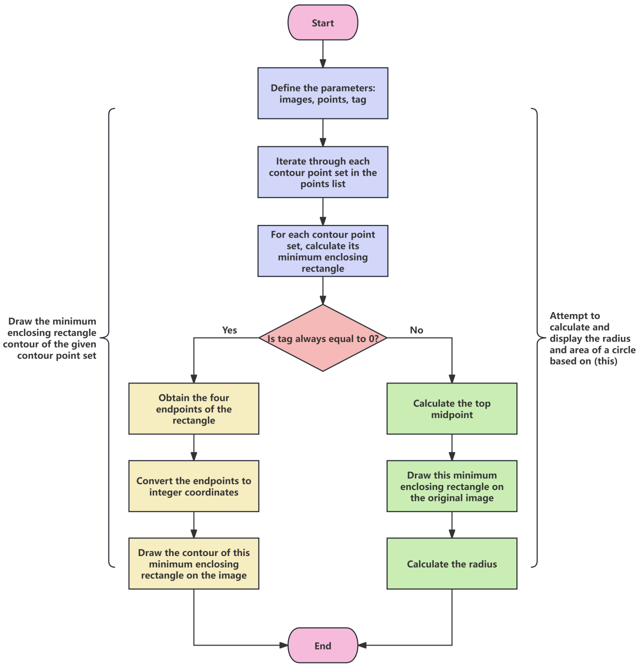

Start
Define the parameters: images, points, tag
Iterate through each contour point set in the points list
For each contour point set, calculate its minimum enclosing rectangle
Is tag always equal to 0?
Yes
No
Draw the minimum enclosing rectangle contour of the given contour point set
Obtain the four endpoints of the rectangle
Convert the endpoints to integer coordinates
Draw the contour of this minimum enclosing rectangle on the image
Attempt to calculate and display the radius and area of a circle based on (this)
Calculate the top midpoint
Draw this minimum enclosing rectangle on the original image
Calculate the radius
End

Figure 7. Flow chart of the border drawing module.

3.3.2 Ratio Calculation

The ratio is defined as the ratio of the distance between two points of the reference object in the metric space to the actual distance between these two points. All dimensions to be calculated are obtained by dividing the Euclidean distance by this ratio. Among them, the Euclidean distance is a classic method in mathematics for measuring the "ordinary" straight-line distance between two points in multi-dimensional space. It is named after the ancient Greek mathematician Euclid, and this distance represents the shortest straight-line distance between two points. Specifically, the calculation formula of the Euclidean distance between two points $A(x_1, y_1)$ and $B(x_2, y_2)$ in a two-dimensional plane is as shown in Equation (4):

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{5}$$

The calculation formula of the Euclidean distance between two points $A(x_1, y_1, z_1)$ and $B(x_2, y_2, z_2)$ in a three-dimensional space is as shown in Equation (5):

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{6}$$

The calculation formula of the Euclidean distance between two points $A = (a_1, a_2, \cdots, a_n)$ and $B = (b_1, b_2, \cdots, b_n)$ in an N-dimensional space is as shown in Equation (6):

$$d = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \tag{7}$$

The specific implementation scheme of this part is as follows: instead of directly calculating the distance of the actual object in the physical world, first solve the Euclidean distance between two reference points (left_point and right_point) in the coordinate system (denoted as length_euclidean); then, the code prompts the user to input the length of the reference object based on known standards or information (denoted as length_reference), which may correspond to a certain size of the actual object in the physical world, but the code itself does not contain the corresponding relationship between the two; then, the ratio (denoted as rate) is obtained by calculating the ratio of the Euclidean distance to the length of the reference object. This ratio does not directly represent the distance of the actual object in the physical world, but serves as a scaling factor for estimating the proportional relationship of other relevant dimensions according to the known length of the reference object. It should be noted that the above estimation process is based on the assumption that "the image size scaling is uniform and there is no perspective distortion", but this assumption may not hold in actual scenarios, so the actual object distance calculated by this method is only an approximate value.

