J. Electrical Systems 20-3 (2024): 852-869

[1,*]Song Chen
[2]Dongting Guo
[3]Dagui Wang
[4]Fangbin Wang
[5]Han Tang

Image Recognition System for Mechanical Parts Based on SSD and YOLOv5

Abstract: - With the development of industrial automation, the importance of mechanical parts in the production process is becoming increasingly important. In recent years, deep learning technology has made breakthrough progress in fields such as image recognition and object detection. Deep learning models can learn a large amount of data through multiple epochs, automatically extract features from it, and achieve image recognition and object detection. This provides potent technical assistance for the design of mechanical component recognition systems. In industrial production, there are a wide variety of mechanical parts with complex shapes, and the differences between different models are often very small, making it difficult to identify them manually. Therefore, it is necessary to develop a system that can automatically identify mechanical parts to improve production efficiency and quality. This article takes mechanical parts as the research object and studies the recognition technology of mechanical parts based on deep learning and image recognition. The main content of the article includes the following aspects. Research was conducted on image quality enhancement methods for images with different characteristics, and different image preprocessing methods were applied to different images. YOLOv5 and SSD (Single Shot MultiBox Detector) algorithms are used to perform deep learning and recognition detection on the processed images, respectively. By comprehensively comparing the differences between the two algorithms, YOLOv5 was selected as the focus of this design, and based on this, deep training was conducted, and the code was imported into the embedded system.

Keywords: Yolov5, Mechanical Parts, Deep Learning, Image Recognition, Single Shot Multibox Detector.

I. Introduction

In the field of machinery, whether it is the assembly of robot parts or the classification of parts in the waste recycling station, there are many kinds of Machine element, a large number of them, a single traditional classification environment and low classification accuracy.

After the realization of automatic production of products, the machinery manufacturing industry is gradually looking for a new technology to achieve automatic detection of Machine element, thereby improving production efficiency. It is currently the key direction to achieve the type identification of Machine element and size measurement technology research.

The traditional detection of Machine element mainly depends on manual work, and the staff need to work in a high degree of tension and continuity, which easily leads to eye fatigue, resulting in detection errors. In order to avoid testing becoming a bottleneck in production, it is necessary to accelerate the testing speed and ensure that the testing work is completed in a shorter time. However, this will also increase labor and management costs, affecting the production efficiency and profits of the enterprise. Therefore, the machinery manufacturing industry is looking for a new technology to realize automatic detection and assembly of Machine element, thus improving production efficiency and product quality. The application of machine vision technology can provide the manufacturing industry with efficient, accurate and reliable Machine element detection and assembly solutions, greatly reduce labor costs and management costs, improve production efficiency and quality, and promote the intelligent and sustainable development of the manufacturing industry. As the foundation of mechanical processing automation, part recognition liberates workers from heavy labor, improves work efficiency, and also reduces costs.

Automatic detection of Machine element is an important part of the production system in the manufacturing industry. Modern manufacturing technology widely uses automatic detection and product identification to monitor and ensure product quality in various batch production and multi variety production, making the processing system more reliable. Petr Burian introduces a new method for organizing neural network data in n-tuple memory based on the implementation of this concept using FPGA (Field Programmable Gate Array)

[1] College of Mechanical and Electrical Engineering, Anhui Jianzhu University, Hefei, China
[2] College of Mechanical and Electrical Engineering, Anhui Jianzhu University, Hefei, China
[3] College of Mechanical and Electrical Engineering, Anhui Jianzhu University, Hefei, China
[4] College of Mechanical and Electrical Engineering, Anhui Jianzhu University, Hefei, China
[5] College of Mechanical and Electrical Engineering, Anhui Jianzhu University, Hefei, China
*Corresponding author: Song Chen

devices [1]. Wei Liu et al. proposed a method to detect objects in images using a single depth neural network SSD and scaled the location of the element map [2]. Rahim Panahi and Iman Gholampour proposed an online high-precision system for Automatic License Plate Recognition (ANPR). The system utilizes machine vision and evaluates, compares, and improves various related algorithms to achieve high accuracy rates [3]. Linyu Wang et al. proposed a new method for target recognition using YOLO network model and bounding the target area. Some scholars have improved the YOLO algorithm [4], such as YOLO-L [5], Fast-YOLO [6] and DSM-IDM-YOLO [7]. Canny detection algorithm is used to extract edge feature points, FAST Corner detection algorithm is used, and BRIEF description algorithm is used to construct feature point descriptor. By combining two detection algorithms, the problem of less texture detection and fewer feature matching points can be solved [8]. Yina Wang and Guoqiang Fu have developed a novel Object detection algorithm to solve the problem that the robot system needs to quickly recognize targets in a long distance. It uses a sparse detection algorithm to modify YOLOv5. The detection results demonstrate that this method can classify and locate most small-scale objects faster and more accurately [9]. Zhiping Ying et al. proposed an automated method based on YOLOv5 to address the issues of ultra small size, similar structure, and high reflectivity of steel wire braided hoses, replacing traditional naked eye detection [10]. Yang B improved algorithm to detect obscured targets and small targets [11]. Wang Hai compared five mainstream deep learning object detection algorithms [12]. Ruifang Ye compared the performance of several deep learning models in processing multi defect images. The YOLOv3 model is best used to overcome the diversity and complexity of printed circuit board components through performance comparison testing. The experimental results show that the error rates of components defect detection and classification are as low as 0% and 2.42%, respectively, indicating that the optimized YOLOv3 model can be applied to industrial production lines to achieve the goal of high-precision detection and classification of composite defects [13]. Xupeng Kou selects the ideal feature scale for model training [14]. Wang Peng uses transfer learning methods to train SSD networks [15]. Peng Hanyu designed the binary detection neural network [16]. Vinh Quang Dinh uses two cameras to detect images [17]. Joshi Keyur applied the support vector machine algorithm to the recognition of mechanical parts [18]. Claudio Cusano proposed a visual recognition module for aircraft mechanical components, which was evaluated on real aircraft and considered the need to identify 20 different types of maintenance operations for different mechanical components. The visual recognition module has been tested under different imaging conditions and the proportion and direction of the parts of interest have been changed. The results confirmed its feasibility under such challenging and realistic conditions [19].

It should be noted that the accuracy, efficiency and practicability of the model should be taken into account in the design of the image recognition system for Machine element. It can be adjusted and optimized according to actual application scenarios and requirements, thereby improving the practicality of the system.

Many scholars have not paid attention to the comparison of various image preprocessing results, and have not combined image preprocessing with intelligent algorithms. In addition, the use of computers to process images of mechanical parts is not suitable for industrial use.

Based on machine vision and deep learning technology, an image recognition system for machine element is designed in the paper. The main research contents are as follows:

To achieve a highly practical object, which has the stroboscopic photography function, can automatically identify parts through image input, and can accurately identify the categories and names of different machine element.

Data acquisition: the system needs to collect a large number of image data of machine element for training and testing models. The users can use a camera or mobile phone to capture images and collect your own dataset.

Data processing: The collected images need to be preprocessed, including image histogram equalization, image enhancement, edge enhancement, normalization and other operations, for use in the model.

Model selection: Image classification algorithms are used to classify and recognize machine element. YOLOv5 and SSD algorithm are selected in this paper. We select the final algorithm to be adopted by comprehensively comparing the final recognition speed and confidence level.

II. Hardware Design

A. Hardware Introduction

1) Controller

The Jetson Nano B01 processor is used as the controller, as shown in Figure 1. The specific configuration is shown in Table 1. The power supply is connected to DCIN (type-C). The camera, keyboard and mouse are connected to USB port. The screen is connected via an HDMI port. The interface is shown in Figure 2.

Figure 1: Controller

Figure 2: Seed Studio Interface

Table 1: Jetson Nano B01 Processor Configuration

Project	configuration
Processor	Jetson Nano B01
GPU	GPU NVIDIA Maxwell architecture, 128 NVIDIA CUDA core
CPU	quad core ARM Cortex-A57
Processor memory	4GB 64 bit LPDDR4
Storage	16G EMMC
High-speed interface	1 x PCIe, 4x USB 3.0I/O

2) Camera hardware

The camera used in this design is Sony IMX179, as shown in Figure 3. The Lens Size is 1/3.2inch and the maximum effective pixel is 3264 × 2448. We take photos with the camera and read them for recognition. Alternatively, program flash photography can be used to reading data automatically for recognition.

