International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 59-73

https://doi.org/10.70517/ijhsa46405

Research on automated visual defect detection method based on the combination of deep neural network and image processing algorithm

Jiale Bian[1,*]
[1] School of Computer Science and Technology, North University of China, Taiyuan, Shanxi, 030051, China
Corresponding authors: (e-mail: 2207024104@st.nuc.edu.cn).

Abstract In this paper, we utilize the features of YOLOv3 combined with multi-size prediction of feature pyramid network, which fuses the feature information of multiple feature maps of different sizes through up-sampling from top to bottom to improve the resolution of feature maps of different sizes. The converged target classification loss function and target regression loss function are used to train the YOLOv3 algorithm in combination with the homemade defect mapping dataset to meet the design requirements of the surface defect detection algorithm. Fine-tuning is applied as a pre-training model to optimize the loss function of the improved YOLOv3 algorithm. Analyze the experimental performance of the improved YOLOv3 algorithm with different Image Size parameters, different numbers of Anchors, and different sizes of defect areas. Compare the index performance of the improved YOLOv3 algorithm proposed in this paper with the YOLO series algorithms. The detection precision and recall of the improved YOLOv3 algorithm for different defective regions are 0.9324 and 0.8589, respectively, and the improved algorithm meets the requirements of defect detection algorithm design.

Index Terms yolov3 algorithm, feature pyramid network, loss function, fine-tuning, defect detection

I. Introduction

In recent years, with the rapid development of industrial technology and the increasing concern for product safety, product safety inspection in the industrial field has been emphasized by more and more manufacturers. The traditional manual inspection methods have high false detection rate and leakage rate, and at the same time, there are shortcomings such as slow detection speed and long detection time, which are difficult to meet the needs of large-scale production inspection [1], [2]. While machine vision is a series of judgments through machines instead of human eyes, the application of machine vision technology in defect detection can largely overcome the shortcomings existing in manual detection [3], [4].

Machine vision inspection technology is an emerging non-contact automatic inspection technology, which combines electronics, photoelectric detection, image processing and computer technology, is the most development potential of the new technology in the field of precision testing technology [5]. The principle is to photograph the inspection target by camera vision, and then the captured pictures are sent to image processing systems such as computers for feature extraction of the image, and then the defective part is segmented from the processed image [6]-[8]. The development of machine vision greatly reduces the contact damage to the device produced by manual inspection, while it detects fast efficiency, high accuracy, automation of the inspection process, easy to operate, by the majority of manufacturers of the favorite [9], [10]. At present, machine vision is becoming more and more attention, many manpower to complete the more dangerous work can be done with machine vision instead of complete, which largely improves the level of automation of production and the level of intelligence of the detection system [11]-[13]. However, in the actual detection process, the presence of a variety of factors can lead to the detection system signal-to-noise ratio is reduced, the detection of weak signals will have an impact, while the diversity of defects and the complexity of different detection backgrounds exist, have shown that the research and development of universal, intelligent and reliable detection system program is the development trend [14]-[17]. The selection of image acquisition equipment in the hardware module of the defect detection system directly affects the quality of imaging, and it is a direction that requires continuous efforts to improve the accuracy and robustness of the image processing algorithms through continuous optimization of the algorithms to make the detection system more intelligent [18]-[20].

With the continuous development of machine vision technology, the application of machine vision inspection technology in defect detection has become more and more widespread. Luo, Q. et al. applied machine vision

technology to automated defect detection on flat steel surface, using statistical methods, machine learning methods and other methods to extract image features in order to improve the environmental utility of automated visual defect detection [21]. Zhou, X. et al. investigated a visual inspection method for the bottom of glass bottles, combining the region of interest recognition technique with the wavelet transform multiscale filtering (WTMF) defect detection strategy, which effectively reduces the effect of the texture of the bottom of glass bottles on the defect detection effect and improves the robustness to the localization error [22]. Huangpeng, Q. et al. designed an unsupervised method based on a low-rank representation of texture prior for defect image analysis and used it to construct a weighted low-rank model of the detection process, which plays a great role in the task of detecting defects in manufactured products such as steel and fabrics [23]. Zhou, Y. et al. introduced numerous machine vision inspection methods for printed circuit board (PCB) defect detection and utilized them to empower image data acquisition, processing and analysis in PCB defect detection to contribute to intelligent manufacturing [24]. Qiu, K. et al. established an automated visual inspection framework for surface defects of metal parts, introduced a double-weighted principal component analysis algorithm in the image alignment module to obtain high-precision image alignment results, and proposed a defect detection module based on the image difference algorithm, which significantly improves the defect detection accuracy [25]. Li, C. et al. designed a fabric defect detection algorithm based on bio-visual modeling by simulating the bio-visual perception mechanism, and the algorithm showed high performance in terms of adaptability and detection efficiency for detecting defects in patterned fabrics with or without texture complexity [26]. Since machine vision technology is a cross technology involving multiple disciplines, the image acquisition quality and analysis performance of the above inspection system are affected to varying degrees, impacting the reliability of the inspection system.

Deep learning-based image processing algorithms can make machine vision technology more automated and intelligent by continuously improving the accuracy and execution efficiency of the algorithms, and a large number of scholars apply them to the field of defect detection and promote the research and improvement of related technologies. Czimmermann, T. et al. examined the performance of supervised and unsupervised classifiers as well as deep learning techniques in visual defect detection and classification tasks for industrial applications [27]. Ren, Z. et al. explored the application of deep learning technology in the field of defect detection, showing that image processing and analysis supported by deep learning technology is the key to improving the efficiency, quality and reliability of defect detection [28]. Jha, S. B. and Babiceanu, R. F. illustrated the advantages and challenges of the application of Convolutional Neural Networks (CNN) in modern visual defect detection, supervised and unsupervised learning models based on CNN algorithms can help to improve the accuracy of defect detection, but of course coping with the industrial practice academic research will also face more challenges [29]. Tang, H. et al. proposed an improved YOLOv5-C3CA-SPPF network model for the problem of unrecognizable low-contrast, multi-scale targets in optical lens products, which is able to effectively and quickly detect surface and internal defect types of lenses compared to traditional defect detection algorithms so as to safeguard product quality [30]. Singh, S. A. et al. proposed an image-based pre-trained convolutional neural network (CNN) framework, ResNet-101, which is utilized to extract features from image data and combined with a multi-class support vector machine (SVM) for classification and detection to achieve high-precision defect detection results for small training datasets [31]. Wang, T. et al. surface that previous visual defect detection methods are more dependent on manually extracted optical features, for this reason, a deep convolutional neural network is proposed to realize the automatic extraction of image features, and at the same time, the noise immunity of the image features is strengthened, which provides a solid technological foundation for carrying out fast and high-precision visual defect detection [32]. Yuan, Z. C. et al. utilized a deep neural network based defect extraction and measurement method to generate a large amount of image data to provide a data base for the training and learning of an automated defect detection system [33]. Westphal, E. and Seitz, H. proposed a complex migration learning method for selective laser sintering process, which effectively realizes automatic classification of image features for small datasets and achieves better quality control and defect detection [34]. Not surprisingly, deep learning techniques enhance the ability to process image data to ensure that defect detection systems are able to detect defective targets quickly and accurately, making it all the more important to build on existing techniques to face the challenges that come with them.

