

Article

Identification Method for Cone Yarn Based on the Improved Faster R-CNN Model

Hangxing Zhao ¹ , Jingbin Li ^{1,2,*}, Jing Nie ^{1,2} , Jianbing Ge ^{1,2}, Shuo Yang ¹, Longhui Yu ¹, Yuhai Pu ¹ and Kang Wang ¹

- ¹ College of Mechanical and Electrical Engineering, Shihezi University, Shihezi 832000, China; 20192109058@stu.shzu.edu.cn (H.Z.); niejing19@shzu.edu.cn (J.N.); kuri2012@sohu.com (J.G.); 20202309010@shzu.edu.cn (S.Y.); yulonghui@stu.shzu.edu.cn (L.Y.); puyuhai@stu.shzu.edu.cn (Y.P.); 20202109015@stu.shzu.edu.cn (K.W.)
- ² Industrial Technology Research Institute of Xinjiang Production and Construction Corps, Shihezi 832000, China
- * Correspondence: lijingbin@shzu.edu.cn

Abstract: To solve the problems of high labor intensity, low efficiency, and frequent errors in the manual identification of cone yarn types, in this study five kinds of cone yarn were taken as the research objects, and an identification method for cone yarn based on the improved Faster R-CNN model was proposed. In total, 2750 images were collected of cone yarn samples in real of textile industry environments, then data enhancement was performed after marking the targets. The ResNet50 model with strong representation ability was used as the feature network to replace the VGG16 backbone network in the original Faster R-CNN model to extract the features of the cone yarn dataset. Training was performed with a stochastic gradient descent approach to obtain an optimally weighted file to predict the categories of cone yarn. Using the same training samples and environmental settings, we compared the method proposed in this paper with two mainstream target detection algorithms, YOLOv3 + DarkNet-53 and Faster R-CNN + VGG16. The results showed that the Faster R-CNN + ResNet50 algorithm had the highest mean average precision rate for the five types of cone yarn at 99.95%, as compared with the YOLOv3 + DarkNet-53 algorithm with a mean average precision rate that was 2.24% higher and the Faster R-CNN + VGG16 algorithm with a mean average precision that was 1.19% higher. Regarding cone yarn defects, shielding, and wear, the Faster R-CNN + ResNet50 algorithm can correctly identify these issues without misdetection occurring, with an average precision rate greater than 99.91%.

Keywords: cone yarn; species recognition; Faster R-CNN; feature network



Citation: Zhao, H.; Li, J.; Nie, J.; Ge, J.; Yang, S.; Yu, L.; Pu, Y.; Wang, K. Identification Method for Cone Yarn Based on the Improved Faster R-CNN Model. *Processes* **2022**, *10*, 634. <https://doi.org/10.3390/pr10040634>

Academic Editors: Pablo Chamoso and Wanquan Liu

Received: 10 February 2022

Accepted: 21 March 2022

Published: 24 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The classification of cone yarn plays a vital role in weaving, dyeing, and finishing processes [1]. In the production process, the types of cone yarn are distinguished according to the color and marker on the top of the bobbin. During production, the working environment is dark and there are many kinds of cone yarn. However, the colors on the tops of the bobbins are similar and the size of the markers is limited. At present, the identification of cone yarn types mainly depends on manual classification, which is labor-intensive, inefficiency, and prone to errors after long identification times. This results in the manufacture of defective fabrics in the textile production process. These issues restrict the development of packaging and logistics automation processes [2,3]. Therefore, it is of great significance to carry out research on possible identification methods for cone yarn to improve the identification efficiency and control the yarn quality.

At present, the common practice in domestic and foreign textile mills is to use the colors and markers on the tops of the bobbins as the main basis for distinguishing yarn types [4]. Zhang [5] performed color shift correction, filtering, and other preprocessing

processes on bobbin images to obtain color information. Then, the template matching method was used to identify the bobbin. Yang [6] also used traditional machine learning methods to perform preprocessing, such as HSV (hue, saturation, value) color conversion, filtering, and denoising on each image. Finally, the hue and saturation were used as features and a support vector machine (SVM) was used as the classifier for the final classification of the bobbins. The traditional target detection approach mainly starts from the image features obtained using machine vision methods, then image processing is performed based on the differences in color, shape, or texture of the detected objects. Objects can be recognized by manually extracting features [7,8]. Under factory conditions, due to the variety of cone yarns that are used and the production environment where the work takes place, the colors and shapes on the top markers on the bobbins are often incorrectly matched. Traditional target detection methods are mainly based on experience and are easily affected by subjective human factors [9]. The disadvantage of the traditional identification method is that it requires a lot of human participation across the whole process. Feature extraction also requires designers to have rich experience. Therefore, the traditional target detection methods have limited generalization ability and cannot be used to find a general feature model. In recent years, scholars have gradually turned their research focus for image recognition to target detection algorithms based on deep learning, focusing on local areas of images and specific object categories [10–13]. With the deepening of deep learning algorithm research, deep learning models based on convolutional neural networks (CNNs) have made significant progress in image detection. By using a convolutional neural network to extract image multi-scale feature information, the detection and recognition effects are remarkable. CNNs also have high recognition accuracy and strong anti-interference ability. They can extract targets from complex backgrounds and can solve the issues related to the weak generalizability of the recognized objects [14–16]. They have significant success in distinguishing similar targets and have excellent development potential [17]. Convolutional neural network models have achieved considerable success in areas such as image classification [18], object detection [19], pose estimation [20], image segmentation [21], and facial recognition [22–24]. They also have great expansion advantages, although at the same time they are restricted by resources and data acquisition conditions. On the other hand, their ability for accurate recognition and classification of targets with a small number of samples is gradually attracting researchers' attention [25–29]. However, the tradeoff between data quality and quantity is a complex problem that cannot be ignored [30–35].