B. Operating

The image recognition hardware system is built as shown in Figure 4. We place the workpiece to be identified below the camera's field of view, and the recognition results are shown in Figure 5.

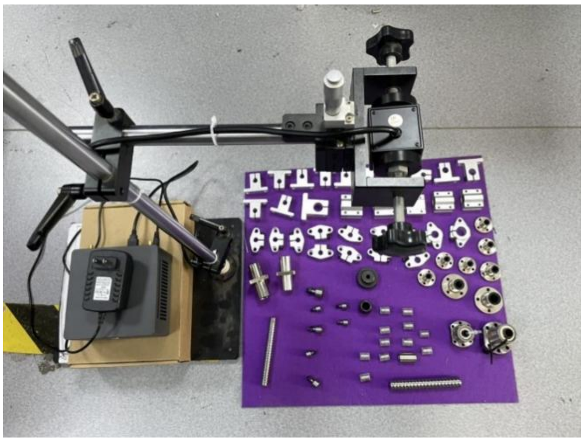


Figure 4: Image Recognition Hardware System

Figure 5: Identification Results

III. SOFTWARE DESIGN

A. Overview

LabelImg is used to annotate the information of the parts on the images to obtain the required dataset for YOLO and SSD. PyCharm is used in Python environment as the carrier to run the code training dataset.

B. Algorithm Principle

1) YOLOv5

YOLO (You Only Look Once) series algorithms are a typical one stage object detection algorithm that combines classification and target localization regression problems using anchor box. The YOLO algorithm uses a separate CNN (Convolutional Neural Network) model to achieve end-to-end object detection. The core idea is to use the entire graph as the input of the network and directly regress the position of the bounding box and its category in the output layer. Its detection system is shown in Figure 6. The confidence formula is shown in Formula 1.

$$Confidence = \Pr(Object) * IoU_{truth}^{pred} \tag{1}$$

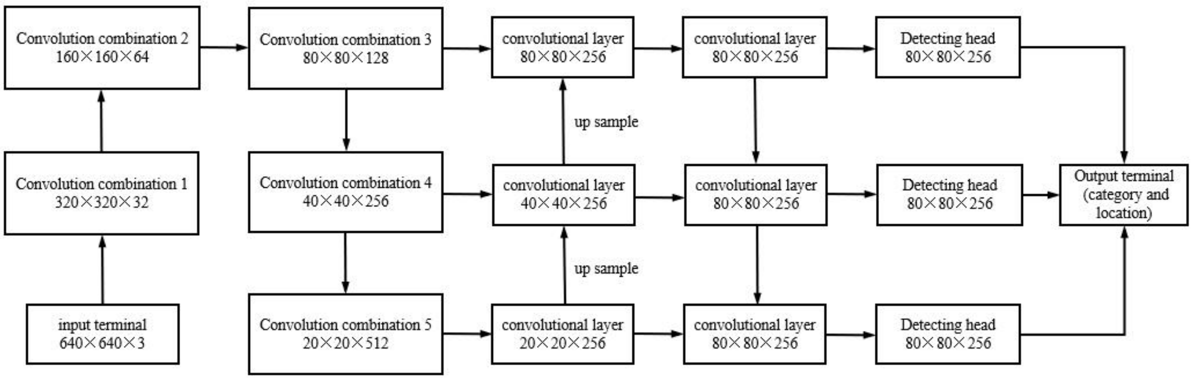


Figure 6: YOLOv5 Detection System

YOLOv5 is a target detection algorithm, and its model structure mainly includes the input end, Backbone network, Neck network, output end, and Activation function. In general, YOLOv5's model structure is relatively simple, but it uses a variety of technologies and strategies, such as CSP structure, FPN network, Mish Activation function and FocalLoss Loss function, so as to improve the performance and robustness of the model.

Each version of YOLOv5 has four open source models, including YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. Other models deepen and widen the network based on this. Each model has different characteristics, and the algorithm performance of commonly used network models is different. In order to achieve faster recognition speed and higher accuracy, YOLOv5l6 is used for training in this design.

2) SSD

SSD (Single Shot MultiBox Detector) is a target detection algorithm based on deep learning. It adopts the design of a single-stage detector, and uses the tag information of input data to guide the training and learning of the model. It is mainly composed of convolution layer, Activation function layer, and pooling layer. It combines target recognition tasks with border regression tasks through multi task learning strategies, and unified into an end-to-end depth Convolutional Neural Network, which makes the optimization and training of the entire model very easy. The algorithm structure diagram is shown in Figure 7.

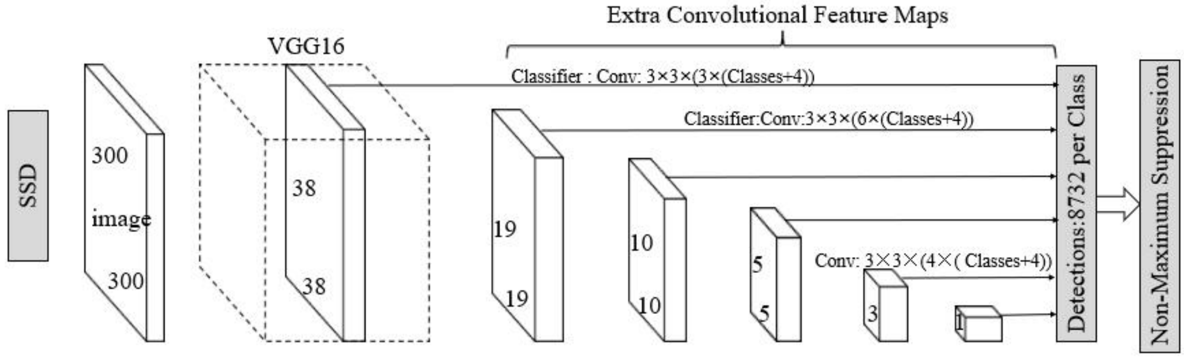


Figure 7: SSD Network Structure Diagram

SSD Convolutional neural network model consists of two parts, feature extraction network and detection and recognition network. The feature extraction network is composed of a basic network and auxiliary structures, mainly responsible for extracting convolutional feature maps of different scales from input images for subsequent multi-scale prediction. For each scale of convolutional feature map obtained by the feature extraction network, the detection and recognition network use an independent convolutional predictor to process, predict the category of the target in the input image and the position of the target in the image, and provide the confidence score of the target belonging to a certain class and the boundary box position information of the target. Then, by synthesizing the predicted information of target categories and positions obtained on multi-scale feature maps, a Non Maximum Suppression (NMS) algorithm is executed to remove redundant target prediction bounding boxes and obtain the final detection and recognition results. The SSD Loss function formula is shown in Formula 2.

$$L(r,c,l,g) = \frac{1}{N}(L_{conf}(r,c) + \alpha L_{loc}(r,l,g)) \tag{2}$$

SSD algorithm realizes target detection by adding multiple prediction layers in Convolutional neural network, and can simultaneously detect targets with different Aspect ratio. At the same time, it also adopts technologies such as multi-scale feature fusion and non-maximum suppression, thereby improving the accuracy and efficiency of detection.

C. Encoding

Each part is assigned a unique code, which refers to the part. The part names and codes are shown in Table 2.

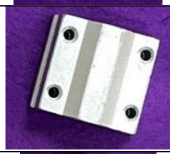
D. Image Preprocessing

Grayscale changes, geometric correction, image enhancement, and image filtering, are commonly used for image preprocessing in machine vision training. Image preprocessing includes single image input and multiple image inputs. Due to the large sample size of this design, multiple image inputs were used, resulting in the use of cv2's imread() and imwrite().

Due to the high clarity of the input image itself, there is no need for excessive image processing to avoid obtaining a blurrier image.

As shown in Figure 9, the images are divided into the original image, enhancement image, spatial filtering, and edge segmented image. Due to the inappropriate parameters used for gamma correction, contrast and brightness enhancement, and median filtering denoising in image enhancement, the enhancement image is not as clear as the original image, resulting in excessive enhancement. The same applies to images after spatial filtering, smoothing, and edge segmentation. Therefore, image preprocessing should reasonably select the necessary processing based on the characteristics of the image. The design finally adopts Histogram equalization, graying and image normalization. As can be seen from Figure 8, the image preprocessing effect is very significant.