In this paper, we create our own defect mapping dataset and preprocess the mapping data. The detection ability of YOLOv3 algorithm for target objects and the network architecture of YOLOv3 are analyzed. A multi-scale prediction network is designed by adding feature pyramid network to YOLOv3 algorithm, and the multi-scale prediction network architecture in YOLOv3 is drawn. The same deep convolutional network is used to complete the target classification and target regression tasks, and the loss function of multi-task is implemented to complete the optimization of defect detection technology based on YOLOv3 algorithm. Pre-training of the improved YOLOv3 algorithm using fine-tuning, and the homemade dataset is brought into the training of the improved YOLOv3

algorithm to get the best loss function. Compare the metric performance of the original YOLOv3 algorithm with the improved YOLOv3 algorithm.

II. Algorithm proposal

II. A. Deep Neural Network Basic Theory

Deep neural network is a neural network model with a large number of hidden layers, the "depth" covers a number of consecutive representation layers, which can perform continuous nonlinear transformations on the original input data, enabling each layer to learn and extract feature information. Compared with shallow neural networks, deep neural networks can better deal with complex nonlinear problems and have stronger feature extraction and feature representation capabilities. Deep neural network is an implementation of deep learning, and its ability to express image feature information is gradually increasing. Compared to traditional algorithms deep learning based super-resolution algorithms have shown obvious superiority [35].

II. A. 1) Deep Learning Overview

Deep learning is a powerful machine learning methodology built with neuronal networks of the human brain as inspiration. It designs multi-layered network structures and learns to acquire multi-layered abstraction of features on the input data so that the model can understand and recognize the feature input.

Deep learning can design a wide variety of network structures to accomplish specific tasks, and for image super-resolution, specific network structures can be designed to extract high-frequency detail information from images to improve the clarity of reconstructed images.

With the development and innovation of technology, there have been deep learning architectures that have been researched and developed. Examples include convolutional neural networks applied in visual tasks, generative adversarial networks used in super-resolution, image denoising, and data enhancement, and recurrent neural networks, which are widely used in natural language processing.

Deep neural networks can be regarded as an extension of multilayer perceptual machines (MLPs), where more convolutional layers, neurons, activation functions, optimization techniques, and model design are introduced to the MLP, giving it greater expressive power and the ability to learn complex models. Deep neural networks can be trained by back propagation algorithms and gradient descent methods, enabling them to handle large datasets, and the field of image super-resolution often requires a large amount of image data for training. The deep neural network structure is shown in Figure 1.

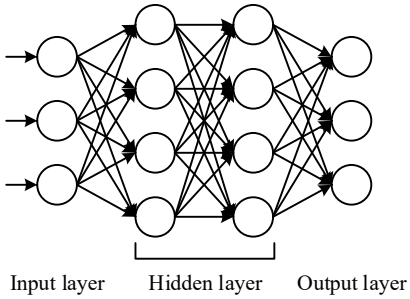


Figure 1: Deep neural network structure

If x_i denotes the input from the i th neuron, w_i denotes the weight value of the i th neuron, b denotes the bias, θ denotes the threshold, and ξ is the activation function, the output y can be expressed as:

$$y = \xi\left(\sum_{i=1}^{n}(w_i x_i + b) - \theta\right) \tag{1}$$

To make the output value y approximate the true value t, the deep neural network defines the loss function $\xi(w,b)$ in terms of the squared difference between the two, and the process can be expressed as follows:

$$\xi(w,b) = \frac{1}{2n}\sum_{x}\| y - t \|^2 \tag{2}$$

The training objective of the deep neural network is to minimize the loss function, and the weight w and bias b of the point of the minimum are determined by the gradient descent method. In the $\xi(w,b)$ multivariate function, the independent variables w and b corresponding to the minima are obtained by the inverse gradient descent

method. The gradient value of the loss function is a vector that can represent the rate of change of the loss function at the point when the rate of change is maximum, and its direction represents the direction of maximum change. Therefore, the essence of gradient descent is to iteratively search for the weight w and bias b corresponding to the point of minimum value along the reverse direction of the gradient, which is constantly updated. Using η to denote the learning rate, the process of updating the parameters by the gradient descent method of deep neural network can be expressed as:

$$w_K \to w_K^{'} = w_k - \eta \frac{\partial f}{\partial w_k} \tag{3}$$

$$b_s \to b_s^{'} = b_1 - \eta \frac{\partial f}{\partial b_s} \tag{4}$$

II. A. 2) Convolutional Neural Networks

CNN is a feed-forward neural network, which is able to reduce the dimensionality of the data and the number of parameters of the network through the local connection of the convolutional kernel, shared weights and pooling layer downsampling, which not only improves the computational efficiency and ensures the model's generalization ability, but also reduces the possibility of the occurrence of overfitting. The convolutional layer can utilize the characteristics of the input image to design the neurons as a three-dimensional tensor, including width, height and depth [36].

CNN mainly consists of convolutional layer, activation layer, pooling layer and fully connected layer. By combining them in a regular architecture for specific usage scenarios and setting appropriate network hyperparameters, a convolutional neural network model can be constructed to be applied to different computer vision tasks, and it is able to obtain the deep robust features of the image information.

II. B.Defect Mapping Data Acquisition and Preprocessing

II. B. 1) Graph data acquisition

In this paper, the welding defects in the welded structure of shaped welded bearing parts are studied. The M2M tabletop ultrasonic phased array detector was used to acquire 3257 ultrasonic full-focus images of internal defects in welded seams by changing different scanning angles using a 10 MHZ probe and a 5 MHZ probe.

The principle of ultrasonic phased array imaging is to scan the workpiece weld to obtain defect images by controlling the delay time of the excitation or reception pulse of each chip array element of the phased array transducer and changing the phase relationship of each chip array element that transmits the pulse signal or receives the echo signal. The full-focus imaging algorithm based on the full-matrix acquisition model is applied. Compared with the traditional ultrasonic inspection technology, the full-focus algorithm imaging can fully utilize the data information of the phased array ultrasound, and it can respond to the characteristics of the defects such as the size and orientation in a more detailed and precise way.

There are 5 types of internal defects in the weld seam provided by the defect data, including "slag inclusion", "crack", "porosity", "not penetrated" and "not fused".

II. B. 2) Spectral data preprocessing

In order to obtain a better neural network model and avoid the low adaptability of the model due to too few samples, the training dataset is usually required to be large enough, and the quality of the image will also directly affect the accuracy of the recognition algorithm's results. Therefore, there is a key step before intelligent recognition and classification of captured images: image preprocessing.

Commonly used image preprocessing methods include “gray scale transformation”, “geometric transformation”, “image enhancement” and “data labeling”.

(1) Gray scale transformation

As the collected weld defect image is a color image with R, G, B three channels of data, so it is necessary to process all three channels of data, which will increase the time required to train the model, after grayscale processing, the grayscale image containing only the brightness information can reduce the amount of data computation and improve the speed of the algorithm. The following formula is used for color image to grayscale image:

$$Y = 0.299R + 0.578G + 0.114B \tag{5}$$

Y represents the pixel gray value, and R, G, and B represent the red, green, and blue channels of the color image, respectively.