Convolutional neural network models used for object detection can be divided into two categories. The first type is based on the use of a regression algorithms, which directly completes the candidate area generation and target recognition processes. Such models SSD (single-shot multi-box detector) [36,37] and YOLO (you only look once) [36–38], among others. They are also known as single-stage detection algorithms. The second category includes algorithms based on region proposals. These are used to generate candidate regions first and then to perform target classification on the candidate regions, and include R-CNN [39], Fast R-CNN [40], Faster R-CNN [41], and other models. They are also known as two-stage detection algorithms. Regression-based algorithms first need to generate regions of interest based on feature extraction. Each region is then classified to directly perform region generation and predicted target classification, which is an end-to-end process. The YOLO series is a single-stage detection algorithm, proposed by Redmon [38], which directly performs region generation and prediction target classification. During prediction, the feature map is divided into grids, which dramatically improves the detection speed. In view of the shortcomings of the YOLOv1 algorithm, such as the inaccurate positioning and low detection accuracy, the YOLOv2 algorithm was also proposed using the Darknet-19 model [42]. The YOLOv2 classification network is trained at high resolution to improve the accuracy in target recognition. However, each cell is only responsible for predicting one target, and the recognition effect for small targets and targets is not good. The YOLOv3 algorithm [43] uses the Darknet-53 network and introduces multi-scale fusion to improve the detection accuracy. This algorithm can be used for the recognition

of small targets, occluded targets, and similar backgrounds in complex environments. In the identification of the categories studied in this paper, the detection accuracy takes precedence over the average detection speed. Compared with two-stage detectors, single-stage detection algorithms are more suitable for package detection [44–46].

The R-CNN series of algorithms generate candidate regions and perform regression classification in two stages, which is why they are called two-stage detection algorithms. The R-CNN algorithm divides the input images into multiple candidate regions of different kinds. It utilizes CNN to extract feature vectors. The feature vector classification is achieved using SVM, which improves the efficiency of the target detection process. To overcome the slow training speed caused by the number of candidate regions and inconsistent sizes, many researchers have proposed improved methods to speed up the detection process. Girshick [40] proposed the Fast R-CNN method, combining the ideas of R-CNN and SPP (spatial pyramid pooling) [41]. Finally, an end-to-end object detection algorithm was implemented to find candidate regions. The Fast R-CNN method directly obtains candidate regions through neural networks, and the detection speed and accuracy are greatly improved compared to R-CNN. The Faster R-CNN method proposed by Ren [42] uses the region generation network (RPN) instead of the candidate box (anchor) extraction method used in R-CNN. An anchor mechanism was proposed to generate candidate boxes, which can simultaneously predict object boundaries and object scores at each location. Compared with Fast R-CNN, the training time is considerably shortened, while the detection accuracy is not reduced. Han [47] proposed a video detection method based on the improved Faster R-CNN model by adjusting VGG16 convolutional layers and using online case mining. By improving the deep learning network framework, the R-CNN series of algorithms have improved the speed and accuracy of the detection processes. The Faster R-CNN model based on regional features has a low recognition error rate and a low missed detection rate. It can directly output the recognized detection results and can be used for real-time detection [48,49]. Therefore, the Faster R-CNN model has certain advantages in the target detection of cone yarn types [50,51].

To solve the problem of cone yarn identification in industry environments, the application of deep learning method has been promoted for target detection. This paper takes different cone yarn samples under actual production conditions as the research objects and uses a self-established cone yarn image set combined with multi-model and multi-scale positioning methods to add candidate frames. Finally, an improved Faster R-CNN model for cone yarn detection is proposed. Aiming to overcome the problem of small targets being prone to false and missed detection errors, the original network algorithm of the Faster R-CNN model is improved. ResNet50 [52], having strong representation ability, is used as the feature network instead of the original VGG16 backbone network. We conduct research and build a deep-learning-based identification model for cone yarn types. We are able to reduce the amount of human participation required in the design process and the dependence on manual feature extraction. This model can not only reduce the errors associated with the manual selection of cone yarn features, but is also more practical. It can be applied to detection applications in different scenarios, and can effectively improve the identification speed for use with cone yarns. The recognition algorithm embedded into the device in this work has important reference significance for the development of intelligent cone yarn classification and grading devices.

2. Materials and Methods

2.1. Construction and Preprocessing of Cone Yarn Image Dataset

2.1.1. Cone Yarn Image Data Acquisition

In the setup, the image acquisition device is located above the conveying channel of the cone yarn, as shown in Figure 1. It consists of a shading box, light source, photoelectric sensor, and Basler industrial camera (Basler Vision Technology Co., Ltd., Beijing, China, model acA 1300-30gc, resolution = 1294 pixels × 964 pixels). The two cameras are located at a height of 75 cm above the conveying channel with a depression angle of 45°. The image

acquisition device is driven by the photoelectric sensor signal. When the cone yarn to be detected reaches the designated working position, the photoelectric sensor is blocked to send out a working signal. After the image acquisition device receives the signal, the light source group cooperates with the camera group to collect the cone yarn image. In order to improve the processing speed and robustness of the algorithm, the external natural light is isolated through the structural design of the system. An artificial light source is then installed for the acquisition device. By designing the light path and illumination timing control system, the illumination conditions of the system are in the state required by the algorithm. The exposure time of the camera is set to 3 ms, and the brightness of the LED light source is adjusted to match the camera to take pictures. This can ensure the quality of the collected images and reduce the complexity of the algorithm.

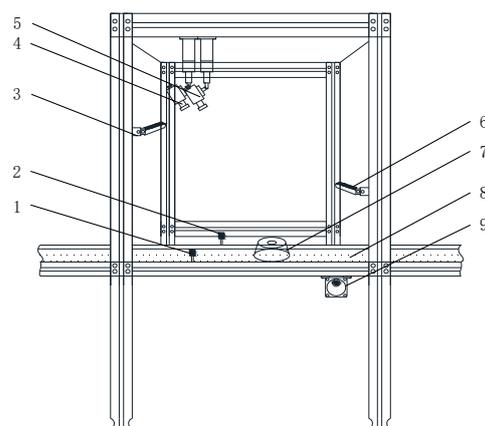


Figure 1. Image acquisition device: (1) photoelectric sensor transmitter; (2) photoelectric sensor receiver; (3) top light source in front of the cone yarn; (4) industrial camera on the top left of the cone yarn; (5) industrial camera on the top right of cone yarn; (6) upper light source behind the cone yarn; (7) cone yarn; (8) cone yarn conveying channel; (9) servo motor.

The research object of this paper is five types of cone yarn produced by Xinjiang Renhe Textile Technology Co., Ltd. (Huyanghe City, Xinjiang Province, China), namely a black twill bobbin, black fork bobbin, pure black bobbin, rose red bobbin, and green twill bobbin. The raw materials used in the company's yarn include cotton, Tencel, polyester, acrylic, and nylon, meaning the color of the yarn is white. In March 2021, the 5 types of cotton spinning cones in the production process were photographed in real time using an image capture device. There were 550 images in each category for a total of 2750 images. Some of the cone yarn samples are shown in Figure 2.