Table 2: Part Name and Code

Code	Name	Image
SK	Vertical Support	
SHF	Horizontal Support	
SCS	Box Aluminum Slide	
LMF	Circular Flange Linear Bearing	
KC	Extension Middle Method Lan	
LM-1	Sprocket	
LM-2	Motor boss gear	
CF	Bolt needle roller bearing	
LM	Linear bearing	
1204-1	Ball screw nut	
1204-2	Precision ball screw	

Histogram equalization technology transforms the grayscale histogram of the original image from a relatively concentrated grayscale interval to a uniform distribution within the entire grayscale range. Due to its simple algorithm and no need for parameter settings from external factors, it can operate independently and effectively enhance image contrast. It is a commonly used image enhancement method.

Image normalization refers to the process of performing a series of standard processing transformations on an image to transform it into a fixed standard form. This standard image is called a normalized image.

The original image can obtain multiple duplicate images after undergoing some processing or attack, and these images can obtain the same form of standard images after being normalized with the same parameters.

The Histogram equalization, graying and image normalization formulas are as follows:

$$s_k = \sum_{j=0}^{k} \frac{n_j}{n}, k = 0,1,2,,,L-1 \tag{3}$$

$$Gray = (R + G + B)/3 \tag{4}$$

$$X_{normslization} = \frac{x - Min}{Max - Min} \tag{5}$$

$$X_{normalization} = \frac{x - \mu}{\sigma} \tag{6}$$

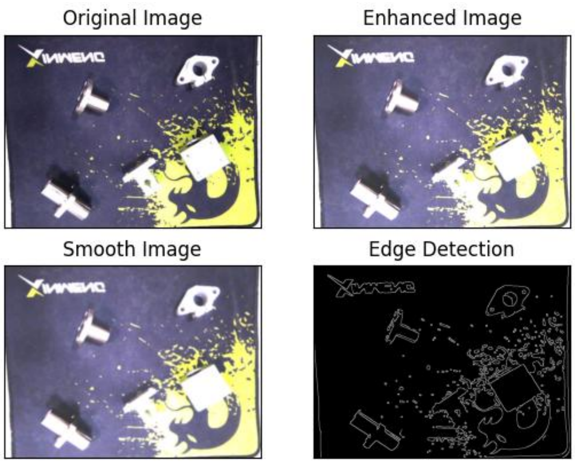


Figure 8: Original Image, Enhanced Image, Smooth Image and Image after Edge Detection

E. YOLOv5 Training

Although the YOLO series has been updated to YOLOv8, considering the required functionality, the latest version is not required. Therefore, YOLOv5 is used as the algorithm for model training in this design.

When building a machine learning model, we need to divide the preprocessed dataset into three folders: train, validation, and test. The training set is used for model training, the validation set is used to adjust the hyperparameters of the model and prevent overfitting, and the testing set is used to evaluate the performance and generalization ability of the model.

1) Dataset Generation

Labelimg software is used for dataset generation. Set the mode of Labelimg to YOLO, to label the images obtained from image acquisition. It can generate a txt format labels file, and automatically save the labeled data to the specified folder.

2) Training

a) Data preparation

Create a folder called datasets in the YOLOv5 directory to store our data. Create images and labels in the datasets folder to hold the required images and annotation files for training. Create a yaml file, and write the names and codes of the parts that need training. If the datasets are not in the YOLOv5 directory or if the file name in the datasets is not the default images and labels, train.py can also find the dataset by modifying the corresponding contents of the yaml file and generous.py file.

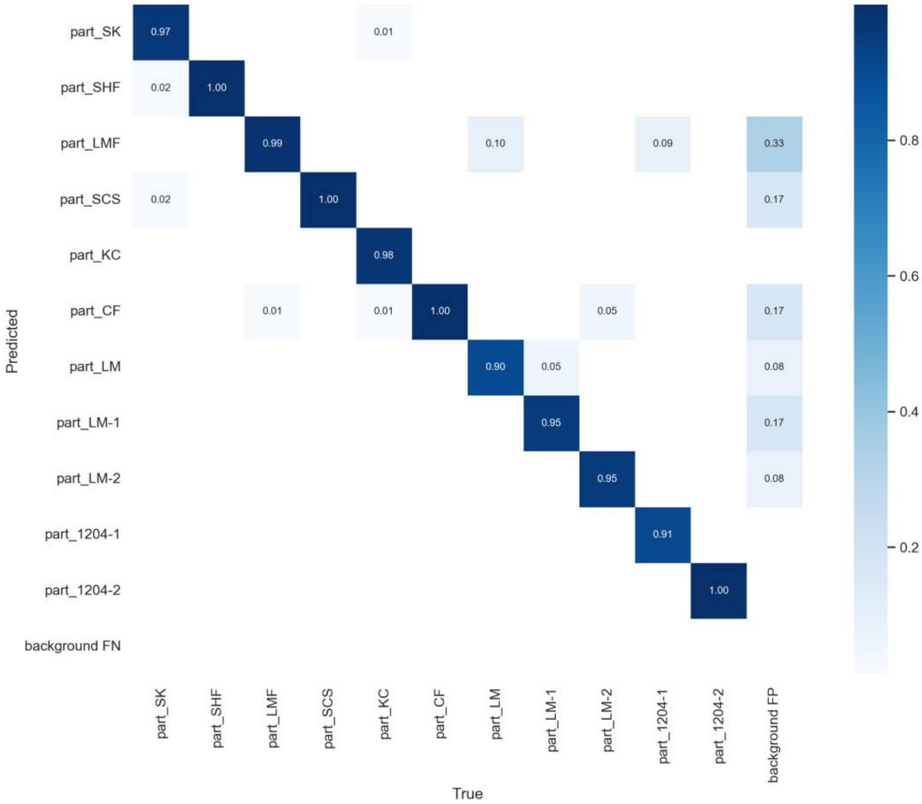
The dataset of this design is named VOCdevkit, and the path is at the same level as the YOLOv5 directory. Therefore, it is necessary to modify the code to allow the program to find the dataset. It is worth noting that if both YOLOv5 and other YOLO models exist in a processor, it is necessary to change all default relative paths to absolute paths in the training code to avoid the program from not finding the corresponding files.

b) Network training

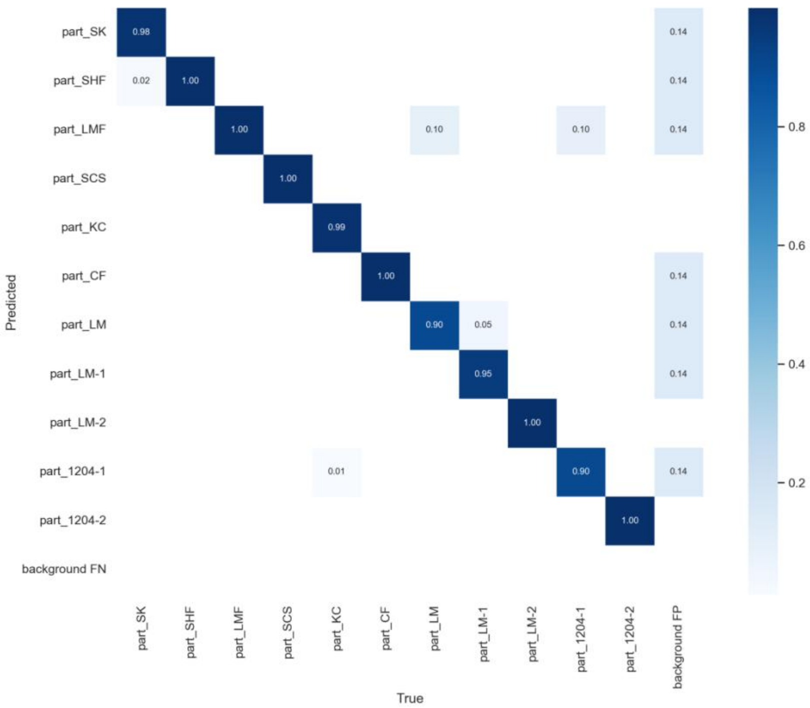
Run the training program in the terminal and wait for the training results. The training results are saved in the root directory of training program.