(2) Geometric transformation

The geometric transformation of an image is also known as a spatial transformation, that is, the acquired image is "translated", "transposed", "mirrored", "rotated", and "scaled". The purpose of geometric transformations is to correct for systematic and random errors.

(3) Image enhancement

Image enhancement is designed to make the original unclear image clear or emphasize the features of certain regions of interest, enhance the completeness or localization of the image, extend the differences between the features of different objects in the image and suppress the features of uninterested objects. Image enhancement can improve the visual effect, interpretation and recognition of the image. Image enhancement algorithms can be generally categorized into: “null domain methods” and “frequency domain methods”.

(4) Data labeling

After the above processing, 5224 sample images are obtained, and the sample images are divided into training set, validation set and test set according to 7:2:1.

Finally, all the defect images in the data set are manually labeled according to the standard data set labeling format, and the Label Img labeling tool is used to label the defects in the sample images in terms of category, location, size and other information, and then generate xml files corresponding to the images for subsequent algorithm training, validation and testing.

II. C.Surface defect detection algorithm based on YOLOv3

II. C. 1) Principles of the YOLOv3 algorithm

YOLOv3 combines the features of feature pyramid network (FPN) with multi-size prediction, and fuses the feature information of three different sizes (13*13, 26*26, and 32*32) feature maps through up-sampling in a top-down manner, which ensures that each size feature map has the appropriate resolution and semantic information to be used for detecting the target object of the corresponding resolution size. The introduction of the feature pyramid idea greatly improves the detection capability of the YOLOv3 algorithm, especially for small-sized objects [37], [38].

In order to adapt to the feature pyramid structure, YOLOv3 presets 9 a priori frames with different size sizes by clustering. Then the 9 a priori frames are divided into three sizes of large, medium and small to be assigned to 13*13, 26*26 and 52*52 feature maps respectively. And the output tensor of YOLOv3 network was changed to $S\times S\times\left[3\times\left(4+1+C\right)\right]$. Where $S\times S$ is the number of grids after the feature map is divided, and $3\times\left(4+1+C\right)$ denotes the location, size, confidence, and categorical probability information contained in the three different sized prediction boxes in each grid.

The YOLOv3 algorithm uses a binary cross-entropy function rather than a softmax function to compute the category loss of the prediction frame. Because softmax is set to have only a single category attribute for each detected object, however, in some cases an object may have multiple category attributes, which leads to its ineffectiveness in categorizing some objects with complex relationships in the image. The binary cross-entropy function can solve this problem well, so the binary cross-entropy function can achieve better detection results on datasets with more complex object category relationships.

The network architecture of YOLOv3 is inspired by ResNet, and Darknet53, which has deeper network layers and better effect on image feature extraction, is used as the backbone network for feature extraction. Darknet-53 prevents performance degradation due to network deepening by making extensive use of the residual structure in ResNet to fuse feature information from feature maps of different depths. And YOLOv3 no longer uses the pooling layer in YOLOv2, and its downsampling of the feature map is achieved by changing the step size of the convolution kernel.

II. C. 2) Multi-scale prediction networks

The feature extraction process of convolutional neural network on an image is actually a downsampling process of the image, so multiple sizes of features can be conveniently obtained using convolutional neural network to form a feature pyramid. In general, large-size feature maps contain more detailed feature information in the image, while small-size feature maps have stronger semantic expression. The small-size feature maps are up-sampled and fused with the large-size feature maps for feature fusion, and the new features after fusion tend to have better feature expression ability.

Based on the idea of feature pyramid network, YOLOv3 fuses three different scales of feature maps and designs a multiscale prediction network, which improves the target detection accuracy of YOLOv3, especially significantly improves the detection accuracy of small targets.The multiscale prediction network in YOLOv3 is shown in Fig. 2.

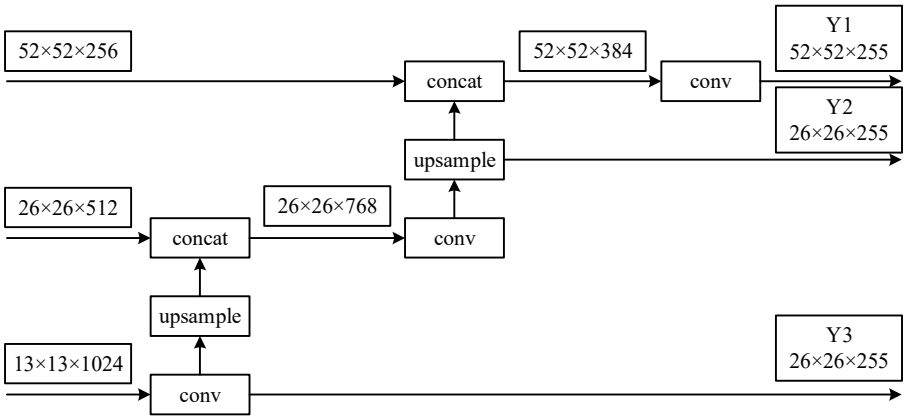


Figure 2: Yolov3 multi-scale prediction network

There are three prediction branches in YOLOv3, which are Y1, Y2, and Y3. Each prediction branch has the same prediction principle, and this paper introduces the prediction principle of YOLOv3 by taking the feature map of size 13*13 as an example. The 13*13 feature map can be viewed as dividing the input image into 13*13 grids, and the grid is responsible for predicting the current target in whichever grid the center of the target in the input image falls into. The grid uses the width and height of the a priori box, the coordinates of the center point of the grid, and the offset of the network prediction to obtain the prediction box when making the prediction of the target. The prediction box is computed as follows:

$$\begin{cases} b_x = \sigma(t_x) + c_x \\ b_y = \sigma(t_y) + c_y \\ b_w = p_w e^{t_w} \\ b_h = p_h e^{t_h} \end{cases} \tag{6}$$

where c_x, c_y denotes the coordinates of the upper left corner of the grid responsible for predicting the target, and t_x, t_y denotes the coordinate offsets learned by the network. The σ denotes the Sigmoid function that normalizes t_x, t_y to between 0 and 1, ensuring that the target can only be predicted by the current grid. Using t_x, t_y and Sigmoid function to calculate the centroid coordinates of the prediction frame in the feature map, based on the mapping relationship between the feature map and the input image, the centroid coordinates of the target prediction frame in the input image can be obtained. The p_w, p_h denote the width and height of the a priori frames. YOLOv3 draws on the a priori frame mechanism of the FasterR-CNN algorithm to preset nine kinds of a priori frames in advance for target prediction at three scales, and three a priori frames are used in each prediction branch.

YOLOv3 gives the corresponding confidence level confidence, and the target belongs to the category C. Confidence has two meanings, the first meaning indicates whether the current prediction frame contains the target or not. The second meaning is if the current prediction frame contains the target. Confidence is calculated as follows:

$$confindence = p_r(Object) * IOU_{truth}^{predict} \tag{7}$$

where $p_r(Object)$ indicates whether the prediction frame contains the target. The $p_r(Object) = 1$ indicates that the current prediction frame contains the target. The IOU_{tmuth}^{predct} indicates the intersection ratio of the predicted box to the true box, which is used to measure the confidence level that the predicted box contains the target.