3.3.3 *Real-time Detection*

The core function of this part is to realize real-time measurement. Algorithm 1 summarizes the pseudocode of the real-time measurement algorithm in this part, and its specific execution process is as follows: first, initialize the camera, create a camera object camera, which usually represents the device ID of the camera in the form of an integer (for example, 0 corresponds to the built-in camera, 1 corresponds to the external camera, etc.); then the function enters an infinite main loop to continuously capture video frames from the camera and perform processing and display operations on each frame of image. Inside the loop, the first step is to capture a frame, obtain a frame of image from the camera through a specific method, which returns a boolean value indicating whether the capture is successful (usually True means success) and an image frame (stored in the form of a NumPy array). Since only the image frame is needed, it is extracted through the [1] index; the second step is to flip the image, call the function to horizontally flip the captured image frame to make the image consistent with the camera preview direction (because the camera preview image is usually in a mirror state); the third step is to obtain contour points, process the flipped image and return the contour endpoints of all reference objects, which can characterize the boundaries or specific feature points of objects in the image; the fourth step is to filter contour points, first initialize an empty list selected_points to store the filtered contour points, then traverse each contour in the points list through a list comprehension, calculate the area of each contour, and if the contour area is greater than the preset threshold, add it to the selected_points list; the fifth step is to draw the border, draw the border corresponding to all contours in the selected_points list on the original image; the sixth step is to display the image, and display the image with the drawn border in a window named "Camera"; finally, perform the operation of exiting the loop to end the real-time measurement process.

Algorithm 1: Real-time measurement algorithm

Input: Camera device ID, display mode
Output: Processed image with detected contours

```
// Define the real-time processing function
begin
    PRINT "Enter the real-time processing program"
    // Initialize the camera object: parameters are camera device type and display flag
    camera $\leftarrow$ Initialize video capture object (camera device ID, display mode)
    // Enter the real-time processing main loop
    while True do
        frame $\leftarrow$ Read image frame from camera   // Capture a frame of image from the camera, only get the image frame data
        flipped_frame $\leftarrow$ Perform horizontal flip operation on frame   // Horizontally flip the image
        points $\leftarrow$ Process flipped_frame   // Process the flipped image to obtain the contour endpoints of all reference objects
        selected_points $\leftarrow$ Empty list   // Filter contours: retain contours with area greater than the threshold
        foreach contour $i$ in points do
            if area of contour $i \geq$ preset area threshold then
                Add contour $i$ to selected_points
            end
        end
        Draw the border corresponding to selected_points on frame   // Draw the border of the filtered contour on the original image
        Display image window ("Camera", frame)   // Display the processed image (window name is "Camera")
        if (key input is detected) or (window "Camera" is closed) then
            // Detect exit conditions: any key is pressed or the window is closed
            Close the "Camera" window
            break
        end
    end
end
```

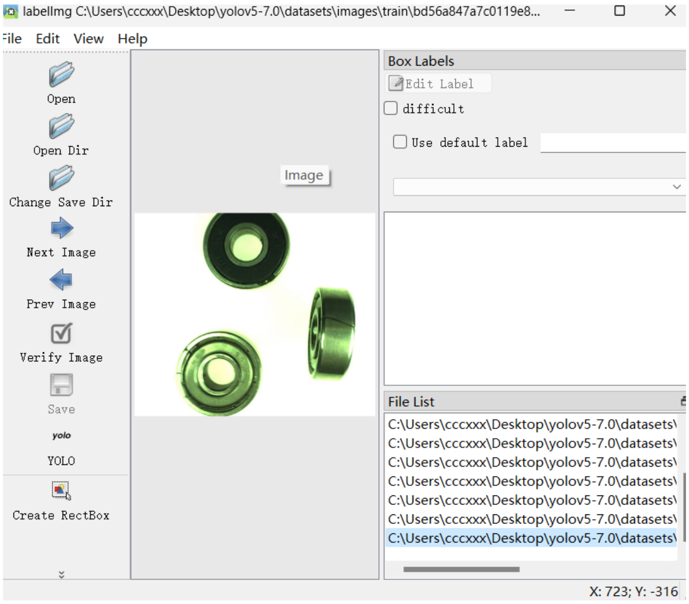
4 Experiments and Results

In this paper, a general bearing test dataset is used to evaluate the SimAM-YOLO model designed in this paper.