Batch size is the number of samples passed to the program for training in a single pass, and the larger the current processor allows, the better. Epochs is the number of training rounds, and the larger the epochs, the higher the recognition confidence and accuracy after training. If the number of datasets is not large enough, the results can also be converged and accuracy improved by increasing the number of training rounds.

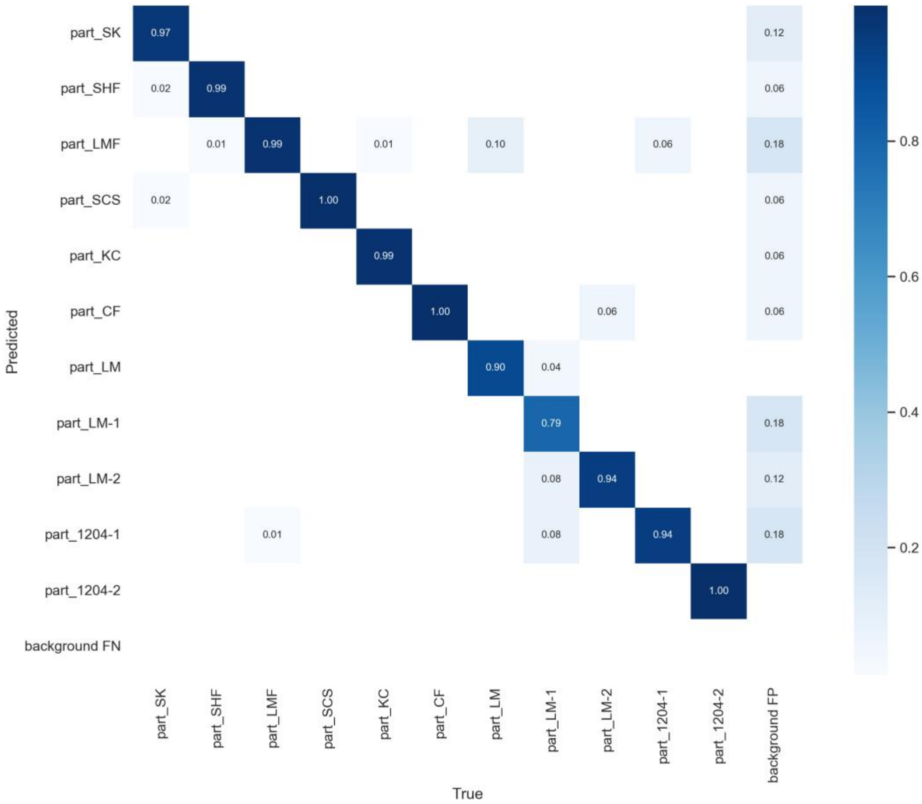
Due to the large dataset, this design trained 200 epochs each time, a total of four times, including the original image and three different preprocessed images. These three kinds of image pre-processing are (1) image graying, denoising, gamma correction and normalization, (2) Histogram equalization, image graying, denoising, gamma correction and normalization, (3) Histogram equalization, image graying and normalization. The Confusion matrix and result graph of each training result are shown in Figure 9 and Figure 10 respectively.



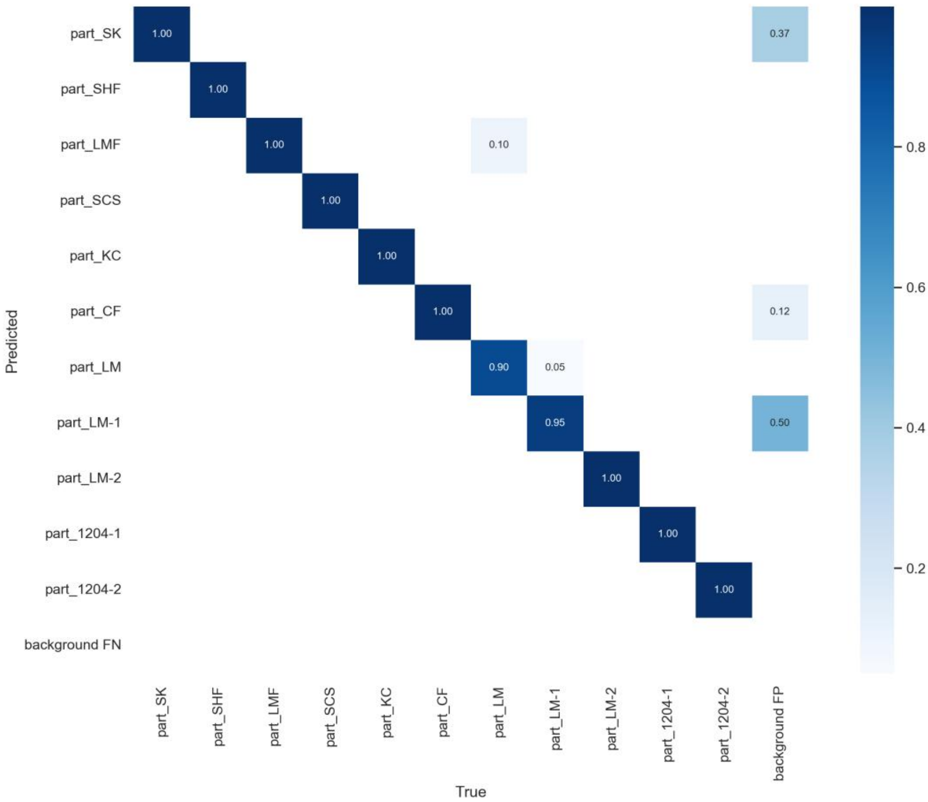
(a) Original image



(b) Image grayscale, denoising, gamma correction, and normalization

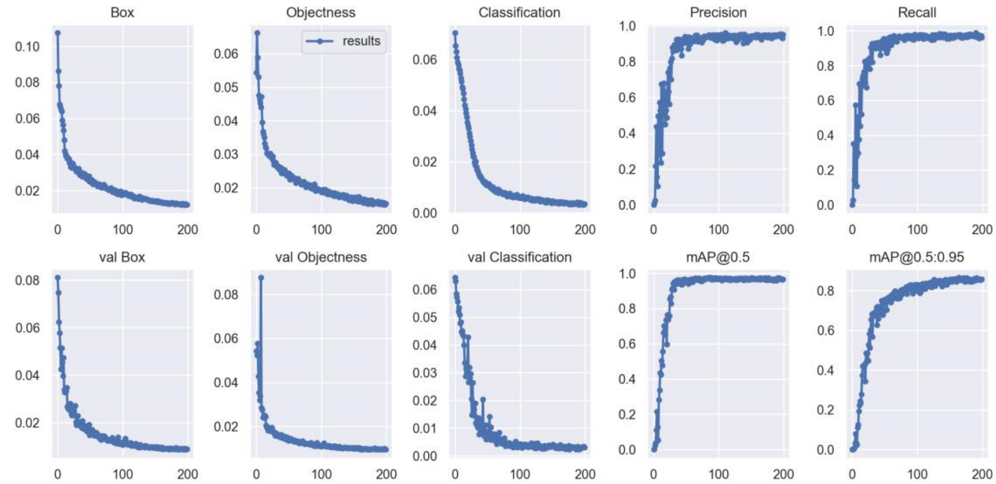


(c) Histogram equalization, image graying, denoising, gamma correction and normalization