II. C. 3) Loss function

YOLOv3 accomplishes both target classification and target regression tasks using a single deep convolutional network, and realizes a multi-task loss function by unifying the target classification loss function and target regression loss function into the same loss function.

YOLOv3 completed the target regression task using t_x, t_y, t_w, t_h from the prediction results, and completed the target classification task using the confidence and conditional probability information of all categories from the prediction results. Therefore, the error between the network prediction results and the real results mainly appears in the six categories of data such as t_x, t_y, t_w, t_h, CONFIDENCE, and all categories of conditional probabilities.The

loss function of YOLOv3 consists of three main parts, which are t_x, t_y, t_w, t_h brought about by the location error, confidence error and the error brought about by the categories, the loss function L of YOLOv3 is:

$$L = \lambda_{coond} L_{box} + \lambda_{conf} L_{confidence} + \lambda_{cls} L_{class} \tag{8}$$

where L denotes the YOLOv3 total loss function and denotes the position loss function. $L_{confidence}$ denotes the confidence loss function, and L_{class} denotes the classification loss function. $\lambda_{coonl}, \lambda_{conf}, \lambda_{conf}, \lambda_{cls}$ denote the proportionality coefficients of the three types of loss functions in the total loss function, respectively.

The L_{box} is divided into two major parts for computation, which are the coordinate error of the center point of the border brought about by t_x, t_y, and the error of the width and height of the border brought about by t_w, t_h, which are computed by Eqs:

$$L_{xy} = \sum_{i=0}^{S\times S}\sum_{j=0}^{B} I_{ij}^{abj}(2 - w_i \times h_i)\begin{bmatrix}(-\hat{x}_i * \log(x_i) - (1-\hat{x}_i) * \log(1-x_i)) + \\ (-\hat{y}_i * \log(y_i) - (1-\hat{y}_i) * \log(1-y_i))\end{bmatrix} \tag{9}$$

$$L_{wh} = \sum_{i=0}^{S\times S}\sum_{j=0}^{B} I_{ij}^{obj}(2 - w_i \times h_i)\left[(w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2\right] \tag{10}$$

where $S \times S$ denotes the size of the feature map and B denotes the number of a priori frames in each prediction branch. I_{ij}^{obj} denotes whether the j th a priori frame of the i th grid is responsible for predicting the current target, if the a priori frame is responsible for predicting the current target then $I_{ij}^{obj} = 1$, and vice versa $I_{ij}^{obj} = 0$. x_i, y_i, w_i, h_i denote the predicted values of the edges, and $\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i$ denote the true values of the edges. The $(2 - w_i \times h_i)$ can be viewed as a weighting factor, which is used to increase the proportion of smaller sized edges in the loss function. The L_{xy} and L_{xy} have the same proportion in L_{box}, and the bezel loss function L_{box} is calculated as follows:

$$L_{box} = L_{xy} + L_{wh} \tag{11}$$

The $L_{confidence}$ consists of two components, the confidence error from the edges that are responsible for predicting the target and the confidence error from the edges that are not responsible for predicting the target. They are calculated by the formula:

$$L_{confidence\_obj} = \sum_{i=0}^{S\times S}\sum_{j=0}^{B} I_{ij}^{obj}[-\hat{C}_i * \log(C_i) - (1-\hat{C}_i) * \log(1-C_i)] \tag{12}$$

$$L_{confidence\_noobj} = \sum_{i=0}^{S\times S}\sum_{j=0}^{B} I_{ij}^{noobj}[-\hat{C}_i * \log(C_i) - (1-\hat{C}_i) * \log(1-C_i)] \tag{13}$$

where I_{ij}^{noobj} denotes whether the j th a priori frame of the i th grid is responsible for predicting the current target, if not then $I_{ij}^{noobj} = 1$, and vice versa $I_{ij}^{noobj} = 0$. C_i denotes the predicted value of confidence and $\hat{C}_i$ denotes the true value of confidence. The confidence loss function is:

$$L_{confidence} = \lambda_{obj} L_{confidence\_obj} + \lambda_{noobj} L_{confidence\_noobj} \tag{14}$$

where λ_{obj} and λ_{noobj} are the proportionality coefficients of the two loss functions in the confidence loss.YOLOv3 defaults to the fact that the two loss functions contribute the same amount to the confidence loss, so $\lambda_{obj} = \lambda_{n\alpha obj} = 1$.

The L_{class} classification loss function is:

$$L_{class} = \sum_{i=0}^{S\times S}\sum_{j=0}^{B} I_{ij}^{obj} \sum_{c\in classes}[-\hat{p}_i(c) * \log(p_i(c)) - (1-\hat{p}_i(c)) * \log(1-p_i(c))] \tag{15}$$

where $p_i(c)$ denotes the predicted value of the target category and $\hat{p}_i(c)$ denotes the true value of the target category.

III. Algorithm experimentation and analysis

III. A. Experimental correlation

To ensure the future portability of the model, different training and testing environments are used for the experiments in this paper. The training and validation environments use Colab Notebook provided by Google Inc. and its environment and configuration are shown below.

The training and validation configurations are shown in Table 1.

Table 1: Training and validation configuration

Training environment	Colab notebook
CPU	-
Memory	24G
GPU	Tesla T4
Show off	32G
Third-party library	Pytorch,Cuda,Caffe,Opencv,Matlab

The test was conducted in a local environment using Windows 10 system, as shown in Table 2.

Table 2: Test hardware configuration

Operating system	Ubuntu18.04
CPU	Intel i7 8300H
Memory	32G
GPU	RTX 2060
Show off	8G
Third-party library	Pytorch,Cuda,Caffe,Opencv,Matlab

In order to quantify the performance capability of the model detection model, this paper uses mAP and F1-Socre as the evaluation metrics of model capability. The two are calculated by the formula:

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{16}$$

$$F1 = \frac{2PR}{P+R} \tag{17}$$

where N represents the number of wheat ears detected and AP represents the trade-off between accuracy and recall. The formula is:

$$AP = \int_0^1 P(R)dR \tag{18}$$

The use of PR in the calculation of Fl-Score and AP refers to the precision and recall. The formula for this is:

$$P_{recision} = \frac{TP}{TP+FP} \tag{19}$$

$$R_{ecal} = \frac{TP}{TP+FN} \tag{20}$$

where true positive (TP) indicates that the sample was predicted to be correct and was actually positive. False positive (FP) indicates that the sample was predicted to be positive but was actually negative. In addition, false negative (FN) indicates that the sample was predicted to be negative but was actually positive.

In target detection, TP, FP, and FN were redefined by intersection and integration ratio (IOU). The redefined concepts are:

TP: the number of prediction frames with IOU > 0.5 from the true bounding box.FP: the number of prediction frames with IOU <= 0.5 from the true bounding box, or prediction frames with no object in the prediction frame.FN: the number of true bounding boxes that are not detected.

