ISSN: 2688-8645

Research on YOLO-based Component Detection Algorithms

Haojie Zhu[1, *], Ziyi Li[1], Yuedong Lang[2], Yongqi Li[1] and Yunlong Liu[1]

[1]School of Mechanical Engineering, Tianjin University of Technology and Education, Tianjin 300222, China

[2]Beijing Aerospace Control Instrument Research Institute, Beijing 100854, China

*Corresponding author

Abstract

Components are the foundation of machine manufacturing, and domestic manufacturing enterprises require a wide variety of parts in large quantities during production. In industrial production, the inspection of parts is a critical step. This paper primarily addresses the issue of part inspection, focusing on industrial components as the research subject. Building upon the research and analysis of object detection in the field of deep learning, the YOLOv5 detection model is optimized. The experimental results indicate that the optimized model exhibits high accuracy and detection efficiency. It also demonstrates the superiority of deep learning-based object detection methods in the field of irregular precision component inspection, compared to traditional object detection algorithms.

Keywords

Deep Learning; Industrial Parts; Target Detection; YOLOv5.

1. Introduction

In recent years, with the rapid development of information technology and the manufacturing industry in China, artificial intelligence technologies such as deep learning have simultaneously driven the advancement of computer vision technology. Currently, vision systems integrated with deep learning technologies have become a new focal point of research [1]. With the rapid development of deep learning technology, industrial inspection algorithms based on deep learning have been widely applied [2]. Traditional part inspection methods typically rely on manually designed feature extraction and classifiers, but they face certain challenges in detecting industrial parts. Compared to traditional object detection algorithms, deep learning-based object recognition methods can avoid the drawbacks of conventional approaches, overcome their performance bottlenecks, and improve the speed and accuracy of part recognition. It can automatically, quickly, and efficiently recognize various types of parts, greatly enhancing the efficiency and accuracy of part inspection. This research provides a reliable automated part inspection method for precision manufacturing and maintenance industries, improving production efficiency and quality control. It holds significant importance for industrial part recognition.

Deep learning-based object detection is divided into two categories based on the detection approach: two-stage detection [3][4]and single-stage detection[5][6]. The former is a "coarse-to-fine" process, while the latter is an end-to-end "one-step" process [7]. Typically, two-stage detection has higher accuracy in localization and object recognition, while single-stage detection is faster [8]. Single-stage detection attempts to directly classify each region of interest as either background or object [9]. That is, it directly outputs the object's class probability and location coordinates through a single stage. A typical example is YOLO (You Only Look Once), among others.

ISSN: 2688-8645

In recent years, object detection has become a popular direction in computer vision and digital image processing. Many scholars in China have explored the application of object detection algorithms in various fields. Based on the original YOLO algorithm, various types of YOLO algorithms have been improved. For example, the improved algorithms have been applied in fields such as vehicle detection and recognition, mask-wearing detection, emergency medicine packaging boxes, etc. [10]-[12]. By utilizing computer vision, these methods reduce the reliance on human capital, which has significant practical implications.

2. Introduction to the Dataset and Part Detection with Dataset Creation

2.1. Introduction to the Dataset

A dataset is a collection of interrelated data samples, widely used in core fields such as information retrieval, data mining, and machine learning. These samples come in various forms, including images, videos, and texts, providing the foundation for training and testing machine learning algorithms and models. Researchers can continuously optimize and improve the predictive ability and accuracy of models through datasets. In the field of object detection, one of the core elements is data. Computers need to analyze a large number of data features to train and build accurate models. Therefore, the quality of the dataset has a decisive impact on the performance and accuracy of machine learning algorithms and models.

2.2. Precision Part Detection and Dataset Creation

2.2.1. Data Collection

This paper utilizes the YOLO object detection algorithm to identify industrial parts. The selected parts for recognition include gears, bearings, bolts, and nuts, considering factors such as shape, size, and material. Considering the application of parts in different fields, the dataset is created by randomly placing the four types of parts or using inconsistent background environments during image capture. The data collection includes images of single parts, combinations of multiple parts, as well as images with object occlusion, as shown in Figure 1.