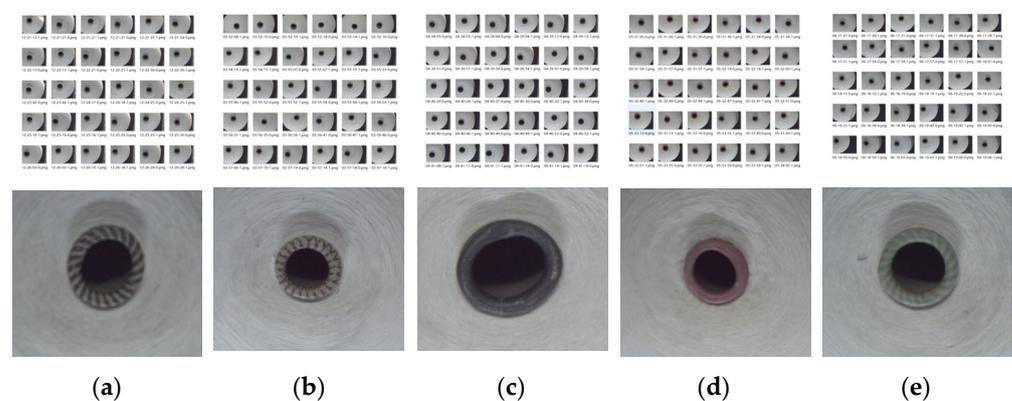


Figure 2. Cone yarn samples: (a) cone yarn with black twill bobbin; (b) cone yarn with black fork bobbin; (c) cone yarn with the pure black bobbin; (d) cone yarn with the rose red bobbin; (e) cone yarn with green twill bobbin.

2.1.2. Image Preprocessing

In order to speed up the training and convergence of the network model and to improve the accuracy, the image dataset for model training needs to be preprocessed. Convolutional neural network models require a large number of sample images to train with good recognition results [50]. The background for the images shown in this paper was relatively simple and had less noise. Therefore, we only performed data enhancement and label classification preprocessing, and no other processing was performed on it. Image preprocessing enhances the model generalization and adaptability to avoid weakening the training results of the model due to having too few sample images. Here, we use the Pytorch algorithm tool library to perform spatial scale transformation and normalization on the dataset. The image preprocessing process mainly includes the following aspects:

(1) Ensure that the training and test sets are mutually exclusive. First, the original images are randomly divided into different datasets. Ensure each dataset is mutually exclusive. Then, each dataset is augmented separately. After the expansion is completed, the final training set and test set are formed. Make sure the image is in the test set and that the augmented version of that image is definitely not in the training set;

(2) Ensure the uniformity of the dataset and improve the robustness of the model. Each image is expanded once by randomly rotating 45° , 90° , and 270° . Finally, the image of each type of cone yarn is guaranteed to reach 1100;

(3) In this paper, the image is proportionally scaled. The maximum side length after scaling is 800 pixels, and the minimum side length is 600 pixels. This ensures the speed and quality of model training;

(4) This article uses the mainstream LabelImg annotation tool. The cone yarn image is annotated according to the PASCAL VOC annotation format. Finally, a data format suitable for the PyTorch deep learning framework is generated;

(5) After preprocessing and expansion, the total number of final images is 5500.

2.2. Recognition Methods

2.2.1. Faster R-CNN Model Framework

Faster R-CNN has an improved network structure over R-CNN and Fast R-CNN. The main improvements in the network structures of R-CNN, Fast R-CNN, and Faster R-CNN are shown in Table 1.

Table 1. Comparison of the three network structures.

Networks	Extract Candidate Boxes	Feature Extraction	Feature Classification
R-CNN	selective search	CNN	SVM
Fast R-CNN	selective search	CNN + ROI pooling	CNN + ROI pooling
Faster R-CNN	RPN	CNN + ROI pooling	CNN + ROI pooling

The Faster R-CNN network model introduces region proposal networks (RPN) based on the Fast R-CNN model and replaces the selective search method in Fast R-CNN with the RPN network. Thus, end-to-end training is achieved through back propagation and stochastic gradient descent, which improves the accuracy and speed of detection. In this paper, Faster R-CNN is selected as the primary network model for cone yarn detection and identification, and its structure is shown in Figure 3.

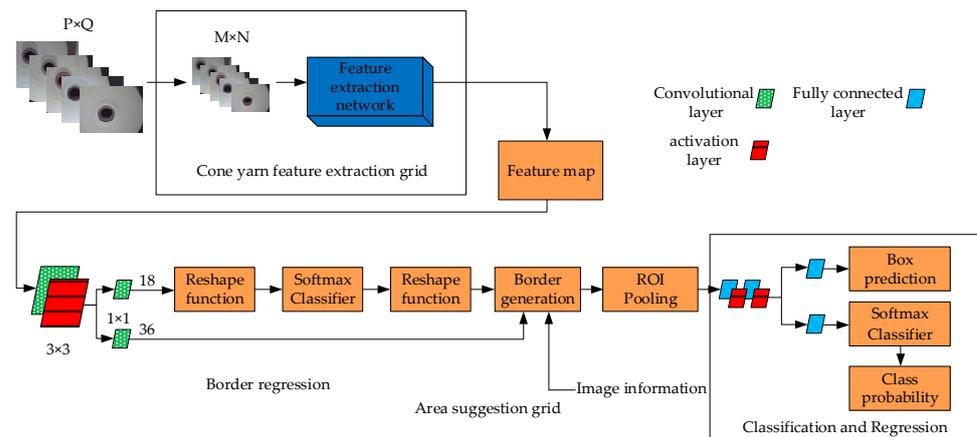


Figure 3. The framework for cone yarn detection based on Faster R-CNN. $P \times Q$ represents the original image size. $M \times N$ represents the model input image size.

The process of cone yarn detection and recognition is mainly divided into four parts, namely the cone yarn feature extraction, RPN network, ROI pooling interest region, and cone yarn multi-classification regression. The steps for cone yarn detection and recognition based on Faster R-CNN are as follows:

(1) Use the feature extraction network to extract the feature map of the cone yarn, which is shared by the subsequent RPN network and Fast R-CNN network;

(2) The RPN network uses a 3×3 sliding window to traverse the entire feature map. The Softmax classifier is mainly used to distinguish cone yarn and background information and to determine whether anchors belong to the foreground or background. The candidate frame position is obtained by anchor point regression. Bounding box regression is mainly used to adjust the four parameters of the proposal box (that is, the x-axis and y-axis coordinates of the center point of the proposal box and the width and height), and preliminary screening is performed on the obtained proposal box. In this way, the area containing the cone yarn is determined to the greatest extent, and the feature submap is obtained;

(3) The ROI pooling process obtains the feature map and the proposal frame simultaneously, and then uses the proposal frame to intercept the feature map. To adjust the acquired feature maps with different sizes to the size required by the classifier, the feature submaps of different sizes can be normalized. The feature submaps of the same size are sent to the subsequent fully connected layers for object classification and position adjustment;

(4) The classification and regression networks are mainly used to judge whether the intercepted feature map contains cone yarn information and to adjust the suggestion box to judge its category. At the same time, the bounding box is regressed to obtain the precise shape and position of the candidate box.