Currently, IOU is commonly used in target detection tasks to determine the degree of overlap between the predicted bounding box and the true bounding box.IOU is denoted as:

$$IOU = \frac{A \cap B}{A \cup B} \tag{21}$$

where A is the predicted bounding box and B is the bounding box given by the label.

III. B. Model Training

Network training is a top priority to be able to meet the requirements when modeling the surface defect detection network. When training the model, a combination of using pre-training and self-built dataset training is used. That is, a publicly available dataset is used to train the network first. Then the parameters in the front layer of the network are saved and discarded from the back layer of the network. Finally, the network is trained again with the new dataset, and the model trained at this time is used as the final model.

The advantage of this is that the public dataset can be used to train the front layer of the network, which is responsible for the low-level features of the samples, and the self-constructed dataset can be used to train the back layer of the network, which is responsible for the high-level features of the samples. On the one hand, it can greatly reduce the demand for data, making it possible to obtain a high network recognition rate with less network data. On the other hand, the training speed of the network can be greatly accelerated by replacing the learning of the entire deep neural network with the training of certain layers in a deep neural network, which reduces the number of network participants and the demand for hardware.

The pre-training model, also called fine-tuning, is a very commonly used means of tuning the parameters. In practice, since the dataset for training the network is not large enough, if it is used directly for network training there is a possibility of serious overfitting and the time to retrain a new network is too long. Therefore pre-training one's model through some online public datasets, this method of pre-training the model is called fine-tuning.

After obtaining the pre-trained model for model surface defect detection, the self-built dataset can be used for secondary training. After going through multiple training sessions, the parameters are set as shown in Table 3.

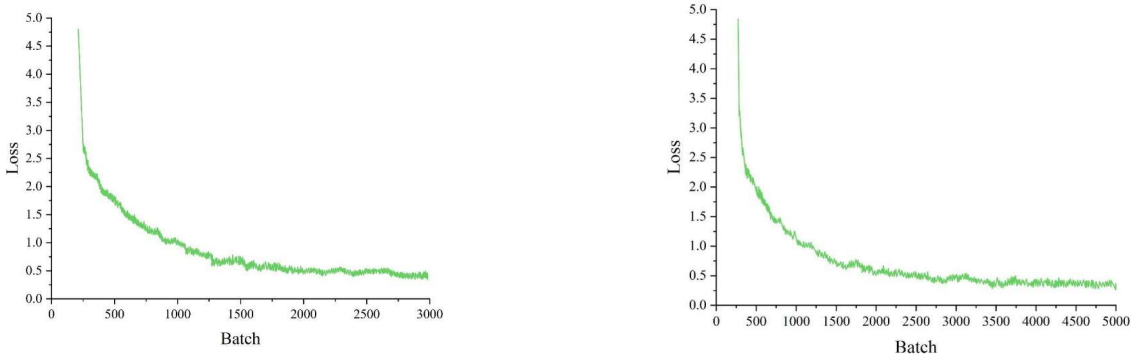
IMAGE_SIZE=400 means that the input image size is defined as 400*400*3. CELL_SIZE=8 means that the input image will be partitioned into 8*8 grids. BOXES_PER_CELL=2 means that 2 bounding boxes will be generated for each grid, so that a maximum of 128 bounding boxes will be generated for one image.

LEARNING_RATE=0.0001, DECAY_STEPS=0.1, ITERATION=5000 means that the network will be trained 5000 times with a learning rate of 0.0001 and a decay rate of 0.1. BATCH_SIZE=50 training means that the data is fed to the neural network in the form of one BATCH by one BATCH. Where every 50 samples in the dataset are divided into one BATCH.

Table 3: Training model parameters

CELL_SIZE=8	BATCH_SIZE=50
BOXES_PER_CELL=2	LEARNING_RATE=0.0001
GPU=1	ITERATION=5000
IMAGE_SIZE=400	

The training results of the model in this paper are shown in Fig. 3, Figs. (a) and (b) show the model training process and the final training results of the model, respectively.



(a) Training process (b) Final training results

Figure 3: The training results of this model

Figure (a) shows the trend of the loss function, and the total number of iterations is 3000 batch.

Figure (b) shows that the model surface defect model of this paper was trained for 5000 batch at a learning rate of 0.0001, which took 8h to complete the training. From the figure, it can be seen that the loss function when the network is first trained is far beyond the vertical coordinate of the loss function plot. At 302 iterations the loss function drops to less than 5. As the number of iterations increases it keeps an oscillating downward trend. The curve fluctuates considerably at 1750 and 3500 iterations. The loss function oscillates around 0.5 and then starts to oscillate down again, and finally the loss function fluctuates between 0.4 and 0.5 at 3500 iterations. Finally, when the number of iterations reaches 4000, the loss function stays at 0.3508, and the secondary training of the model is completed.

III. C. Experimental results

III. C. 1) Different parameters

Here we mainly explore the Image Size parameter selection, and its experimental results with different parameters.

The experiments were conducted using 64*64, 128*128 and 256*256 image size training sets, and tested the loss value test and mAP test of the model after training with three different image sizes. (The three different image sizes were derived by OPENCV's resize interpolation algorithm).

The changes of loss and m AP values for 64*64 size image resolution are shown in Fig. 4. It can be seen that the loss value of 64*64 size is close to the convergence of 1.5, but the downward trend fluctuates a lot.

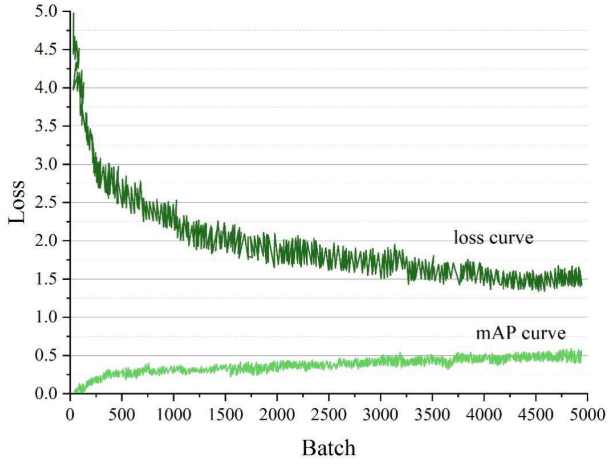


Figure 4: The loss value and m a pvalue change in the image resolution of 64*64 size

The variation of loss value and m AP value for 128*128 size image resolution is shown in Fig. 5. In the figure for the model with 128*128 image size, the loss value decreases to near 1.2~1.4, and the fluctuation becomes less when the loss value converges. m AP value curve increases with the number of iterations.

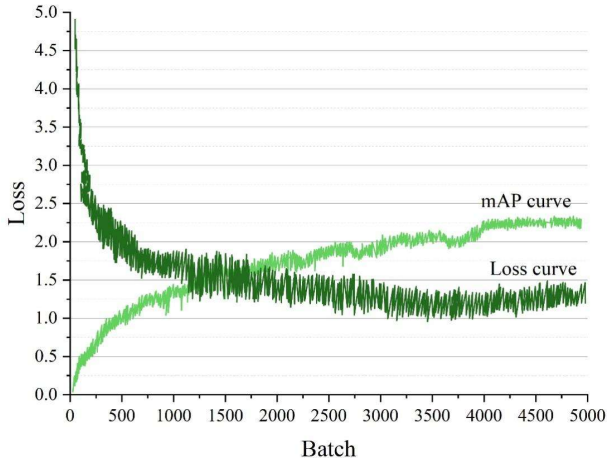


Figure 5: The loss value and m a pvalue change in the image resolution of 128*128

The variation of the loss value and m AP value at the image resolution of 256*256 size is shown in Fig. 6.