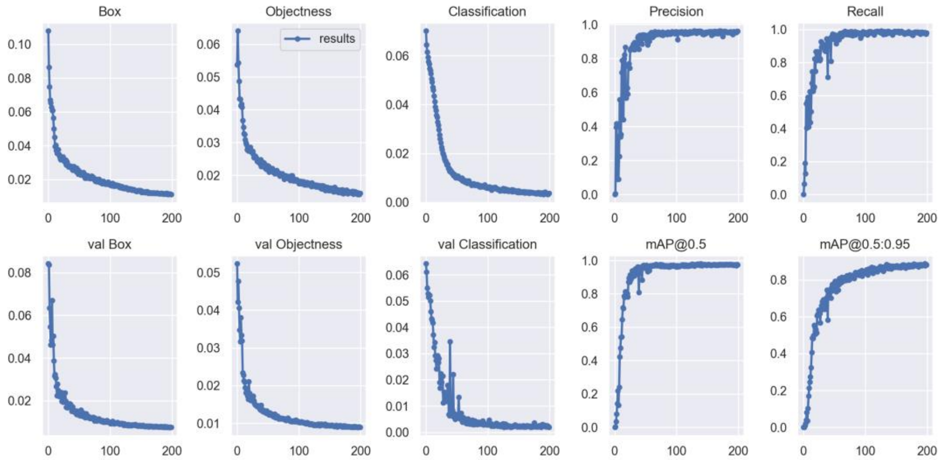


(d) Histogram equalization, image graying, normalization

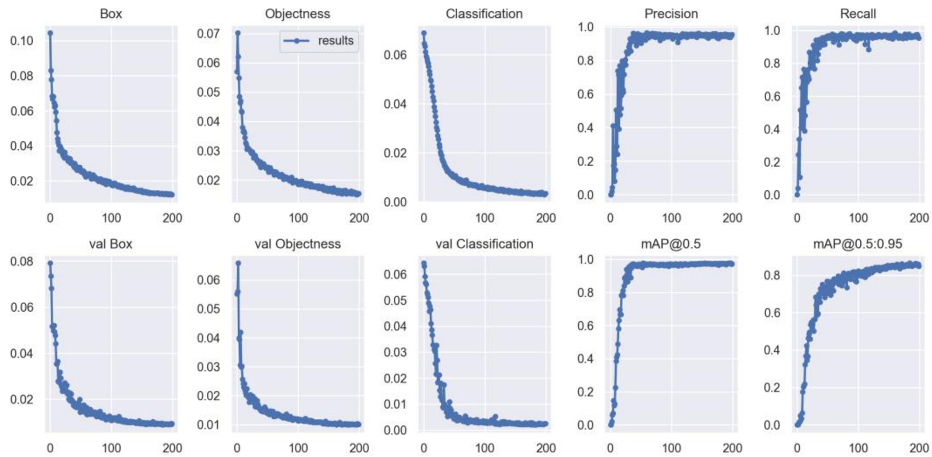
Figure 9: Confusion Matrix



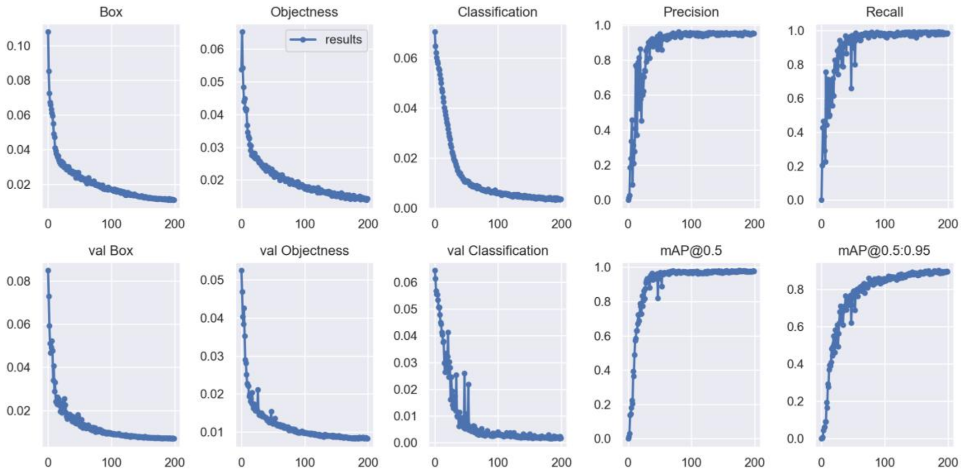
(a) Original drawing



(b) Image graying, denoising, gamma correction, and normalization



(c) Histogram equalization, image graying, denoising, gamma correction and normalization



(d) Histogram equalization, image graying, normalization

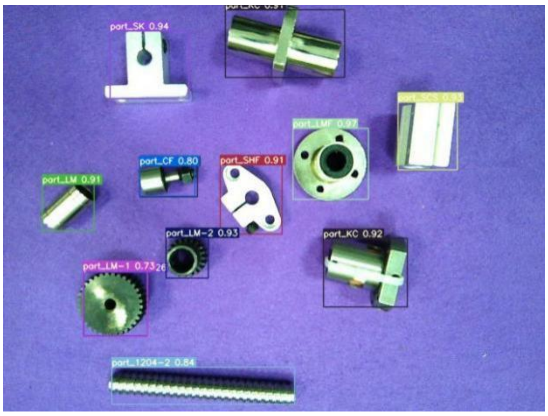
Figure 10: Parameters of Image Preprocessing Results

3) Testing

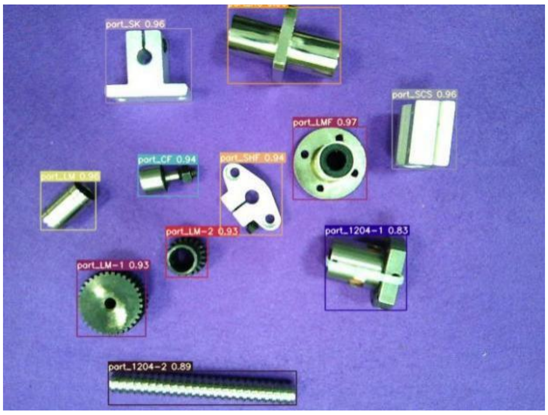
Run the test program in the terminal. Best.pt is the weight file obtained through training. Different weight files are selected to detect the same test set, and the results are compared to select the optimal weight file for comparison with SSD. Figure 11 shows the test results of VOCdevkit test images using different training weight files.

(a) Test results of original image training

(b) image graying, denoising, gamma correction, and normalization training



(c) Test results of Histogram equalization, image graying, denoising, gamma correction and normalization training



(d) Test results of Histogram equalization, image graying, normalization training

Figure 11: Test Results

From Figure 11-a, it can be seen that the confidence level of the original image recognition after training is relatively high, but some parts are recognized incorrectly. As shown in Figure a, part 1204-1 is recognized as part KC. Comparing Figure 11-a and Figure 11-b, it can be seen that the use of image grayscale, denoising, gamma correction, and normalization in image preprocessing can improve the confidence level of some part recognition, but it does not solve the problem of recognition errors, and the recognition of part CF errors is replaced by part KC. Comparing Fig. 11-a and Fig. 11-c, it can be seen that the use of Histogram equalization, image graying, denoising, gamma correction and normalization in image preprocessing has improved the confidence of some parts recognition, but also has not solved the problem of part-1204-2 recognition errors. Comparing Figure 11-a and Figure 11-d, we can see that word Histogram equalization, image graying and normalization in image preprocessing not only improve the confidence of recognition, but also reduce the errors of recognition.

Compared with the third preprocessing method, the first two preprocessing methods both increase workload and do not improve recognition accuracy, so these two image preprocessing methods are not used. From this, it can be concluded that not all image enhancement can amplify the features of the image, thereby achieving the effect of data enhancement. The wrong choice may cause image blurring, leading to a decrease in the recognition accuracy of the training results. Therefore, the selection of image preprocessing is very important during training.

F. SSD Training

1) Dataset generation

Dataset generation can be generated using Labelimg like YOLO datasets, or you can directly run the program to convert YOLO labels into VOC corresponding XML. Due to the large number of images, the latter is chosen for this design.

Create a new SSDsets folder, where the pictures, txt, and XML folders are created to hold the images, labels, and converted XML files required for training. Set the path to obtain the XML file required for SSD training.

2) Network training

a) Preparation of Datasets: Create a new VOCdevkit folder and create a folder named VOC2 under that folder. Create Annotations, JPEGImages, and ImageSets in VOC2. Annotations are used to store files for training, while JPEGImages are used to store image files for training.

b) After completing the placement of the dataset, we need to use voc_annotation.py to process the dataset. Divide the dataset to obtain the Train.txt and Val.txt for training purposes. Before running the program, we need to establish a cls_Classes.txt, which includes the categories you need to distinguish. Modify the classes_path in voc_annotation.py so that it corresponds to cls_classes.txt, and run voc_annotation.py to get the txt file required for training.

c) Start network training by adjusting the epochs and batch size of the freezing and thawing stages according to the processor configuration and dataset size, in order to obtain the optimal results under the current configuration. After adjusting various parameters, run train.py for training.