4.1 Experimental Setup

The code runs on the Windows 11 system environment, using the NVIDIA GeForce RTX 4060 graphics card. During the experimental training process, the SGD optimizer is adopted, the initial learning rate is 0.01, and the weight decay coefficient is 0.0005. The confidence threshold is set to 0.5 for mAP-50 and 0.95 for mAP-95. Each training cycle of the model is 300 epochs, the batch size is 32, and the image input size is 640×640 pixels.

Data collection is a key step in building a bearing surface defect detection system. To train a high-performance YOLOv5 model, we collect a large number of bearing surface images containing various defect types. These images can be obtained through laboratory shooting, production line collection or public datasets. Place the camera above the bearing to be detected, first perform image collection and save it in the specified path on the computer; then, open the LabelImg software in the YOLOV5 environment to perform image coordinate calibration, as shown in Figure 8. It should be noted that the more images included in the training set, the better, which is conducive to improving the accuracy.


labelImg C:\Users\cccxxx\Desktop\yolov5-7.0\datasets\images\train\bd56a847a7c0119e8...

C:\Users\cccxxx\Desktop\yolov5-7.0\datasets\
C:\Users\cccxxx\Desktop\yolov5-7.0\datasets\
C:\Users\cccxxx\Desktop\yolov5-7.0\datasets\
C:\Users\cccxxx\Desktop\yolov5-7.0\datasets\
C:\Users\cccxxx\Desktop\yolov5-7.0\datasets\
C:\Users\cccxxx\Desktop\yolov5-7.0\datasets\
C:\Users\cccxxx\Desktop\yolov5-7.0\datasets\

X: 723; Y: -316

Figure 8. Coordinate calibration.

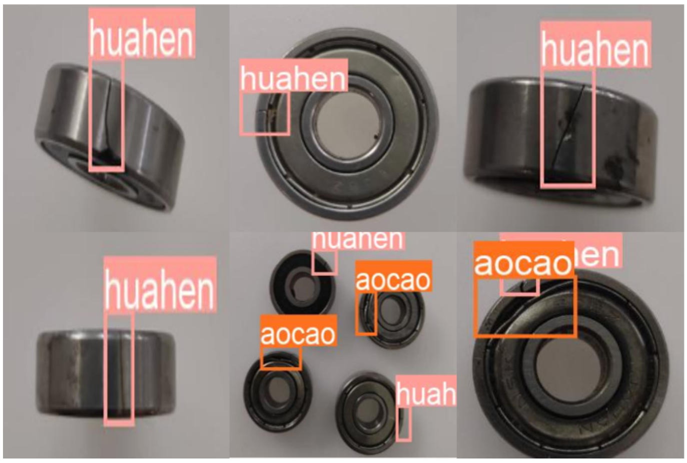


Figure 9. Scratch detection effect.

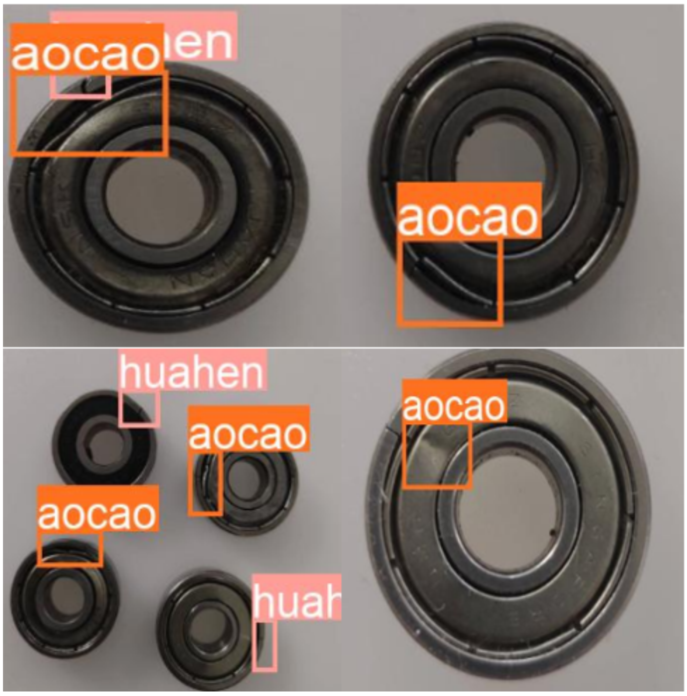


Figure 10. Groove detection effect.