Observing the 256*256 image, it is found that the loss value can converge to about 0.87, and the fluctuation at convergence has a smaller jump relative to the previous models.

Observing the mAP curves of the three models, it is found that as the image resolution gets higher and higher, the recognition accuracy also gets higher and higher.

Compared with the YOLOv3 base network model, it has different sensitivity to different resolutions. After comparing each structural parameter between the two models, it is found that in the structure of the convolutional layers, the convolutional kernel of the improved YOLOv3 in this paper is smaller than that of the traditional YOLOv3, and the number of convolutional layers is more than that of the traditional YOLOv3. This can often make the improved YOLOv3 network model extract deeper features than that of the traditional YOLOv3, so the improved YOLOv3 network model is able to recognize more features at a more features at a larger image resolution, which can lead to a higher recognition rate.

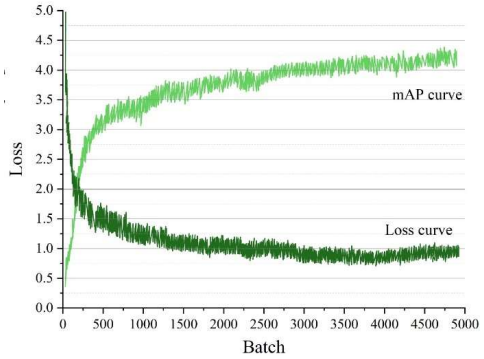


Figure 6: The loss value and m a pvalue change in the image resolution of 256*256

Anchors are a manifestation of the inference ability of convolutional neural networks.The principle of Anchors in Yolo is to resize inputs of different sizes into outputs of the same size, so that feature maps of any size can be converted into fixed-size feature vectors for judgment.

Anchors may be larger or smaller than a set confidence value. If the probability of Anchors is larger than the confidence value, it is equivalent to recognizing the region of the target and judging the whole by its parts. If it is smaller than Confidence value, it is that the target region is larger than the detected region, from judging which the detected region is not the target that needs to be detected in this paper.

And the number of Anchors as well as the size, need to be determined by K-means.K-means algorithm is very typical distance-based clustering algorithm, using distance as the evaluation index of similarity, that is, it is considered that the closer the distance between two objects, the greater their similarity. The algorithm considers that clusters are composed of objects that are close in distance. Therefore, it takes getting compact and independent clusters as the ultimate goal, and the value of the most suitable ANCHORS at the final fit can be calculated as the IOU value (the overlapping rate between the candidate boxes and the original labeled boxes).

This test is divided into 14 tests respectively Anchors are 1 to 14 and they use K-means algorithm respectively. The size of the Anchors and the maximized IOU are calculated from the defective images.The final practical results are shown in Table 4. The m AP value is stabilized at around 80% for multiple tests. As the size of the Anchors changes, the Iou value produces changes. After 14 tests, the Iou value shows 85.15%.

It can be seen that as the number of Anchors increases, the rate of overlap (IOU) of candidate frames with the original labeled frames also increases, but the correctness rate reaches the highest mAP=86.68% with Anchors=13.

Table 4: Changes in IOU and mAP in different anchors

Anchors number	1	2	3	4	5	6	7
Iou(%)	0.3251	0.4579	0.5636	0.6012	0.6359	0.6828	0.6978
m AP(%)	0.8252	0.8263	0.8419	0.8334	0.8207	0.8124	0.7885
Anchors number	8	9	10	11	12	13	14
Iou(%)	0.7253	0.7548	0.7696	0.7989	0.7992	0.8024	0.8515
m AP(%)	0.7556	0.8436	0.8221	0.8239	0.8455	0.8668	0.8483

By adjusting the confidence parameter, the model was tested for accuracy (how many of the samples predicted as defective were truly defective samples) and recall (how many of the number of defects in the sample were predicted correctly).

The P-R curves are shown in Figure 7, where (with an IOU of 50%) the P-R curves of the model were tested for Anchors of 3-14, respectively. It can be seen that the highest correctness rate is about 96%. Finally, considering the recall (recall) and correctness (mAP), the confidence parameters of 90% for mAP and 89% for recall were chosen.

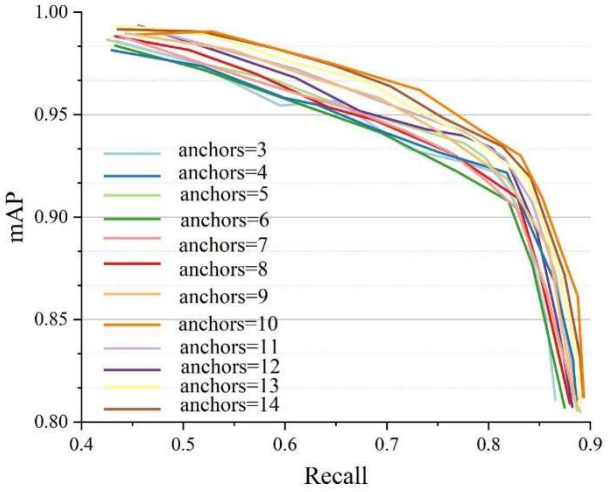


Figure 7: P-R curve

Finally the correctness of the model for different defect areas was tested separately according to their size. The accuracy and recall under different sizes of defects are shown in Table 5. In the table, it is found that the region with pixels smaller than 16*16, the worse the recognition rate is, and the recognition rate is below 90%. Defects larger than 40*40 pixels can achieve high recognition rate. In the region of 1024*1024 pixels, the recognition rate of this paper's model is 0.9983 and the recall is 0.9657.

Table 5: Accuracy and recall rate of defects under different sizes

Defect size(pixel.)	All	2*2	4*4	8*8	16*16
Accuracy rate	0.9324	0.6231	0.8702	0.8990	0.9124
Recall rate	0.8589	0.5089	0.8113	0.8365	0.8565
Defect size(pixel.)	32*32	40*40	64*64	90*90	128*128
Accuracy rate	0.9337	0.9546	0.9637	0.9661	0.9700
Recall rate	0.8421	0.8571	0.8967	0.8507	0.9042
Defect size(pixel.)	180*180	200*200	256*256	512*512	1024*1024
Accuracy rate	0.9878	0.9892	0.9904	0.9954	0.9983
Recall rate	0.9084	0.9115	0.9306	0.9447	0.9657

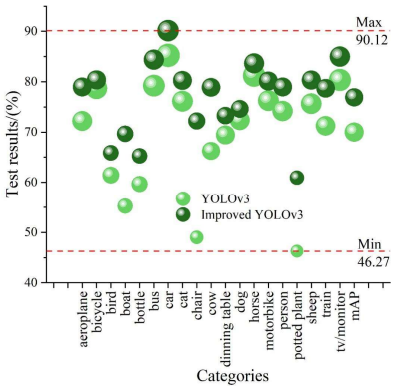


Figure 8: The two algorithms are tested in different categories

III. C. 2) Objective evaluation

The experiments use the accuracy rate as a performance evaluation metric to compare the original YOLOv3 algorithm as well as the improved YOLOv3 algorithm.