Figure 1. Example images of single parts, multiple parts, and part occlusion

During the part detection process, challenges such as similar part shapes, part stacking, and unclear features are often encountered as shown in Figure 2. To address these issues, we used a camera to capture a variety of images, covering scenarios such as similarity, overlap, and dense clutter. The camera was set to capture images from multiple angles and different distances, further enhancing the generalization ability of the dataset. At the same time, to better extract image features and prevent model overfitting, data augmentation techniques such as adjusting brightness and exposure were applied to some images, enriching the image dataset. This ensures that our dataset can fully reflect the various complex situations that may be encountered during actual detection.

ISSN: 2688-8645

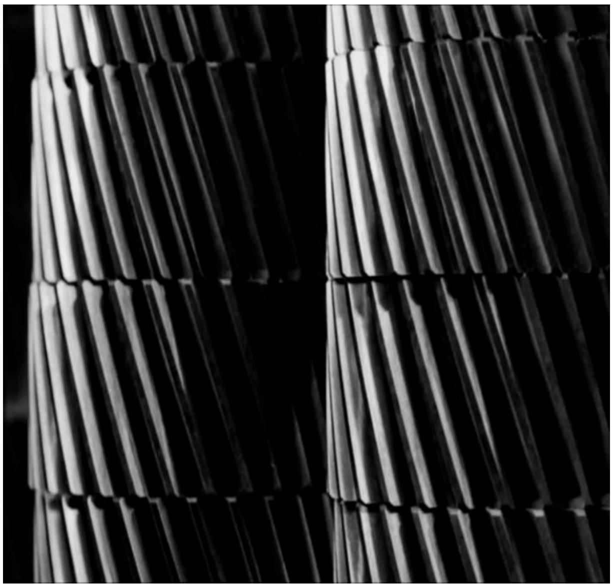


Figure 2. Example image of part stacking

2.2.2. Data Annotation

After the industrial part data collection is completed, the collected dataset should be annotated. Before training the YOLOv5 model, annotating the collected data is an important task. It involves accurately labeling the images in the dataset to generate annotation files. The main goal of this process is to provide detailed information such as the category and location of objects in the images, so that the YOLO object detection model can accurately recognize the objects in the images. Therefore, the quality of our annotations directly affects the performance of the model's detection results. This paper uses the Labelimg annotation tool to annotate the dataset and selects the YOLO annotation format to label the collected image data as shown in Figure 3. The annotated images are saved as txt files, with the annotated information including the object category and location. This includes four annotation categories as shown in Figure 4: Bearing (category 0), Gear (category 1), Bolt (category 2), and Nut (category 3). The annotated dataset is then classified and organized, ensuring that the file names correspond one-to-one with the image names. The dataset is then randomly divided into a 90:10 split. 90% of the annotated images are used as the training set.

Figure 3. Diagram of the part calibration process

Figure 4. Diagram of annotation categories

3. YOLOv5-based Industrial Part Detection Algorithm

3.1. The Principle of the YOLOv5 Algorithm

In 2020, Ultralytics optimized and upgraded YOLOv4, launching the YOLOv5 version. The YOLOv5 algorithm consists of four different model sizes: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. These four models are primarily distinguished by their differences in depth and width, making them suitable for different application scenarios and performance requirements [13].

The YOLOv5 object detection algorithm is a deep learning-based object detection algorithm that uses Convolutional Neural Networks (CNN) for feature extraction and classification. It achieves end-to-end object detection through Single Shot Detection. The basic principle of the algorithm is to divide the image into grids of different sizes and make predictions for each grid to achieve object detection.

3.2. Features of the YOLOv5 Model

(1) Lightweight: Compared to YOLOv4, YOLOv5 uses a smaller model, which reduces the consumption of computational resources and storage space while maintaining high accuracy.

(2) Faster: YOLOv5 improves processing speed compared to its predecessor YOLOv4, efficiently handles high-resolution images while maintaining accuracy, and uses a more streamlined computational load and fewer parameters for object detection tasks. As a result, it achieves faster inference speeds, significantly outperforming other object detection models.