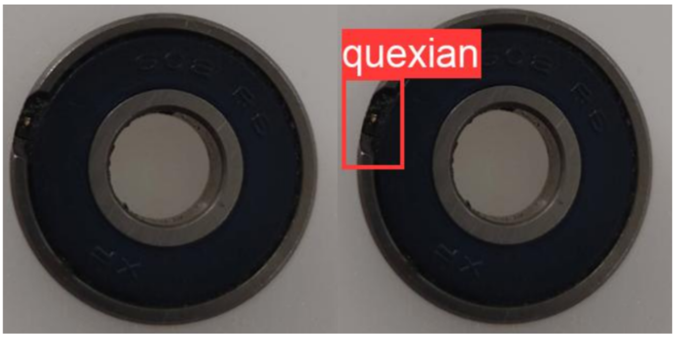


Figure 11. Defect detection effect.

This paper adopts Precision (Metrics/Precision), Recall (Metrics/Recall), mean average precision (mAP-0.5 & mAP-0.5:0.95) and frame rate (FPS) as core performance indicators. Among them, Precision is used to measure the accuracy of positive sample prediction, reflecting the proportion of correctly identified among all positive predictions; Recall evaluates the model's ability to capture all relevant instances, indicating the proportion of correctly identified among actual positive samples. Accuracy can be calculated by combining True Positive (TP), True Negative (TN), False Negative (FN) and False Positive (FP) through Equation (9). The Average Precision (AP) is defined by Equation (10), where R_n and P_n represent the recall rate and precision

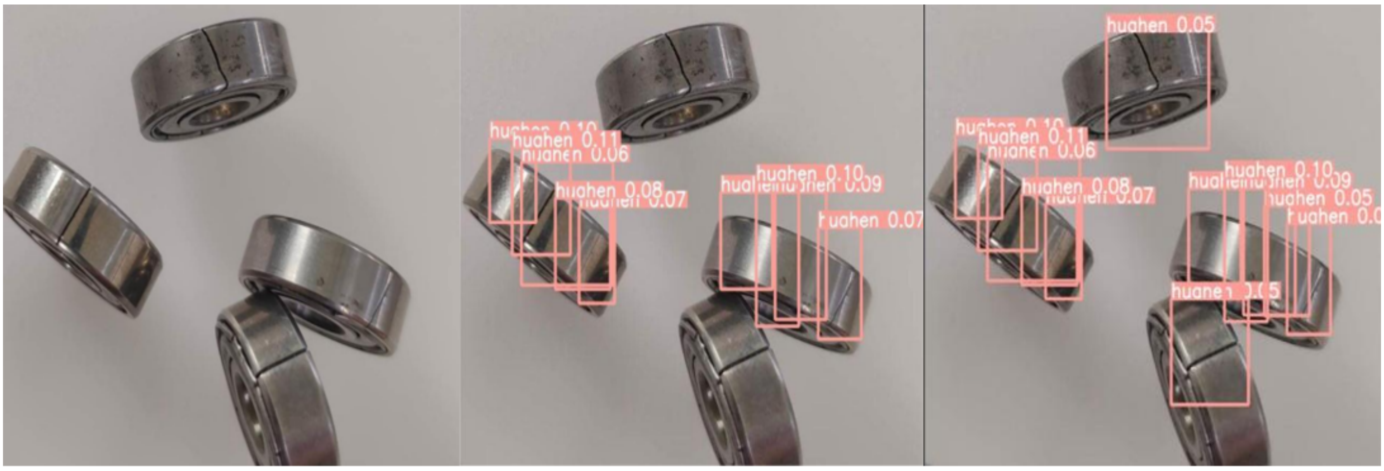


Figure 12. Original image, YOLOv5 detection image and SimAM-YOLO detection image.