The test results of the two algorithms on different categories are shown in Fig. 8, where mAP is equal to the average of the sum of AP values of each category. The mAP values of the original YOLOv3 algorithm as well as the improved YOLOv3 algorithm are 70% and 76.91% respectively. It can be seen in the figure that the maximum value of the test results comes from the improved YOLOv3 algorithm and the minimum value of 46.27% comes from the classification results of the original YOLOv3 algorithm for the potted plant category.

III. C. 3) Comparison experiments

Since the target detection algorithms are all compiled on different languages, frameworks and tested and evaluated in different hardware environments. In order to evaluate the performance of the methods in this paper more fairly as well as more comprehensively, this paper is compared with two better target detection algorithms that have been open-sourced and compiled based on C++.

The experiments are all done on the local 1660 Ti with Max-Q, and the comparison methods include YOLOv1, YOLOv2, YOLOv3, Tiny YOLOv3, YOLOv4, and YOLOv5. Among them, Tiny YOLOv3 is one of the typical representatives of the fast detection speed, while YOLOv4 is one of the typical representatives of the high detection accuracy. The experimental results of the comparison are shown in Table 6.

In this paper, the original YOLOv3 algorithm is improved, and the improved YOLOv3 algorithm is improved by 16.10% on the AP@0.5 (%) indicator compared with the traditional YOLOv3 algorithm. However, there is still a gap between the improved YOLOv3 algorithm and the YOLOv4 algorithm in this paper.

Table 6: Experimental results of different algorithms

	AP@0.5(%)	F1@0.5(%)	AP@0.75(%)	F1@0.75(%)
YOLOv1	0.5693	0.6236	0.6538	0.6883
YOLOv2	0.6231	0.6900	0.6121	0.6217
YOLOv3	0.7526	0.7123	0.5639	0.6231
Tiny YOLOv3	0.8263	0.7326	0.4721	0.4689
YOLOv4	0.9307	0.9011	0.7236	0.7521
YOLOv5	0.9128	0.8714	0.7187	0.7419
Ours	0.9136	0.8927	0.7124	0.7459

The computational complexity of each algorithm is shown in Table 7. Compared with YOLOv4, this paper's method has a certain disadvantage in detection accuracy, but it has a great advantage in indicators such as detection speed, computational complexity and model size.

Compared with the lightweight network Tiny YOLOv3, this paper's method occupies an absolute advantage in all indicators such as detection accuracy and detection speed. In summary, the method in this paper achieves a good balance between detection speed and detection accuracy, and has better comprehensive performance than other detection methods, which is suitable for deployment on low performance PCs.

Table 7: Computational complexity of each algorithm

	FPS (frame/second)	Computational complexity	Model size (MB))
YOLOv1	95.62	158.14	612.23
YOLOv2	72.85	112.40	417.09
YOLOv3	54.13	50.72	64.54
Tiny YOLOv3	55.91	5.695	35.71
YOLOv4	10.34	60.631	254.69
YOLOv5	7.52	82.314	326.01
Ours	129.17	3.547	8.13

The results of the comparison of the metrics of different algorithms are shown in Fig. 9, and the accuracy rate of the improved YOLOv3 algorithm has been further improved compared with the original YOLOv3 algorithm. In terms of training time, the training time of the improved YOLOv3 algorithm is 21h, which reduces a certain amount of workload compared with the original YOLOv3 algorithm.

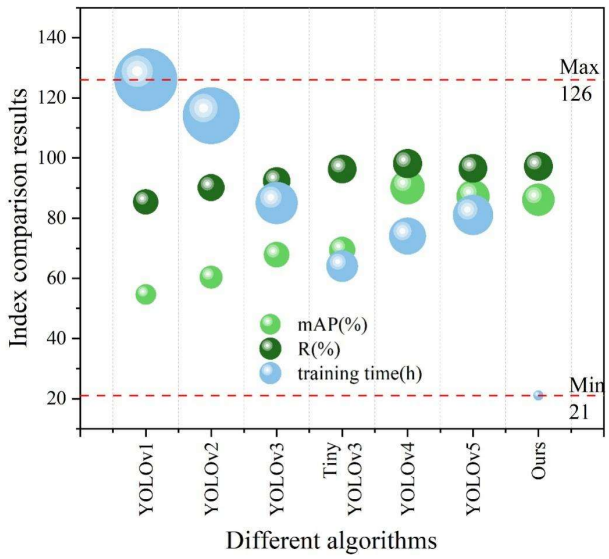


Figure 9: The comparison results of different algorithms

IV. Conclusion

In this paper, YOLOv3 algorithm is selected as the main analysis algorithm in this paper among deep neural networks, and the target detection accuracy is improved by bringing in the idea of feature pyramid network. And optimize the model training process to improve the performance of YOLOv3 algorithm's indexes and achieve visual defect detection improvement.

Fine-tuning is applied to pre-train the network of the improved YOLOv3 algorithm, and the loss function of the improved YOLOv3 algorithm is finally kept at 0.3508. The Image Size parameter selection in the performance experiments includes the image size training set of 64*64, 128*128, and 256*256, and observes the mAP curves of the three models, and finds that with image resolution gets higher and higher, the recognition accuracy also gets higher and higher. As the number of Anchors increases, the IOUs to improve the YOLOv3 algorithm also increase. The mAP of the improved YOLOv3 algorithm is the largest when the number of Anchors is 13. In the performance comparison of the algorithmic indexes of YOLO system, the improved YOLOv3 algorithm performs well in the indexes of detection accuracy, training time, and computational complexity, and thus it is more necessary to be improved.

References

[1] Fang, X., Luo, Q., Zhou, B., Li, C., & Tian, L. (2020). Research progress of automated visual surface defect detection for industrial metal planar materials. Sensors, 20(18), 5136.

[2] Chen, Y., Ding, Y., Zhao, F., Zhang, E., Wu, Z., & Shao, L. (2021). Surface defect detection methods for industrial products: A review. Applied Sciences, 11(16), 7657.

[3] Mezher, A. M., & Marble, A. E. (2024). Computer vision defect detection on unseen backgrounds for manufacturing inspection. Expert Systems with Applications, 243, 122749.

[4] Tulbure, A. A., Tulbure, A. A., & Dulf, E. H. (2022). A review on modern defect detection models using DCNNs–Deep convolutional neural networks. Journal of Advanced Research, 35, 33-48.

[5] Baygin, M., Karakose, M., Sarimaden, A., & Akin, E. (2017, September). Machine vision based defect detection approach using image processing. In 2017 international artificial intelligence and data processing symposium (IDAP) (pp. 1-5). Ieee.

[6] Tao, X., Zhang, D., Ma, W., Liu, X., & Xu, D. (2018). Automatic metallic surface defect detection and recognition with convolutional neural networks. Applied Sciences, 8(9), 1575.