(3) Real-time Performance: Through optimization of the network structure and the use of more efficient model designs, YOLOv5 achieves faster inference speeds while maintaining high accuracy.

(4) Multi-scale Detection: YOLOv5 introduces a multi-scale detection mechanism that uses features from different scales to detect objects in the image. This allows the model to comprehensively cover objects and improves the accuracy and robustness of object detection.

(5) Strong Versatility: YOLOv5 can run on multiple platforms, including PCs, mobile devices, and embedded systems. It can also be applied to various object detection tasks, detecting and classifying multiple different object categories, including vehicle detection, object detection, etc.

In summary, these features enable YOLOv5 to achieve high real-time performance and accuracy when handling complex scenarios, with a broad range of potential applications in the field of object detection.

3.3. YOLOv5 Model Optimization

This study uses the YOLOv5s object detection model from the YOLOv5-7.0 version. The focus is on the detailed optimization of the dataset annotation for the irregular precision part detection model. In the early stages, there were issues with some images being incorrectly annotated or inaccurately labeled. Therefore, the dataset underwent a re-labeling process. Next, Python code was used to apply data augmentation techniques to the images, such as the Mosaic augmentation method. This method randomly selects four images, applies random scaling, cropping, and rearrangement, and finally stitches them together to create a new image. The generated new image contains multiple different scenes and objects, thereby increasing the

ISSN: 2688-8645

diversity and richness of the dataset. It also helps the model generalize better and reduces overfitting.

3.4. Experimental Design for Industrial Part Detection

3.4.1. Experimental Setup

The hardware and software configurations used in the irregular precision part detection experiment are shown in Table 1.

Table 1. Experimental Configuration

Name	Parameter
Operating System	Windows10 (64)
CPU	Intel(R) Core(TM) i5-1135G7 @2.40GHz
GPU	RTX 3080Ti，12GB
Anaconda	Anaconda3 2019.10
Deep Learning Environment	Pytorch