corresponding to the nth threshold respectively. The mean Average Precision (mAP) is the average result of the AP values of all instances, which is equivalent to the area under the Precision-Recall (P-R curve). Among them, mAP-0.5 & mAP-0.5:0.95 as a comprehensive indicator provides a comprehensive evaluation of model quality by calculating the average precision across multiple categories. These indicators together constitute a comprehensive evaluation of model effectiveness, balancing accuracy, efficiency and comprehensiveness, making it suitable for a wide range of applications from autonomous driving to medical image analysis. The mathematical expressions of these indicators are as shown in Equations (7)-(10):

$$\text{precision} = \frac{TP}{TP + FP} \tag{8}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{9}$$

$$\text{accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \tag{10}$$

$$AP = \sum_{n} (R_n - R_{n-1}) P_n \tag{11}$$

4.2 Feature Performance of Detection Results

In this paper, the characteristic manifestations of various common defects on the bearing surface in images are divided into three categories: grooves (aocao), defects (quexian), and scratches (huahen). Specifically, the scratch detection results are shown in Figure 9; the groove detection results are shown in Figure 10; and the defect detection results are shown in Figure 11. It can be seen that our algorithm can clearly distinguish various defects, achieve no missed detection and no false detection, and the accuracy of the detection frame is controlled at the millimeter level.

4.3 Comparison After Introducing the SimAM Attention Mechanism

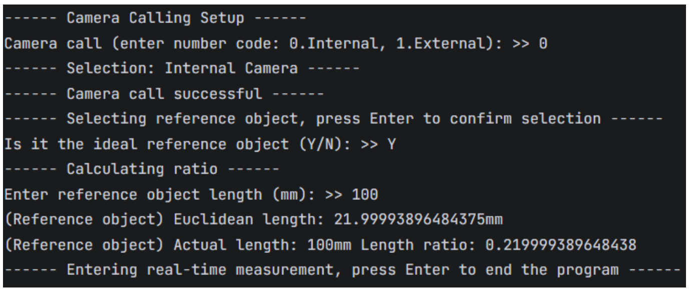
After introducing the SimAM mechanism, the model will be significantly improved. Taking scratches as an example, the detection effect is shown in Figure 12. It can be seen that the performance of the SimAM-YOLO model is better than that of the original YOLOv5 model, and its performance in scratch recognition is significantly improved. Table 1 shows the quantitative analysis results of the two models. The data shows that SimAM-YOLO surpasses the original YOLOv5 model with higher mean average precision (mAP) in all three categories. Specifically, its precision is improved by 12.1%, 14.2% and 18.1% in scratches, pits and grooves respectively, and the recall rate is also significantly improved by 12.9%, 4.8% and 8.6%.

Table 1. Quantitative comparison between the original YOLOv5 and SimAM-YOLO.

Model	Defect type	Precision (%)	Recall (%)	Map-50 (%)	Map50-95 (%)
YOLOv5	Scratch	73.1	71.8	69.4	39.3
	Defect	75.3	72.4	81.2	47.5
	Groove	65.3	63.5	68.3	39.6
SimAM-YOLO	Scratch	85.2	84.7	81.9	50.2
	Defect	89.5	77.2	88.2	52.9
	Groove	83.4	72.1	76.4	50.8

4.4 Real-time Monitoring Implementation Effect

The code terminal implementation flow chart and the measurement result chart are shown in Figures 13 and 14 respectively. For bearings, we use BJK-608RS deep groove ball miniature bearings as the detection



```
------ Camera Calling Setup ------
Camera call (enter number code: 0.Internal, 1.External): >> 0
------ Selection: Internal Camera ------
------ Camera call successful ------
------ Selecting reference object, press Enter to confirm selection ------
Is it the ideal reference object (Y/N): >> Y
------ Calculating ratio ------
Enter reference object length (mm): >> 100
(Reference object) Euclidean length: 21.99993896484375mm
(Reference object) Actual length: 100mm Length ratio: 0.219999389648438
------ Entering real-time measurement, press Enter to end the program ------
```

Figure 13. Flow chart of real-time size detection.