2.2.2. RPN Network

The RPN structure is shown in Figure 4. The region candidate frame is mainly generated by the end-to-end training method, enabling Faster R-CNN to achieve end-to-end object detection. Instead of the original sliding window method and SS (selective search) [53] algorithm, the generation speed of the detection frame is greatly improved. The RPN network is a fully convolutional network that performs convolution operations on the feature map through a sliding window. After the feature map is passed into the RPN, each position in the feature map corresponds to an n-dimensional feature vector. Each position corresponds to 9 possible candidate windows in the original image, and these candidate windows have three areas measuring 128^2 , 256^2 , and 512^2 , respectively. Each area size has three aspect ratios, which are 1:1, 1:2, and 2:1, respectively. These candidate windows are called anchors. We use a 3×3 sliding window to generate a feature vector of n-dimensional length and then pass this feature vector to the classification layer and

regression layer, respectively. In the classification layer, the Softmax classifier is used to judge the foreground and background of the anchor points, and the probability that the nine anchors at each position belong to the foreground and background are the outputs. In the regression layer, by adjusting the center coordinates and length and width of the anchor point border, we output the parameters for the window corresponding to the 9 anchors at each position, which should be translated and magnified. Then, we fit the position of the candidate frame. The loss function [42] generated during the RPN network training process consists of the classification layer loss function L_{cls} [42] and the regression layer loss function L_{reg} [40]. The training loss function of the model is as follows:

$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{1}$$

$$L_{cls}(p_i, p_i^*) = -\log(p_i p_i^* + (1 - p_i)(1 - p_i^*)) \tag{2}$$

$$L_{reg}(t_i, t_i^*) = \sum_{L_1} \text{smooth}_{L_1}(t_i - t_i^*) \tag{3}$$

$$\text{smooth}_{L_1}(t_i - t_i^*) = \begin{cases} 0.5(t_i - t_i^*)^2 (|t_i - t_i^*| < 1) \\ |t_i - t_i^*| - 0.5 (|t_i - t_i^*| \geq 1) \end{cases} \tag{4}$$

$$p_i^* = \begin{cases} 1 & \text{(Anchor points are positive samples)} \\ 0 & \text{(Anchor points are negative samples)} \end{cases} \tag{5}$$

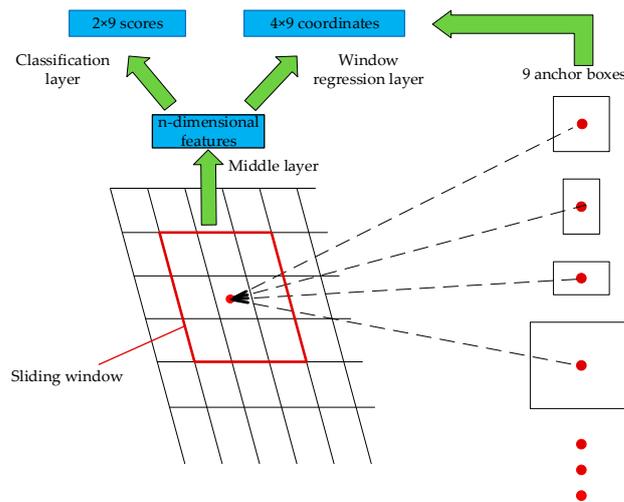


Figure 4. RPN structure.

In Formulas (1)–(5), L represents the loss of the RPN network; L_{cls} represents the loss of the classification layer; L_{reg} represents the loss of the regression layer; i represents the anchor index; t_i represents the predicted bounding box coordinate vector; t_i^* represents the true bounding box coordinate vector; N_{cls} represents the number of classification samples; N_{reg} represents the number of regression samples; p_i represents the predicted probability of the target; p_i^* represents the anchor point discriminant value; λ represents the weight parameter; L_1 represents the mean absolute error; smooth_{L_1} represents the robust L1 loss.

2.2.3. Backbone Feature Extraction Network

In the process of cone yarn feature detection, the Faster R-CNN network needs to select a trained network to increase the depth of the network and extract more abstract image features, in order to improve the detection ability of the model and obtain an ideal detection effect. However, as the number of network series increases, gradient vanishing and exploding problems also arise. A previous study [52] proposed a residual

network based on preserving the depth of the network, which makes the redundant layers in the network perform identity mapping, effectively solving the problem of gradient disappearance caused by the increase in the number of network layers. The residual network structural unit is shown in Figure 5.

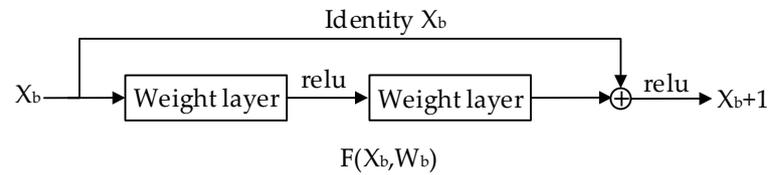


Figure 5. Residual unit.

The residual unit can be expressed as:

$$X_{b+1} = X_b + F(X_b, W_b) \quad (6)$$

For L of arbitrary depth:

$$X_L = X_b + \sum_{i=b}^{L-1} F(X_i, W_i) \quad (7)$$

Equation (7) has good back propagation, assuming that the loss is ε , which can be obtained according to the chain derivation rule:

$$\frac{\partial \varepsilon}{\partial X_b} = \frac{\partial \varepsilon}{\partial X_L} \frac{\partial X_L}{\partial X_b} = \frac{\partial \varepsilon}{\partial X_L} \left(1 + \frac{\partial}{\partial X_b} \sum_{i=b}^{L-1} F(X_i, W_i) \right) \quad (8)$$

In Formula (8) $F(x_i, w_i)$ is the residual, and when $F(x_i, w_i)$ is 0, it is the identity mapping process. Among them, $\frac{\partial}{\partial X_b} \sum_{i=b}^{L-1} F(x_i, w_i)$ cannot be negative in the process of derivation and continuous addition. Therefore, it is guaranteed that no gradient disappearance or gradient explosion will occur in the parameter update of this node.