3.4.2. Experimental Process and Parameter Settings

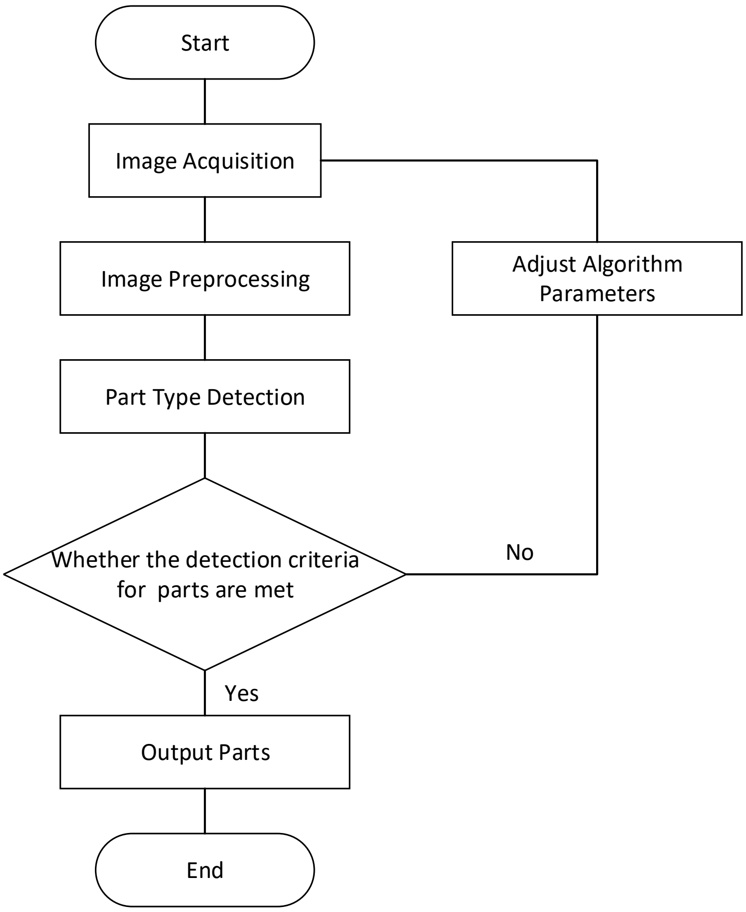


Figure 5. Part Detection System Flowchart

After the dataset is prepared and organized, the next crucial stage is training and testing the YOLOv5 model. The main goal is to ensure that the algorithm can converge efficiently and quickly, and the training process will be monitored and evaluated using a series of metrics. Additionally, to ensure a smooth and efficient training process, it is important to closely monitor the program's operating status and continuously adjust the algorithm parameters based on the results and model performance to achieve better detection results. The flowchart of the detection system in this paper is shown in Figure 5. First, different types of irregular precision parts are photographed using a camera to create the dataset for parts. To improve the model's generalization ability and recognition accuracy, Python code is used to apply data augmentation to some images. Then, the Labelimg tool is used to manually annotate the completed irregular precision part dataset, ensuring that each part is accurately identified and classified. After annotation, the dataset is organized to ensure data quality and consistency. The dataset is randomly split into training and testing sets, and then the part dataset is input into the YOLOv5 algorithm model for training and testing. Finally, after training and testing, the target part recognition results are obtained. The parameters for algorithm training are as follows: the number of training epochs is set to 300, the image resolution is 640×640 pixels, and the batch size is set to 6. The Stochastic Gradient Descent (SGD) optimizer is used during training, with an initial learning rate of 0.01, a momentum parameter set to 0.937, and a final learning rate of 0.01. Mosaic data augmentation is also applied.

4. Experimental Evaluation Metrics and Results Analysis

4.1. Experimental Evaluation Metrics

When evaluating the experimental results of a target detection model, the selection of evaluation metrics is crucial, as they serve as the main basis for assessing the performance of a model. For evaluating the performance of the target detection model, this paper uses various commonly used evaluation metrics to provide a comprehensive assessment of the YOLOv5 target detection model. These metrics mainly include Accuracy, Precision, Recall, Intersection over Union (IoU), Precision-Recall (P-R) curve, F1 score, Average Precision (AP), mean Average Precision (mAP), and Frames Per Second (FPS) for detection speed.

(1) Accuracy: Accuracy is a fundamental metric for assessing model performance. It represents the proportion of correctly classified samples to the total number of samples. The higher the accuracy, the better the model's performance, as it can more accurately identify the categories of the samples. The calculation process is shown in Equation 1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

(2) Precision: Precision, also known as positive predictive value, measures the proportion of true positive samples among all samples predicted as positive by the model. The higher the precision, the more true positive instances there are among the samples predicted as positive by the model. The calculation process is shown in Equation 2.

$$\mathrm{Pr}ecision = \frac{TP}{TP + FP} \tag{2}$$

(3) Recall: Recall, also known as sensitivity, measures the proportion of actual positive samples that are correctly identified by the model. The higher the recall, the more true positive samples the model has identified. The calculation process is shown in Equation 3.

$$\mathrm{Re}call = \frac{TP}{TP + FN} \tag{3}$$

(4) mean Average Precision: mAP (mean Average Precision) is one of the most important evaluation metrics in object detection. It measures the average performance of the model across all classes. The calculation of mAP involves the Precision-Recall (P-R) curve, where the Average Precision (AP) for each class is computed, and then the average of the APs across all classes is taken to obtain the final mAP. The calculation process is shown in Equation 4.

$$mAP = \frac{1}{n}\sum_{i=1}^{n}\int_{0}^{1} p(r)dr \tag{4}$$

4.2. Analysis of Experimental Results

4.2.1. Model Performance Testing

After training the YOLOv5 model, the performance of the model is tested by detecting videos of irregular precision parts and by testing the model on the irregular precision parts test set. The detection results of the irregular precision parts model are shown in Figures 6.