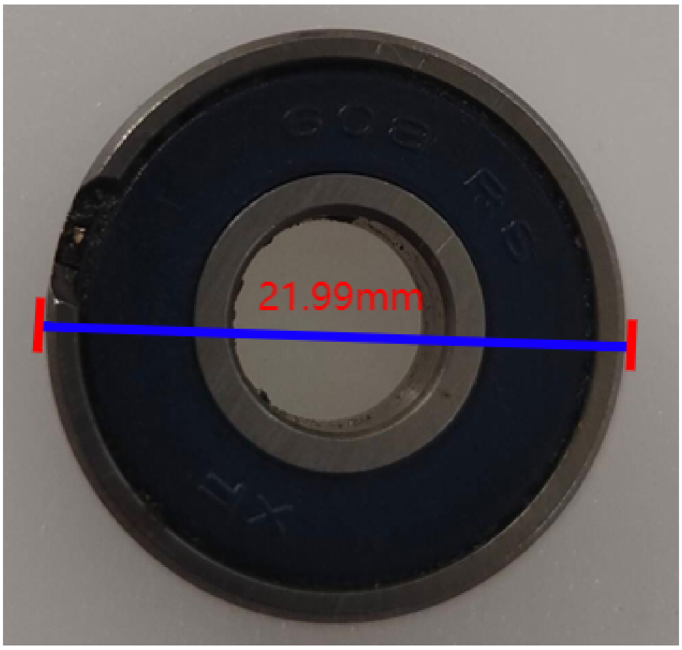


Figure 14. Measurement result chart of real-time size detection.

object, with specifications of outer diameter 22mm and inner diameter 8mm. In comparison, it can be seen that the error of the algorithm designed in this paper in real-time bearing size is only 0.000061mm, which can efficiently and accurately complete the size measurement task, providing strong support for fields such as industrial automation and quality control.

5 Conclusion

Aiming at the industry pain points of strong subjectivity, low efficiency in traditional bearing surface defect detection and insufficient dimensional measurement accuracy, this paper proposes an integrated detection scheme SimAM-YOLO that combines the improved YOLOv5 algorithm with machine vision size measurement technology, realizing the synchronous and efficient completion of bearing surface defect recognition and key dimensional parameter measurement.

This paper first builds a three-level machine vision detection platform of "image acquisition - data processing - result output", completes the selection and integration of core hardware such as industrial cameras, optical lenses and annular LED light sources, and provides hardware support for high-quality image acquisition and real-time detection; at the algorithm level, based on the YOLOv5 framework, the original C3 module is replaced with the C2F network structure to enhance gradient flow and reduce computational complexity, and the SimAM parameter-free attention mechanism is embedded to strengthen the characterization of defect area features, constructing a SimAM-YOLO defect detection model with both detection speed and accuracy; at the same time, a size measurement algorithm including image preprocessing, Canny edge detection, contour extraction, Euclidean distance calculation and ratio calibration is designed based on Python-OpenCV to realize real-time and accurate calculation of bearing radius, area and other parameters, and a visual detection interface is developed to improve the usability of the system.

Experimental results show that the SimAM-YOLO model achieves an average detection precision of 86.03%, a recall rate of 78%, and an mAP-0.5 of 82.17% for three typical bearing defects: scratches, defects and grooves, which are 14.8%, 8.77% and 9.2% higher than the original YOLOv5 respectively; the measured error for BJK-608RS deep groove ball bearings is only 0.000061mm, which fully meets the requirements of industrial detection accuracy. In addition, the system realizes 100% reuse of the hardware platform for defect detection and size measurement, supports the detection of bearings of different specifications with outer diameters of 30-80mm, is easy to operate and flexible to deploy, and effectively reduces the cost of industrial application.

The innovation of this paper lies in proposing the technical path of "feature enhancement - attention guidance - integrated detection", improving the defect recognition ability through the collaborative optimization of the C2F module and the SimAM attention mechanism, and realizing the deep integration of detection and measurement functions. Future research can further optimize the feature adaptation strategy of the attention mechanism, correct the measurement error caused by perspective deformation by combining binocular vision or laser ranging technology, and explore the lightweight deployment scheme of the model on embedded devices to meet the real-time detection needs of more complex industrial scenarios.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Harris, T. A., & Crecelius, W. J. (1986). Rolling bearing analysis. [CrossRef]

[2] Bhushan, B. (2013). *Principles and applications of tribology*. John wiley & sons. [CrossRef]

[3] Malamas, E. N., Petrakis, E. G., Zervakis, M., Petit, L., & Legat, J. D. (2003). A survey on industrial vision systems, applications and tools. *Image and vision computing, 21*(2), 171-188. [CrossRef]