The basic convolutional neural network can choose different structures, such as VGG-16 or ResNet [51], which have different performance and running times. Among them, ResNet won first place in the classification task of the ImageNet competition, and is widely used in detection, segmentation, recognition, and other fields due to its powerful performance. Common ResNet structures are ResNet50 and ResNet101, where 50 and 101 correspond to the numbers of layers in the network. The deeper the network layer, the slower the speed. Considering the speed and precision of cone yarn recognition, here we select ResNet50 as the basic convolutional neural network model of the Faster R-CNN detection model.

The ResNet50-based residual module includes two structures, the identity block and convolution block, as shown in Figure 6. The dimensions of the input and output vectors of the Identity Block module are the same, and the network can be directly deepened by concatenation to learn deep semantic information. The dimensions of the input and output vectors of the Conv Block module are different, and a 1×1 convolution process needs to be performed to match the dimensions. ResNet50 consists of one convolutional layer, one fully connected layer, and four sets of residual modules.

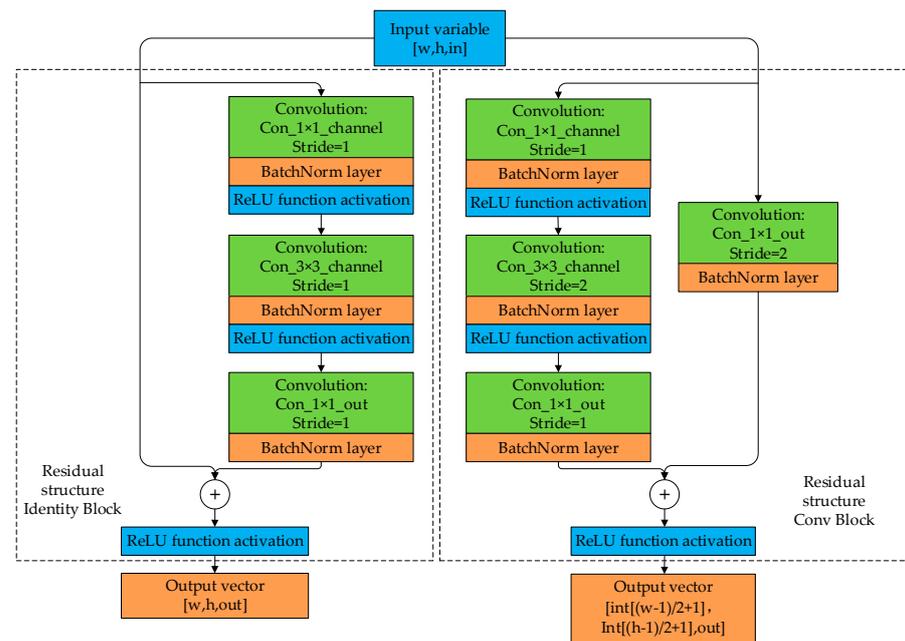


Figure 6. Identity block and Conv block residual structures.

Each group has 3, 4, 6, and 3 blocks, and each block has three convolutional layers. The structure is shown in Figure 7.

The size of the original input image is 1278 pixels \times 958 pixels, but the model network of ResNet50 will resize the input image to 600 pixels on the shortest side. Therefore, the inputs (800, 600, 3) in the structure all represent the cone yarn image with an input size of 800 pixels \times 600 pixels \times 3 channels. The output (400 \times 300 \times 64) on the left side of the model network structure represents the size of the output feature map. The right side represents the convolutional composition of the multi-layer neural network. For example, the size of the convolution kernel in “7 \times 7 conv, 64, /2” is 7 \times 7, the number of channels is 64, and the stride is two. Conv Block1 represents the first convolutional block1, containing one convolutional layer. Conv Block2-1, Conv Block3-1, Conv Block4-1, and Conv Block5-1 represent residual blocks with added scales, and the number of residual blocks is one. Conv Block2-x, Conv Block3-x, Conv Block4-x, and Conv Block5-x represent residual blocks that do not change the size, while the numbers of residual blocks are 2, 3, 5, and 2, respectively. Conv Block2 to Conv Block5 are defined as residual blocks 2, 3, 4, and 5, respectively. Each residual block consists of three convolutional layers; that is, a total of $3 \times (3 + 4 + 6 + 3) + 1 = 49$ convolutional layers.

In the neural network structure, Conv is a convolutional layer, BN is a batch normalization operation, Pool represents a maximum pooling layer, Avg Pool represents an average pooling layer, and Fc is a fully connected layer. The structure can be understood as five convolution blocks as a whole, including a 7 \times 7 convolution layer and four 3 \times 3 residual blocks, and the numbers of convolution kernels are 64, 64, 128, 256, and 512, respectively. For a 7 \times 7 convolutional layer, “/2” means that the moving step size of the convolution kernel is two, the data padding is one by default, and the convolutional layer does not add bias. BN [54] accelerates the network training and places the BN layer between the Conv and the activation function. For a 3 \times 3 pooling layer, the stride equals two and the data padding equals one.

The preprocessed image is input and processed through a series of residual blocks, and the output feature map size equals 25 \times 19 \times 2048. After the average pooling layer and flattening process, the output feature matrix becomes 1 \times 1 \times 2048. Then, this is input to the fully connected layer Fc, so the number of channels of the output feature map equals five. Finally, the corresponding probability values of the 5 kinds of cone yarn are output by the Softmax classifier, thereby giving the identification results.