Figure 6. Test set detection results

4.2.2. Experimental Results and Analysis

The experimental results obtained from the irregular precision parts detection model are now analyzed, and the analysis results are as follows.

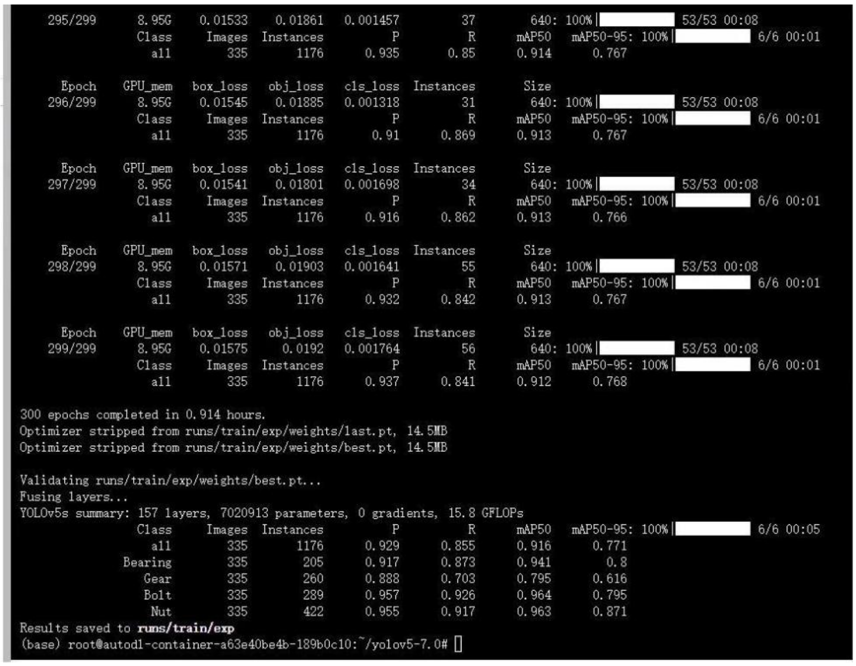
After renting a server on the computing cloud AutoDL website, the labeled image dataset and the training set archive are compressed and uploaded to the server for program execution. The program runtime results are shown in Figure 7. As seen from the figure, the precision (P) is 0.929, the recall (R) is 0.855, mAP50 is 0.916, and mAP50-95 is 0.771.

The training results are generally observed by the size of the mAP curve area and the fluctuations of precision and recall (the four images on the far right). The closer the area of the PR curve is to 1, the better. If the fluctuations in precision and recall are small, the training effect is better. As shown in Figure 8, the model has been trained well, and the curves on the graph steadily rise.

As shown in Figure 9, the curve in the figure is the Precision-Recall (PR) curve, which represents the relationship between the precision and recall of the classifier at different thresholds. Recall is used as the horizontal axis, ranging from 0.0 to 1.0, and Precision is used as the vertical axis, ranging from 0.0 to 1.0. From Figure 9, we can see that there are five different colored curves. The green curve represents the precision-recall curve for the Bolt category, with the highest precision of 0.964. The curve is relatively smooth and stable, indicating that the classifier