In YOLOv5 training, Histogram equalization, image graying and normalization training are the best methods for preprocessing, so this image preprocessing is directly used in SSD training.

3) Detection and evaluation

Two files ssd.py and predict.py are required for the prediction of training results. Modify model_path and classes_path in ssd.py. After completing the modifications, run predict.py, and then type the image path to detect it.

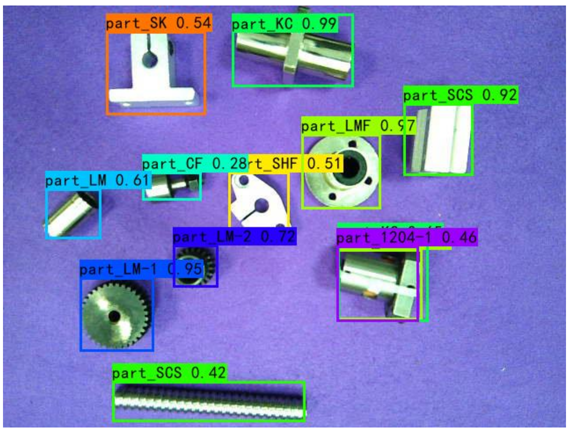


Figure 12: SSD Test Results

It can be seen from Figure 12 that the confidence of SSD training results in the recognition of Machine element in this picture is seriously polarized, and there are several recognition results at the same time for individual parts with high similarity. For example, the ball screw nut in the figure is coded 1204-1, but it has been identified as three types of parts, namely part-1204-1, part-KC, and part-CF. The confidence level for identifying it as part-KC is 0.65, which is greater than the confidence level of the real code of 0.46.

G. Comparison

There are many factors to consider when selecting an algorithm, such as training speed, recognition speed, training difficulty, accuracy, and the bearing capacity of the hard drive. The selection should be based on actual use. If there is a high demand for accuracy and a low demand for speed, speed can be appropriately discarded to improve accuracy; In situations where both requirements are high, it is necessary to repeatedly compare various parameters. Next, this article compares YOLOv5 and SSD from three aspects: speed, accuracy, and accuracy of marker box position, and draws conclusions.

1) Speed

a) Training speed

For the same device, with the same batch size and no CUDA acceleration, the average one epoch of YOLOv5 training is 10 minutes; The average freezing phase of SSD is 5 minutes per epoch, and the average thawing phase is 20 minutes. With 200 rounds of training and a thawing epoch of 50, the average epoch of SSD is 16.25 minutes. It can be seen that YOLOv5 has a faster training speed.

b) Recognition speed

For the same image, the recognition time for YOLOv5 is 12-17ms, while for SSD, it takes 2-3 seconds. Therefore, the recognition speed of YOLOv5 is also faster.

2) Accuracy

Figure 13 shows the relationship between the recognition accuracy and confidence of YOLOv5. From the figure, it can be seen that the recognition confidence of all types of parts is 0.954, and the accuracy is 1. According to Figure 14, the average accuracy of SSD recognition is 95.32%. From a data analysis perspective, the accuracy of the two is very similar.

Figure 15 shows the actual recognition accuracy of YOLOv5 and SSD for a certain image. As shown in Figure a, the accuracy of YOLOv5 recognition is very high, and except for part-1204-2, the confidence level is above 0.92, reaching a maximum of 0.97. As shown in Figure b, it can be seen that although the confidence level of SSD in identifying some parts is higher than 0.95, the overall confidence level is not high, and for some parts with high similarity, there may be recognition errors and multiple marking boxes for one part. From the actual recognition results, YOLOv5 has higher recognition accuracy.

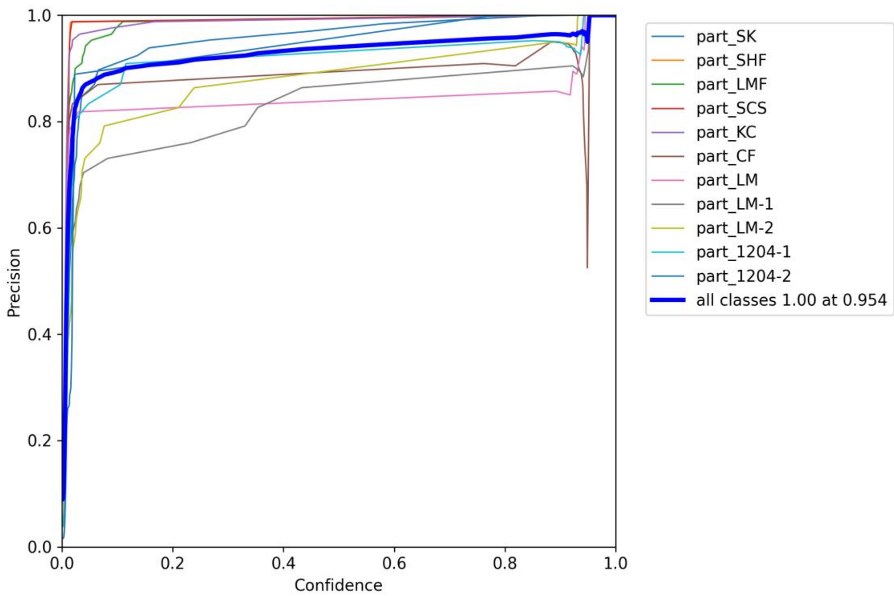


Figure 13: Relationship between Accuracy and Confidence of YOLOv5

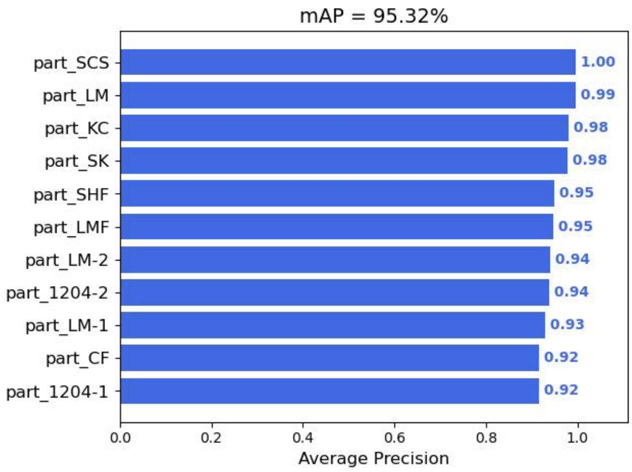
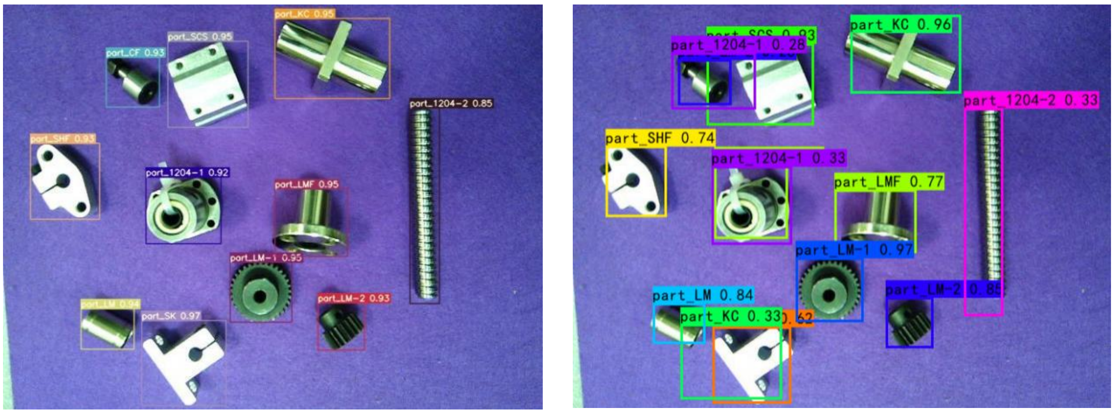


Figure 14: Average Accuracy of SSD Recognition



(a) YOLOv5 (b) SSD

Figure 15: Actual Detection Results of Two Algorithms

3) Marking box

As shown in Figure 15, SSD marking boxes often appear too large, too small, and offset. In contrast, the marking box of YOLOv5 reflects the true position of the parts better.