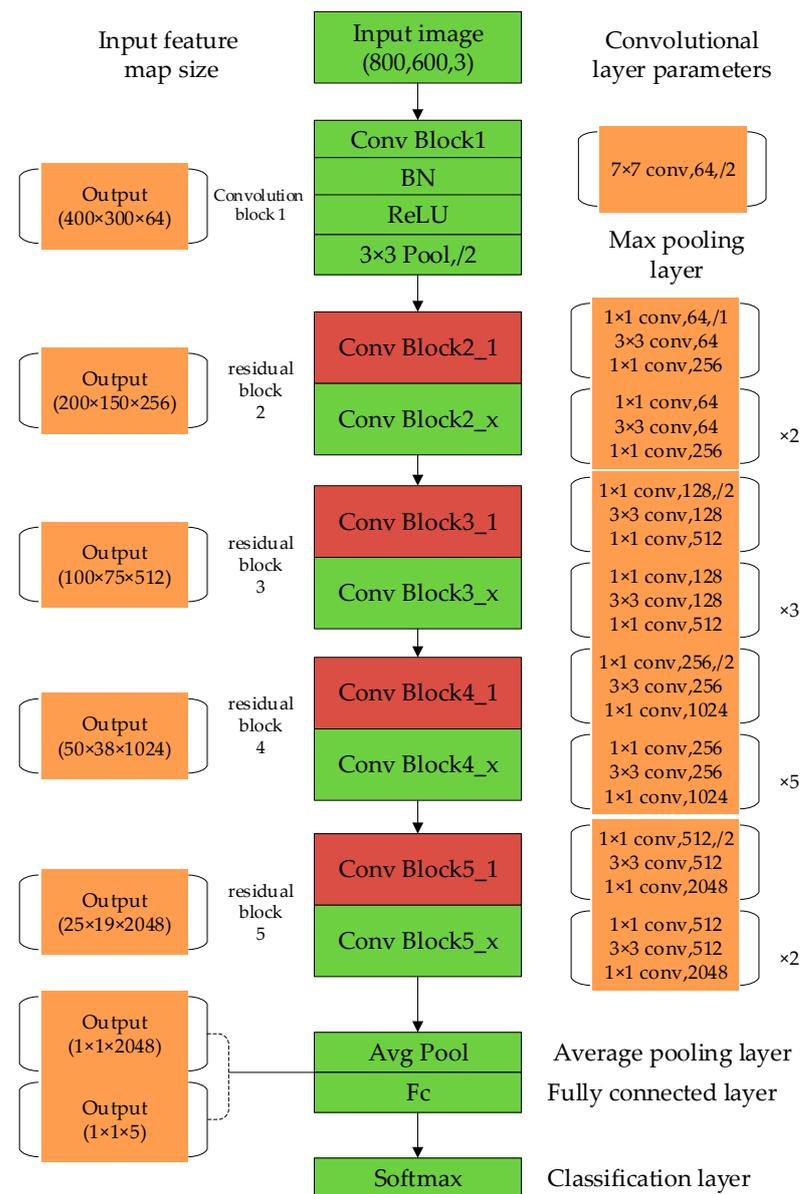


Figure 7. ResNet50 model network structure.

2.2.4. Dropout Optimization Algorithm

To avoid the overfitting phenomenon in the model, the dropout optimization algorithm is selected to improve the model’s performance [55,56]. The function of the dropout algorithm is to randomly discard the neuron output values during the network training process according to the set probability. “Inactivating” some neurons can avoid overfitting during network training. In this paper, in the fully connected layer Fc of the ResNet50 model network structure, adding dropout can avoid the risk of overfitting and reduce the classification and generalization error. Setting the dropout parameter to 0.5, the network will randomly deactivate half of the neurons during training. When the gradient is back-propagated, the disconnected neurons after deactivation will not participate in the update of parameters. Dropout forward propagation is shown in Figure 8, whereby the gradient is passed from the input neuron to the output neuron.

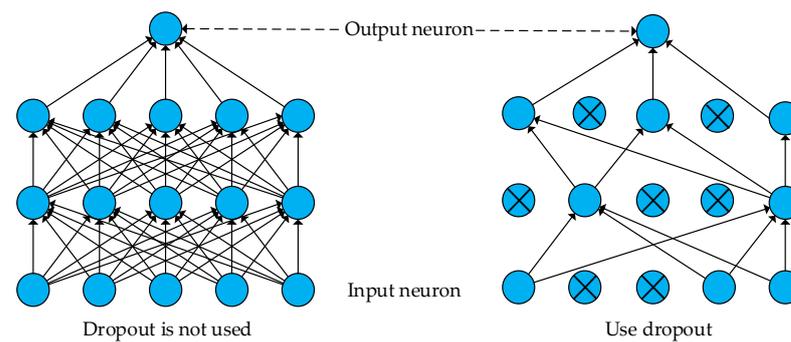


Figure 8. Neural network without dropout and neural network with dropout forward propagation.

3. Training and Evaluation of Species Recognition Models

3.1. Model Training Environment

The experimental environment for model training in this study is as follows: Windows 10 operating system (64 bit), Intel (R) Core (TM) i7-9700 K processor, 3.6 GHz main frequency, NVIDIA GeForce RTX 2080 graphics card, 32 GB memory, 512 GB solid-state drive, while using the CUDA platform to accelerate network training. The programming environment involves Python 3.8, Pytorch deep learning framework, Anaconda 3, and PyCharm.

3.2. Authentication Method

The dataset is further divided into a training set and test set, the proportions of which are 80% and 20%, respectively. The model is trained on the training set, and the model's performance is evaluated on the test set, which is called the common verification method of single verification. The single-shot verification method can obtain training results more quickly and can optimize hyperparameters by training the model once. Cross-validation is a method of model training and evaluation. After the dataset is divided, image samples are reused during training and testing. Multiple sets of disjointed training and test sets can be formed [57]. In this paper, the entire dataset is randomly divided into five subdatasets with no intersection and the same number of samples. One of the 5 datasets is selected as the test set and the other 4 as the training set. Five independent model training and testing procedures are then performed. Finally, the average of the test results of the five models is taken as the generalization error of this model. Using cross-validation provides a better approximation of the generalization error, giving an unbiased assessment of the predictive performance of the model. We use the same training samples and environmental settings to understand the impact of the experimental method on the detection results. The Faster R-CNN + ResNet50 method in this paper is compared with the two mainstream target detection algorithms of the YOLOv3 + DarkNet-53 algorithm and the Faster R-CNN + VGG16 algorithm.

3.3. Network Model Training Process

This paper uses Faster R-CNN as the basic network framework for cone yarn detection and recognition. The initialized weights are obtained using the original Faster R-CNN model trained on the PASCAL VOC 2007 dataset. The use of initial weights can speed up training and reduce training costs. Therefore, we use the "voc weights resnet.pth" pretrained model to initialize the weights. The alternate optimization training method is adopted and the number of images extracted from the training set each time is the batch size (batch size = 8). Training is performed under the ResNet50 feature extraction network, with the learning rate initially set to 0.01, confidence to 0.5, crossover ratio (IOU) to 0.3, and momentum factor to 0.9. The Adam optimization algorithm and cosine annealing are used to adjust the learning rate for each epoch so that it gradually decreases. The parameters are updated during gradient descent once per batch of samples, and the training epoch is set to 200. In this experiment, the training strategy for dropout is adopted, and the value of the dropout parameter is set to 0.5. The specific model training process is shown in Figure 9.