ISSN: 2688-8645

performs well in recognizing the Bolt category, with high accuracy and stability. The orange curve represents the precision-recall curve for the Gear category, with the lowest precision of 0.795. The curve starts to drop from the middle and is not stable enough, indicating that the classifier is not accurate enough in recognizing the Gear category and needs further model optimization. The dark blue curve represents the average precision-recall curve for all categories, with a precision of 0.916.



```
  295/299      8.95G    0.01533    0.01861   0.001457         37        640: 100%|██████████| 53/53 00:08
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 6/6 00:01
                   all        335       1176      0.935       0.85      0.914      0.767

     Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
  296/299      8.95G    0.01545    0.01885   0.001318         31        640: 100%|██████████| 53/53 00:08
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 6/6 00:01
                   all        335       1176       0.91      0.869      0.913      0.767

     Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
  297/299      8.95G    0.01541    0.01801   0.001698         34        640: 100%|██████████| 53/53 00:08
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 6/6 00:01
                   all        335       1176      0.916      0.862      0.913      0.766

     Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
  298/299      8.95G    0.01571    0.01903   0.001641         55        640: 100%|██████████| 53/53 00:08
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 6/6 00:01
                   all        335       1176      0.932      0.842      0.913      0.767

     Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
  299/299      8.95G    0.01575     0.0192   0.001764         56        640: 100%|██████████| 53/53 00:08
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 6/6 00:01
                   all        335       1176      0.937      0.841      0.912      0.768

300 epochs completed in 0.914 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 14.5MB
Optimizer stripped from runs/train/exp/weights/best.pt, 14.5MB

Validating runs/train/exp/weights/best.pt...
Fusing layers...
YOLOv5s summary: 157 layers, 7020913 parameters, 0 gradients, 15.8 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 6/6 00:05
                   all        335       1176      0.929      0.855      0.916      0.771
               Bearing        335        205      0.917      0.873      0.941        0.8
                  Gear        335        260      0.888      0.703      0.795      0.616
                  Bolt        335        289      0.957      0.926      0.964      0.795
                   Nut        335        422      0.955      0.917      0.963      0.871
Results saved to runs/train/exp
(base) root@autodl-container-a63e40be4b-189b0c10:~/yolov5-7.0#
```

Figure 7. Program runtime results

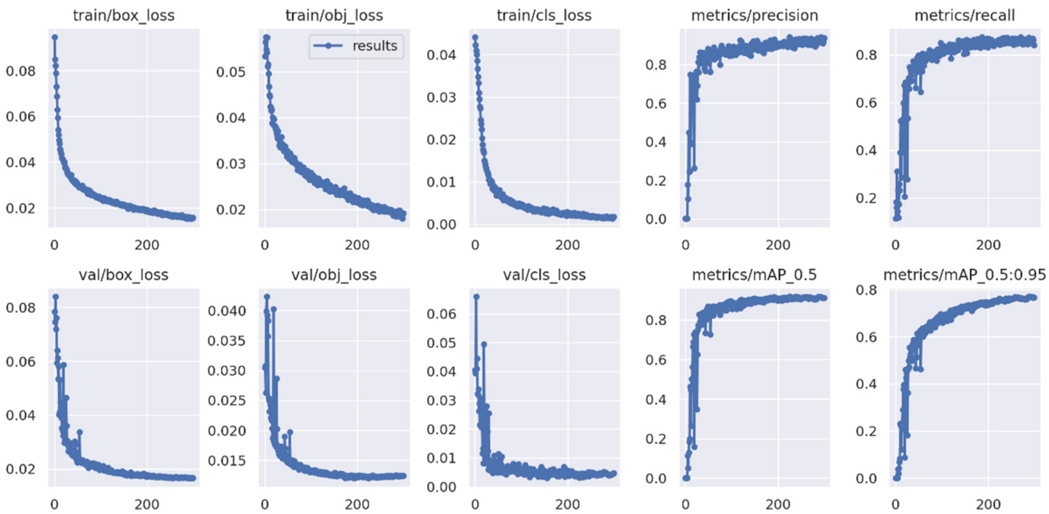


Figure 8. Training results

5. Conclusion

With the rapid development of industrial automation, the detection of industrial parts has become a key factor in improving production efficiency and product quality. Traditional detection methods heavily rely on manual operations, leading to low detection efficiency and being highly susceptible to human factors, which makes it difficult to ensure the accuracy of the detection results. To address these issues, this paper proposes an optimized YOLOv5-based

ISSN: 2688-8645

component detection model algorithm for detecting irregular precision parts, and experiments were conducted using a self-created dataset of irregular precision parts. Through experimental validation, the YOLO-based component detection algorithm demonstrated significant improvements in both accuracy and speed. Compared to traditional detection methods, the YOLO-based algorithm is capable of detecting the position and category information of components more quickly and accurately, thus validating the superiority of the YOLO-based algorithm in the detection of irregular precision parts.