4) Conclusion

To sum up, YOLOv5 is superior to SSD in terms of training speed, recognition speed, recognition accuracy and marker box position, so YOLOv5 is finally used as the training and recognition algorithm in this design.

IV. Analysis of Experimental Results

A. Target Recognition Test

We use Machine element as recognition objects, collect 271 pictures with different characteristics for recognition test, and use the weight file of this design training to identify them. Some results are shown in Figure 16.

From Figure 16, it can be seen that the system has high recognition accuracy and confidence in parts under different backgrounds.

Select 10 images from the test set for statistics randomly, and the confidence level of each component is shown in Table 3. The first row of this table shows the part codes, the last row shows the average confidence level of each part, and the middle 10 rows show the confidence level of each part on each diagram. From Table 3, it can be seen that the confidence level of most parts is above 0.92, and the average confidence level of each type of part is greater than 0.91.

Figure 16: Identification Results

Table 3: Confidence of Each Part in Randomly Selected Test Results

Code	1204-1	1204-2	LM	LM-1	LM-2	CF	SCS	SHF	LMF	KC	SK
Confidence level	0.96	0.95	0.95	0.97	0.96	0.96	0.96	0.7	0.97	0.93	0.97
	0.89	0.89	0.83	0.97	0.94	0.95	0.96	0.93	0.98	0.92	0.98
	0.91	0.96	0.97	0.93	0.95	0.95	0.94	0.95	0.98	0.96	0.95
	0.94	0.93	0.93	0.96	0.96	0.95	0.95	0.91	0.98	0.94	0.96
	0.95	0.96	0.96	0.95	0.95	0.94	0.94	0.92	0.97	0.97	0.97
	0.91	0.84	0.90	0.96	0.95	0.92	0.95	0.93	0.95	0.95	0.97
	0.96	0.95	0.96	0.95	0.96	0.93	0.97	0.94	0.96	0.94	0.97
	0.96	0.95	0.95	0.95	0.94	0.95	0.96	0.93	0.82	0.94	0.97
	0.96	0.89	0.95	0.93	0.96	0.95	0.96	0.95	0.96	0.94	0.97
	0.92	0.92	0.94	0.95	0.91	0.91	0.96	0.94	0.97	0.94	0.96
Average value	0.936	0.924	0.934	0.952	0.948	0.941	0.955	0.91	0.954	0.943	0.967

B. Training Set Evaluation

In order to test the training set's ability to recognize images, this design also evaluated the YOLO object detection model. Using test.py to detect the images and labels in the test folder, and the results are as follows. Figure 17 and Figure 18 demonstrate the accuracy and confidence of recognition respectively. From these figures, it can be seen that the final model has high accuracy and confidence in identifying various parts.

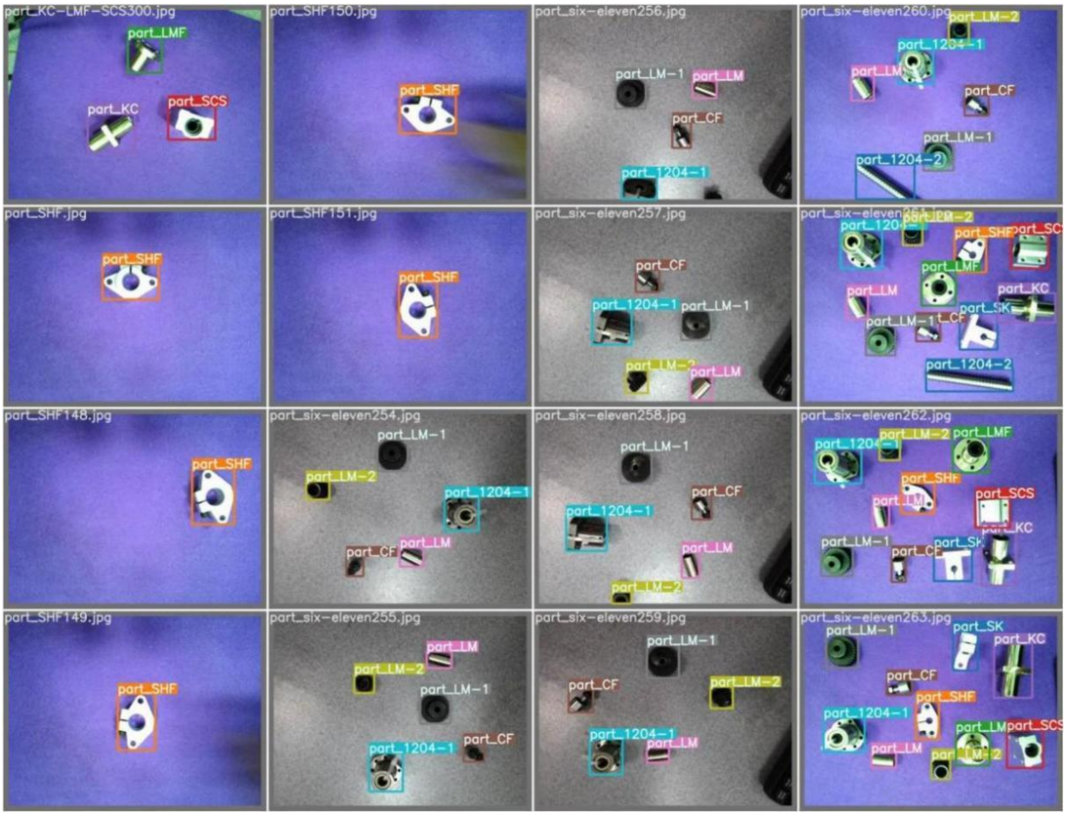


Figure 17: The Accuracy of Recognition Test

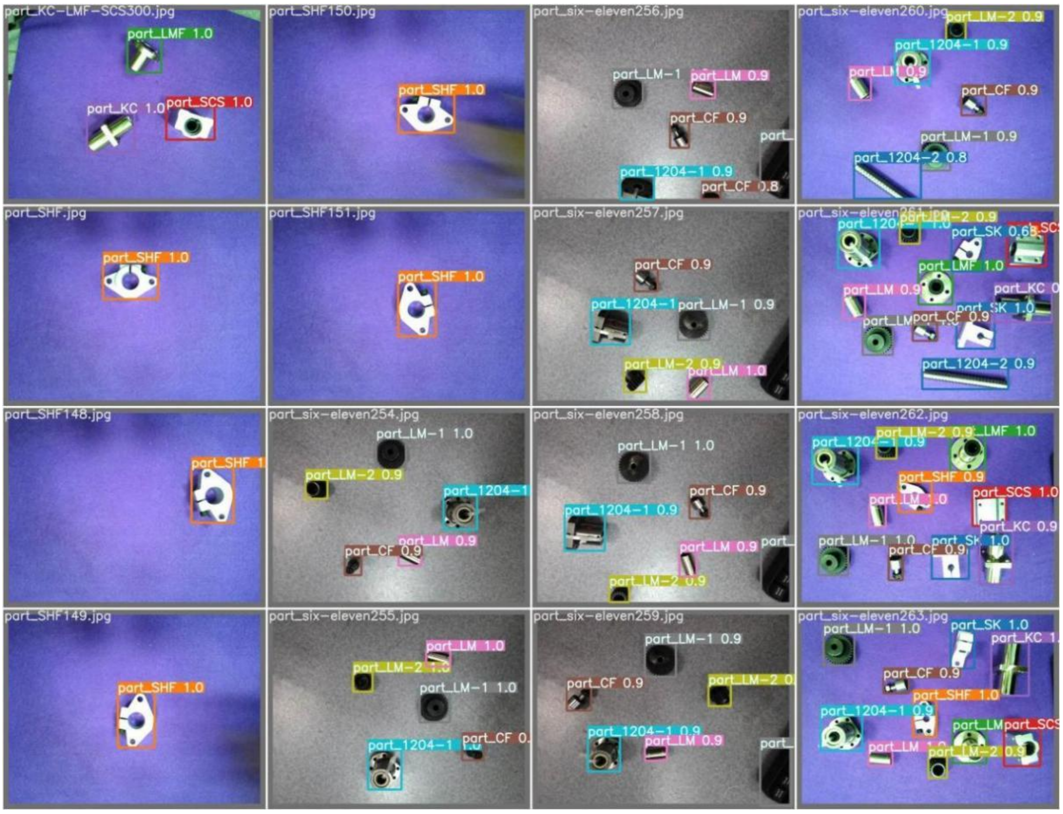


Figure 18: The Confidence of Recognition Test

Figure 19 shows the Confusion matrix tested, from which it can be clearly seen that the recognition accuracy of the system for part SK, part SHF, part LMF, part SCS, part KC, part CF, part LM, part LM-1, part LM-2, part 1204-1, and part 1204-2 is 0.98, 1, 1, 1, 0.9, 0.95, 0.95, 1, 1, 1, respectively. Except for part LM, the recognition accuracy of other parts is above 0.95, and the recognition effect is very ideal. Then we can export the YOLO trained model, deploy it to the hardware system, and optimize the model based on the test results.