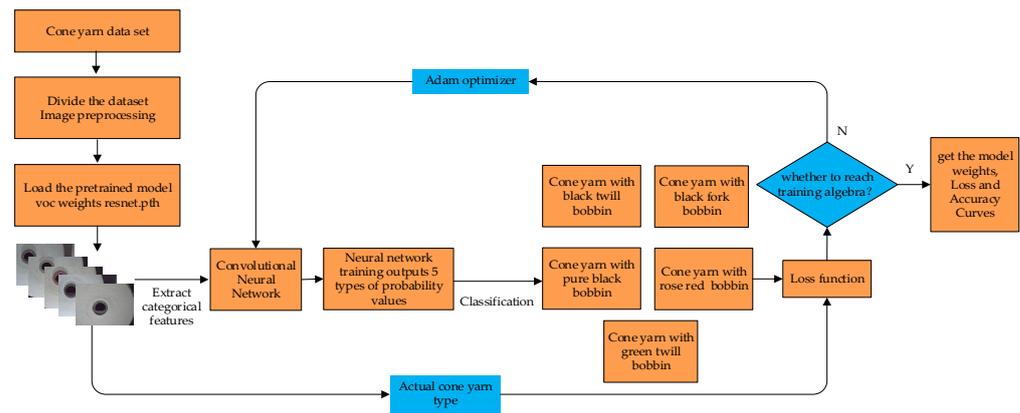


Figure 9. Model training process.

3.4. Evaluation Indicators

To measure the effectiveness of the experimental model for identifying the types of cone yarn, the precision (P_{re}), recall (R_{ec}), average precision (AP), and mean value of the average precision (mAP) were used as evaluation indicators. For each category, a curve can be drawn based on recall and precision. The AP value is the area under the curve. The AP value is used to measure the performance of a certain type of cone yarn. The higher the AP value, the better the model performance. The mAP is obtained by averaging the AP values of all categories. This is used to measure the comprehensive performance of the multi-objective model on five categories of cone yarn, calculated as follows:

$$P_{re} = \frac{T_P}{T_P + F_P} \times 100\% \quad (9)$$

$$R_{ec} = \frac{T_P}{T_P + F_N} \times 100\% \quad (10)$$

$$AP = \int_0^1 P_{re} R_{ec} dR_{ec} \quad (11)$$

$$mAP = \frac{\sum_1^n AP}{n} \quad (12)$$

In Formulas (9)–(10), P_{re} represents the precision, R_{ec} represents the recall, T_P represents the positive sample predicted as a positive sample, F_P represents the negative sample predicted as a positive sample, F_N represents the positive sample predicted as a negative sample, and N represents the number of categories.

The F1 score (F1) is a comprehensive evaluation index of precision and recall, which achieves a good balance between the two. The calculation formula is:

$$F1 = 2 \times \frac{P_{re} R_{ec}}{P_{re} + R_{ec}} \times 100\% \quad (13)$$

4. Results and Discussion

4.1. Training Data Analysis

4.1.1. Training for Algebraic Analysis

After the training is completed, the loss value corresponding to each training is read from the log file, and the curve of the loss value changing with the training algebra is drawn. As shown in Figure 10, the loss values of the three methods decrease with the increase in training algebra. The loss value drops quickly at the beginning, which is because of the large initial learning rate setting. As the training algebra increases, the learning rate gradually becomes smaller, while the change of the curve tends to be flat until it converges.

The loss value of the YOLOv3 + DarkNet-53 algorithm is higher than that of the Faster R-CNN + VGG16 method and the Faster R-CNN + ResNet50 method during the whole experiment. The reason is that the YOLOv3 + DarkNet-53 algorithm employs single-grid regression in the regression process, meaning the positioning of the bobbin is not accurate enough, resulting in low detection accuracy. The loss value of the Faster R-CNN + VGG16 method is not much different from that of the Faster R-CNN + ResNet50 method, but the loss value of the Faster R-CNN + ResNet50 method is always lower than that of the Faster R-CNN + VGG16 method.

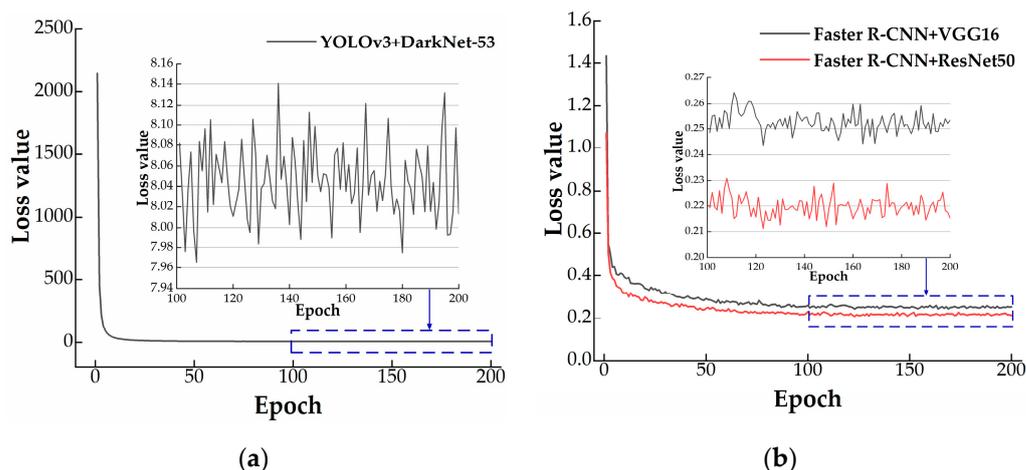


Figure 10. Comparison of loss values and training algebra curves: (a) YOLOv3 + DarkNet-53 algorithm comparison chart; (b) Faster R-CNN + VGG16 and Faster R-CNN + ResNet50.

4.1.2. Analysis of Evaluation Indicators

According to the log file, we can draw the change curve of the evaluation index with the training algebra during the training process, as shown in Figure 11. For generations 0–30 in the training process, the overall trend is for F1, R_{ec} , P_{re} , and mAP to increase. After the 30th generation, the general direction of the evaluation indicators is stable, only fluctuating within a small range. For the black twill bobbin, black fork bobbin, pure black bobbin, rose red bobbin, and green twill bobbin, the maximum F1 values of the Faster R-CNN + ResNet50 algorithm are 99.4%, 99.92%, 99.98%, 99.91%, and 99.95%, respectively; the mean values are higher than for the YOLOv3 + DarkNet-53 model and Faster R-CNN + VGG16 model by 9% and 6%, respectively. The F1, recall, and precision values of the Faster R-CNN + ResNet50 model for the five types of cone yarns are all greater than 99.88%, 99.90%, and 99.91%, respectively. Compared with the YOLOv3 + DarkNet-53 and Faster R-CNN + VGG16 algorithms, the Faster R-CNN + ResNet50 models all show ideal detection results. Here, we select F1 and mAP to evaluate the training results, and finally select the 200th generation as the final training network weight file. This file not only has F1 and mAP values, but also high recall and precision.