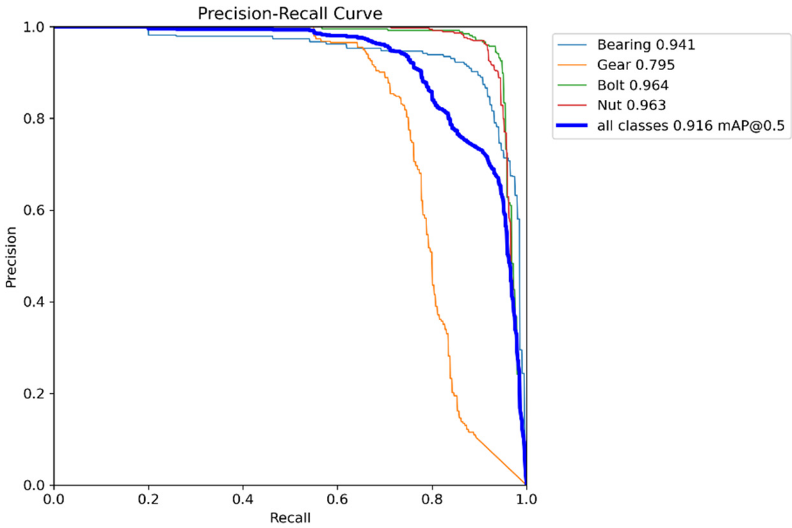


Figure 9. Precision-Recall Curve

Acknowledgments

Funding: This research was funded by the Tianjin Research Innovation Project for Postgraduate Students [Project No. 2022SKYZ053].

References

[1] J.C. Huang, J. Zhou, L. Ding: A fast detection method for part targets based on the improved YOLO v3 algorithm (Journal of Nanjing Engineering University), 2020, 18(03): 6-11.

[2] L. Xu: Research On Surface Defect Detection Algorithm Of Industrial Parts Based On Deep Learning (M.Sc., East China Jiaotong University, China 2022), p.2.

[3] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez: On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data (Remote Sensing), 2020, 13(1): 89.

[4] Y. Liu, J. Han, Q. Zhang: Salient object detection via two-stage graphs (IEEE Transactions on Circuits and Systems for Video Technology), 2018, 29(4): 1023-1037.

[5] A.A. Mustapha, M.S. Yoosuf: Exploring the efficacy and comparative analysis of one-stage object detectors for computer vision: a review (Multimedia Tools and Applications), 2024, 83(20): 59143-59168.

[6] X.G. Li, X. He, Y.W. Zhang: Improved Lightweight Infrared Target Detection Algorithm Based on Deep Learning (Journal of Optoelectronic Technology and Applications), 2024, 39(04): 49-54+89.

[7] Y.H. Shao, Z. Zhang, H.Y. Chu: A Review of YOLO-based Object Detection Using Deep Learning (Journal of Electronics and Information Technology), 2022,44(10): 3697-3708.

[8] L.C. Jiao, F. Zhang, S.C.H. Hoi: Recent advances in deep learning for object detection (Neurocomputing), 2020, 396: 39–64.

ISSN: 2688-8645

[9] X.W. Wu, D. Sahoo, F. Liu: A survey of deep learning-based object detection (IEEE Access), 2019, 7: 128837-128868.

[10] L.L. Pu, L. Yang: Improved YOLOv5 algorithm for real-time multi-vehicle detection and tracking (Science Technology and Engineering), 2023, 23(28): 12159-12167.

[11] R.M. Li, Q.K. Xiao, Z.J. Han: Face mask detection based on the improved YOLOv5 algorithm (Computer Engineering and Design), 2023, 44(09): 2811-2821.

[12] Y.Q. Wang, M.Z. Meng: Establishment and preliminary application of the YOLOv5 deep learning prediction model for accurate identification of emergency medicine packaging boxes (Pharmaceutical Guide), 2023: 1-11.

[13] Y.M. Zhou, P. Li, Q.X. Lin: Detection of missing bolts on subway vehicle bogies based on an improved YOLOv5 algorithm (Mechanical Engineer), 2023(09): 50-54.