[7] Ferguson, M., Ak, R., Lee, Y. T. T., & Law, K. H. (2018). Detection and segmentation of manufacturing defects with convolutional neural networks and transfer learning. Smart and sustainable manufacturing systems, 2(1), 137-164.

[8] Ho, C. C., Hernandez, M. A. B., Chen, Y. F., Lin, C. J., & Chen, C. S. (2022). Deep residual neural network-based defect detection on complex backgrounds. IEEE Transactions on Instrumentation and Measurement, 71, 1-10.

[9] Xing, J., & Jia, M. (2021). A convolutional neural network-based method for workpiece surface defect detection. Measurement, 176, 109185.

[10] Nagata, F., Tokuno, K., Watanabe, K., & Habib, M. K. (2018, October). Design application of deep convolutional neural network for vision-based defect inspection. In 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 1705-1710). IEEE.

[11] Chen, J., Liu, Z., Wang, H., Núñez, A., & Han, Z. (2017). Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network. IEEE Transactions on Instrumentation and Measurement, 67(2), 257-269.

[12] Adibhatla, V. A., Chih, H. C., Hsu, C. C., Cheng, J., Abbod, M. F., & Shieh, J. S. (2020). Defect detection in printed circuit boards using you-only-look-once convolutional neural networks. Electronics, 9(9), 1547.

[13] Xia, C., Pan, Z., Fei, Z., Zhang, S., & Li, H. (2020). Vision based defects detection for Keyhole TIG welding using deep learning with visual explanation. Journal of Manufacturing Processes, 56, 845-855.

[14] Boikov, A., Payor, V., Savelev, R., & Kolesnikov, A. (2021). Synthetic data generation for steel defect detection and classification using deep learning. Symmetry, 13(7), 1176.
[15] Saberironaghi, A., Ren, J., & El-Gindy, M. (2023). Defect detection methods for industrial products using deep learning techniques: A review. Algorithms, 16(2), 95.
[16] Djavadifar, A., Graham-Knight, J. B., Körber, M., Lasserre, P., & Najjaran, H. (2022). Automated visual detection of geometrical defects in composite manufacturing processes using deep convolutional neural networks. Journal of Intelligent Manufacturing, 33(8), 2257-2275.
[17] Ming, W., Cao, C., Zhang, G., Zhang, H., Zhang, F., Jiang, Z., & Yuan, J. (2021). Application of convolutional neural network in defect detection of 3C products. IEEE Access, 9, 135657-135674.
[18] Ouyang, W., Xu, B., Hou, J., & Yuan, X. (2019). Fabric defect detection using activation layer embedded convolutional neural network. IEEE Access, 7, 70130-70140.
[19] Zhang, D., Hao, X., Liang, L., Liu, W., & Qin, C. (2022). A novel deep convolutional neural network algorithm for surface defect detection. Journal of Computational Design and Engineering, 9(5), 1616-1632.
[20] Jing, J., Wang, Z., Rätsch, M., & Zhang, H. (2022). Mobile-Unet: An efficient convolutional neural network for fabric defect detection. Textile Research Journal, 92(1-2), 30-42.
[21] Luo, Q., Fang, X., Liu, L., Yang, C., & Sun, Y. (2020). Automated visual defect detection for flat steel surface: A survey. IEEE Transactions on Instrumentation and Measurement, 69(3), 626-644.
[22] Zhou, X., Wang, Y., Zhu, Q., Mao, J., Xiao, C., Lu, X., & Zhang, H. (2019). A surface defect detection framework for glass bottle bottom using visual attention model and wavelet transform. IEEE Transactions on Industrial Informatics, 16(4), 2189-2201.
[23] Huangpeng, Q., Zhang, H., Zeng, X., & Huang, W. (2018). Automatic visual defect detection using texture prior and low-rank representation. IEEE Access, 6, 37965-37976.
[24] Zhou, Y., Yuan, M., Zhang, J., Ding, G., & Qin, S. (2023). Review of vision-based defect detection research and its perspectives for printed circuit board. Journal of Manufacturing Systems, 70, 557-578.
[25] Qiu, K., Tian, L., & Wang, P. (2021). An effective framework of automated visual surface defect detection for metal parts. IEEE sensors journal, 21(18), 20412-20420.
[26] Li, C., Gao, G., Liu, Z., Yu, M., & Huang, D. (2018). Fabric defect detection based on biological vision modeling. Ieee Access, 6, 27659-27670.
[27] Czimmermann, T., Ciuti, G., Milazzo, M., Chiurazzi, M., Roccella, S., Oddo, C. M., & Dario, P. (2020). Visual-based defect detection and classification approaches for industrial applications—a survey. Sensors, 20(5), 1459.
[28] Ren, Z., Fang, F., Yan, N., & Wu, Y. (2022). State of the art in defect detection based on machine vision. International Journal of Precision Engineering and Manufacturing-Green Technology, 9(2), 661-691.
[29] Jha, S. B., & Babiceanu, R. F. (2023). Deep CNN-based visual defect detection: Survey of current literature. Computers in Industry, 148, 103911.
[30] Tang, H., Liang, S., Yao, D., & Qiao, Y. (2023). A visual defect detection for optics lens based on the YOLOv5-C3CA-SPPF network model. Optics express, 31(2), 2628-2643.
[31] Singh, S. A., & Desai, K. A. (2023). Automated surface defect detection framework using machine vision and convolutional neural networks. Journal of Intelligent Manufacturing, 34(4), 1995-2011.
[32] Wang, T., Chen, Y., Qiao, M., & Snoussi, H. (2018). A fast and robust convolutional neural network-based defect detection model in product quality control. The International Journal of Advanced Manufacturing Technology, 94, 3465-3471.
[33] Yuan, Z. C., Zhang, Z. T., Su, H., Zhang, L., Shen, F., & Zhang, F. (2018). Vision-based defect detection for mobile phone cover glass using deep neural networks. International Journal of Precision Engineering and Manufacturing, 19, 801-810.
[34] Westphal, E., & Seitz, H. (2021). A machine learning method for defect detection and visualization in selective laser sintering based on convolutional neural networks. Additive Manufacturing, 41, 101965.
[35] Mahsa Mikaeili,Hasan Şakir Bilge & İsa Kılıçaslan. (2024). Speckle noise reduction on aligned consecutive ultrasound frames via deep neural network. Measurement Science and Technology,35(6).
[36] Sushma B. & Aparna Pulikala. (2024). AAPFC-BUSnet: Hierarchical encoder–decoder based CNN with attention aggregation pyramid feature clustering for breast ultrasound image lesion segmentation. Biomedical Signal Processing and Control,91,105969-.
[37] Poorni R. & Madhavan P.. (2024). An implementation of intelligent YOLOv3-based anomaly detection model from crowded video scenarios with optimized ensemble pattern extraction. The Imaging Science Journal,72(8),1147-1168.
[38] Meng Rong,Zhao Zhilong,He Mengtian,Wang Zhiyuan & Duan Yanbo. (2024). Smart UAV Infrared Image Defect Detection using YOLOv3 and Gaussian Mixture Models. Tehnički vjesnik,31(6),1975-1986.