[4] Lynagh, N., Rahnejat, H., Ebrahimi, M., & Aini, R. (2000). Bearing induced vibration in precision high speed routing spindles. *International Journal of Machine Tools and Manufacture, 40*(4), 561-577. [CrossRef]

[5] Kumar, S., Goyal, D., Dang, R. K., Dhami, S. S., & Pabla, B. S. (2018). Condition based maintenance of bearings and gears for fault detection–A review. *Materials Today: Proceedings, 5*(2), 6128-6137. [CrossRef]

[6] Bearings—Damage, R. (2017). Failures—Terms, Characteristics and Causes. *International Organization for Standardization: Geneva, Switzerland*.

[7] Raj, K. K., Kumar, S., & Kumar, R. R. (2025). Systematic review of bearing component failure: Strategies for diagnosis and prognosis in rotating machinery. *Arabian Journal for Science and Engineering, 50*(8), 5353-5375. [CrossRef]

[8] Patil, M. S., Mathew, J., Rajendrakumar, P. K., & Desai, S. (2010). A theoretical model to predict the effect of localized defect on vibrations associated with ball bearing. *International Journal of Mechanical Sciences, 52*(9), 1193-1201. [CrossRef]

[9] Tandon, N., & Choudhury, A. (1999). A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings. Tribology International, 32(8), 469-480. [CrossRef]

[10] Sun, J., Yan, C., & Wen, J. (2017). Intelligent bearing fault diagnosis method combining compressed data acquisition and deep learning. *IEEE Transactions on Instrumentation and Measurement, 67*(1), 185-195. [CrossRef]

[11] Vansia, D. A., & Patel, M. (2025, May). Defect Detection in Bearings Using Machine Learning and Computer Vision. In *2025 Global Conference in Emerging Technology (GINOTECH)* (pp. 1-5). IEEE. [CrossRef]

[12] Sanjrani, A. N., Huang, H. Z., Shah, S. A., Hussain, F., Punhal, M., Narejo, A., & Zhang, B. (2025). High-speed train wheel set bearing analysis: Practical approach to maintenance between end of life and useful life extension assessment. *Results in Engineering, 25*, 103696. [CrossRef]

[13] Golnabi, H., & Asadpour, A. (2007). Design and application of industrial machine vision systems. *Robotics and Computer-Integrated Manufacturing, 23*(6), 630-637. [CrossRef]

[14] Dron, J.-P., Bolaers, F., & Rasolofondraibe, L. (2010). A comparative experimental study on the use of three denoising methods for bearing defect detection. *Meccanica, 45*(2), 265–277. [CrossRef]

[15] Liu, M., Zhang, W., Jiang, L., & Shen, J. (2024). Bearing-detr: A lightweight deep learning model for bearing defect detection based on rt-detr. *Sensors, 24*(13), 4262. [CrossRef]

[16] Chen, B., Zhao, Y., Zhang, Y., Jiang, Y., Zhang, H., & Pan, H. (2024). Analysis of hot spots and trends in rolling bearing fault diagnosis research based on scientific knowledge mapping. *Engineering Research Express, 6*(2), 025536. [CrossRef]

[17] Deng, S., Cai, W., Xu, Q., & Liang, B. (2010, October). Defect detection of bearing surfaces based on machine vision technique. In *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)* (Vol. 4, pp. V4-548). IEEE. [CrossRef]

[18] Le, H. F., Zhang, L. J., & Liu, Y. X. (2022). Surface defect detection of industrial parts based on YOLOv5. *IEEE Access, 10*, 130784-130794. [CrossRef]

[19] Zhang, W., Liu, M., Jiang, L., & Shen, J. (2025). Research on Complex Defect Detection Method on Steel Surface based on EBA-YOLO. *IAENG International Journal of Computer Science, 52*(7).