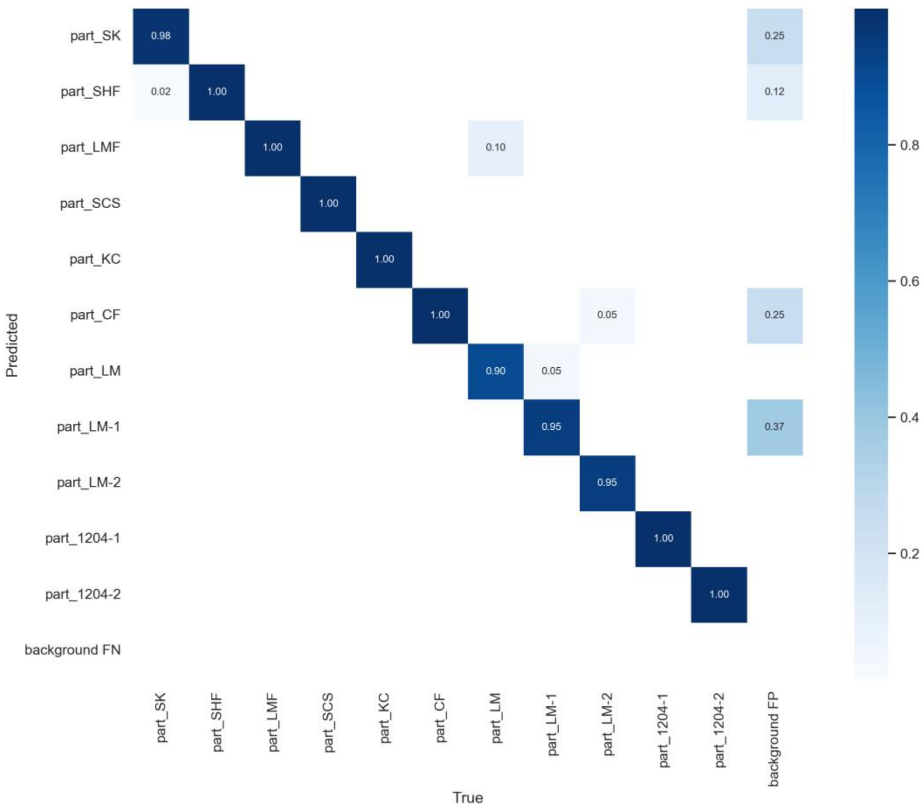


Figure 19: Confusion Matrix

V. CONCLUSION

Mechanical parts image recognition algorithm and hardware are designed in this paper. The main research conclusion s are as follows:

1. The image data is enhanced by image pre-processing for training with Histogram equalization, graying and image normalization to obtain higher recognition accuracy and robustness, compared with other image pre-processing methods.

2. YOLOv5 and SSD algorithms are used to recognize mechanical parts images. After repeated debugging, we obtain a weight file with high accuracy. After comprehensively comparing the advantages and disadvantages of the two algorithms, the one with relatively high accuracy and speed for subsequent operations is chosen. Identify the test set using the weight file obtained from training, and study the accuracy and speed. The training data set and weight file are evaluated, and Confusion matrix and related concepts are used to study the recognition accuracy and robustness. Combined with image preprocessing algorithm, high recognition accuracy is obtained by using YOLOv5 algorithm.

3. An embedded image recognition system is developed for easy installation and use in industrial applications.

ACKNOWLEDGEMENTS

The authors acknowledge the Scientific research project of colleges and universities in Anhui Province (No. 2022AH040044) and Scientific research project of Anhui Jianzhu University (No. JZ202284).

REFERENCES

[1] P. Burian, and R. Holota, Fast image recognition based on n-tuple neural networks implemented in an FPGA, J. Real-Time Image Pr., January 2016, 11: 155–166.

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” European Conference on Computer Vision, 2015.

[3] R. Panahi, and I. Gholampour, Accurate detection and recognition of dirty vehicle plate numbers for high-speed applications, IEEE T. Intell. Transp., April 2017, 18(4): 767–779.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas Convention Center, 2016.

[5] Min Weidong, Li Xiangpeng, Wang Qi,Zeng Qingpeng, Liao Yanqiu, A new approach to vehicle license plate location based on new model YOLO-L and plate pre-identification, IET Image Process., May 2019, 13(7): 1041–1049.

[6] R. Laroca, V. Barroso, M. A. Diniz, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, Convolutional neural networks for automatic meter reading, J. Electron. Imag., February 2019, 28(1): 1–14.

[7] S. Panigrahi, U. S. N. Raju, DSM-IDM-YOLO: Depth-wise separable module and inception depth-wise module based YOLO for pedestrian detection, Int. J. Artif. Intell. T., June 2023, 32(4): 235001.

[8] Wang Linyu, Jiang Qiangwei, Li Shuang, Research on fusion matching object recognition and location based on binocular image multi-feature point fusion, Radio Engin., July 2018, 48(8): 628–633.

[9] Wang Yina, Fu Guoqiang, A novel object recognition algorithm based on improved YOLOv5 model for patient care robots, Int. J. Hum. Robot., July 2022, 19(2) 2250010.

[10] Ying Zhiping, Lin Zhongtao, Wu Zhenyu, Liang Kuan, Hu XuDong, A modified-YOLOv5s model for detection of wire braided hose defects, Measurement, February 2022, 190: 110683.

[11] Yang Bin, Wang Jie, An improved helmet detection algorithm based on YOLOV4, Int. J. Found. Comput. S., September 2022, 33(06n07):887–902.

[12] Wang Hai, Yu YIjie, Cai Yingfeng, Chen Xiaobo, Chen Long, Liu Qingchao, A Comparative Study of State-of-the-Art Deep Learning Algorithms for Vehicle Detection, IEEE Intel. Transp. Sy., Summer 2019, 11(2): 82–95.

[13] Ye Ruifang, Pan Chia-Sheng, Hung Pei-Yuan, Chang Ming, Chen Kuan-Yu, Detection and classification of printed circuit board assembly defects based on deep learning, J. Chin. Soc. Mech. Eng., October 2020, 41(4): 401–407.

[14] Xupeng Kou, Shuaijun Liu, Kaiqiang Cheng, Ye Qian, Development of a YOLO-V3-based model for detecting defects on steel strip surface, Measurement, September 2021, 182: 109454.

[15] Wang Peng, Wu Jiao, Li Xiaoyan, Cai Mengyao, Fuzzy target detection algorithm based on improved SSD and transfer learning, J. Intell. Fuzzy. Syst., March 2021(6):1–12.

[16] Peng Hanyu, Chen Shifeng, BDNN: Binary convolution neural networks for fast object detection, Pattern Recogn. Lett., July 2019, 125: 91–97.

[17] K. D. Joshi, V. Chauhan, and B. Surgenor, A flexible machine vision system for small part inspection based on a hybrid SVM/ANN approach, J. Intell. Manuf., July 2020, 31(1):103–125.

[18] V. Q. Dinh, F. Munir, S. Azam, K.-C. Yow, and M. Jeon, Transfer learning for vehicle detection using two cameras with different focal lengths, Inform. Sci., April 2019, 514: 71–87.

[19] C. Cusano, and P. Napoletano, Visual recognition of aircraft mechanical parts for smart maintenance, Comput. Ind., April 2017, 86: 26–33.