[20] Zhang, R., Liu, D., Bai, Q., Fu, L., Hu, J., & Song, J. (2024). Research on X-ray weld seam defect detection and size measurement method based on neural network self-optimization. *Engineering Applications of Artificial Intelligence, 133*, 108045. [CrossRef]

[21] Wen, S., Chen, Z., & Li, C. (2018). Vision-based surface inspection system for bearing rollers using convolutional neural networks. *Applied Sciences, 8*(12),

2565. [CrossRef]

[22] Ping, Z., Chuangchuang, Z., Gongbo, Z., Zhenzhi, H., Xiaodong, Y., Shihao, W., ... & Bing, H. (2023). Whole surface defect detection method for bearing rings based on machine vision. *Measurement Science and Technology, 34*(1), 015017. [CrossRef]

[23] Nabhan, A., Mousa, M., & Ghazaly, N. (2015). Bearing fault detection techniques-a review. *Turkish Journal of Engineering, Sciences and Technology, 3*(2), 1–18.

[24] Ding, P., Zhang, W., & Liu, M. (2024). A bearing surface defect detection method based on multi-attention mechanism Yolov8. *Measurement Science and Technology, 35*(8), 086003. [CrossRef]

[25] Song, K. K., Zhao, M., Liao, X., Tian, X., Zhu, Y., Xiao, J., & Peng, C. (2022, February). An improved bearing defect detection algorithm based on yolo. In *2022 International Symposium on Control Engineering and Robotics (ISCER)* (pp. 184-187). IEEE. [CrossRef]

[26] Wang, Y., Xia, H., Yuan, X., Li, L., & Sun, B. (2018). Distributed defect recognition on steel surfaces using an improved random forest algorithm with optimal multi-feature-set fusion. *Multimedia Tools and Applications, 77*(13), 16741-16770. [CrossRef]

[27] Wang, H., Yang, J., & Hu, Z. (2011). Current status and prospect of roller bearing surface defect detection. *Procedia Engineering, 15*, 4331–4336. [CrossRef]

[28] Han, L., Tian, Y., & Qi, Q. (2020). Research on edge detection algorithm based on improved sobel operator. In *MATEC Web of Conferences* (Vol. 309, p. 03001). EDP Sciences. [CrossRef]

[29] Yuan, L., & Xu, X. (2015). Adaptive image edge detection algorithm based on canny operator. In *2015 4th International Conference on Advanced Information Technology and Sensor Application (AITS)* (pp. 1–4). IEEE. [CrossRef]

[30] Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of Yolo algorithm developments. *Procedia Computer Science, 199*, 1066–1073. [CrossRef]

[31] Girshick, R. (2015, December). Fast R-CNN. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)* (pp. 1440-1448). [CrossRef]

[32] Horvat, M., Jelečević, L., & Gledec, G. (2022). A comparative study of YOLOv5 models performance for image localization and classification. In *Central European Conference on Information and Intelligent Systems* (pp. 349-356). Faculty of Organization and Informatics Varazdin.

[33] Yang, L., Zhang, R.-Y., Li, L., & Xie, X. (2021). Simam: A simple, parameter-free attention module for convolutional neural networks. In *International Conference on Machine Learning* (pp. 11863–11874). PMLR.

[34] Hu, J., Shen, L., Albanie, S., Sun, G., & Wu, E. (2019). Squeeze-and-Excitation Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 42*(8), 2011-2023. [CrossRef]

[35] Wang, X., Girshick, R., Gupta, A., & He, K. (2018, June). Non-local Neural Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7794-7803). IEEE. [CrossRef]

[36] Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018, September). CBAM: Convolutional Block Attention Module. In *European Conference on Computer Vision* (pp. 3-19). Cham: Springer International Publishing. [CrossRef]

[37] Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., & Hu, Q. (2020, June). ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 11531-11539). IEEE Computer Society. [CrossRef]

[38] Zhang, Q. L., & Yang, Y. B. (2021, June). Sa-net: Shuffle attention for deep convolutional neural networks. In *ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 2235-2239). IEEE. [CrossRef]

[39] Dai, Z., Liu, H., Le, Q. V., & Tan, M. (2021). Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems, 34*, 3965–3977.

[40] Liu, Z., Mao, H., Wu, C. Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022, June). A ConvNet for the 2020s. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 11966-11976). IEEE. [CrossRef]

Liwei Lin received the B.Eng. degree in Electronic Information Engineering from Yancheng Institute of Technology, Yancheng, China, in 2024. He is currently pursuing the M.S. degree with Yancheng Institute of Technology, Yancheng, China. (Email: linliwei8796@126.com)