4.2. Analysis of the Different Methods used for the Identification of Cone Yarns

Here, 550 images in the test set are tested with the trained model, and the results are shown in Table 2. The precision of the Faster R-CNN + ResNet50 model for the five types of cone yarns is higher than the other two algorithms, while the mean average precision is 99.95%. The mean average precision is higher than the 97.71% for the YOLOv3 + DarkNet-53 model and 98.76% for the Faster R-CNN + VGG16 model, while the highest precision for the cone yarn with the green twill bobbin is 99.98%. The above results are due to the use of the residual structure of ResNet and the use of multi-scale features for the detection of the five types of cone yarns, which improve the classification precision. Secondly, the Faster R-CNN + ResNet50 algorithm has an average recall of 99.47%. In the more difficult-to-identify cone yarn with a black fork bobbin and the cone

yarn with a pure black bobbin, the recall is still high, remaining above 99%. Third, in terms of F1, the average value of the Faster R-CNN + ResNet50 algorithm is higher than that of the YOLOv3 + DarkNet-53 model and Faster R-CNN + VGG16 model by nine percentage points and six percentage points, respectively. This is because F1 is affected by the training quantity of the cone yarns. Under the same training set, the more training images contain cone yarns, the greater the F1 coefficient.

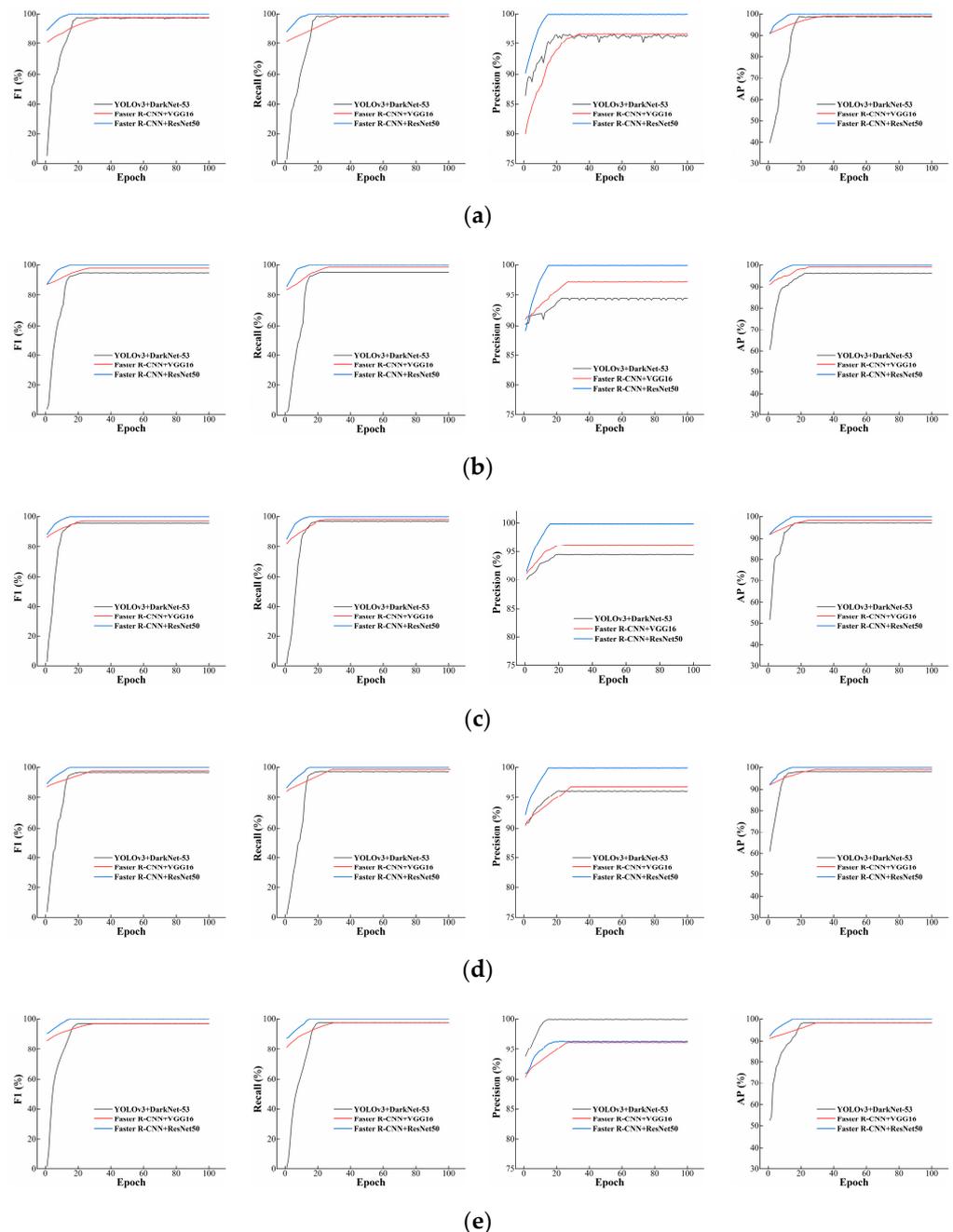


Figure 11. Identification of the training change curve for each index of the various types of cone yarns: (a) the training curve for each index of black twill bobbin yarn; (b) the training curve for each index of black fork bobbin yarn; (c) the training curve for each index of pure black bobbin yarn; (d) the training curve for each index of rose red bobbin yarn; (e) the training curve for each index of green twill bobbin yarn.

Table 2. Identification of cone yarns using different algorithms.

Type of Cone Yarn	Algorithm	P _{re} /%	R _{ec} /%	F1/%	AP/%
cone yarn with black twill bobbin	YOLOv3 + DarkNet-53	96.43	98.23	97.32	98.89
	Faster R-CNN + VGG16	96.68	98.54	97.60	99.01
	Faster R-CNN + ResNet50	99.93	99.96	99.94	99.97
cone yarn with black fork bobbin	YOLOv3 + DarkNet-53	94.96	95.23	94.84	96.17
	Faster R-CNN + VGG16	97.26	98.62	97.94	99.12
	Faster R-CNN + ResNet50	99.92	99.93	99.92	99.94
cone yarn with pure black bobbin	YOLOv3 + DarkNet-53	94.52	96.86	95.68	97.18
	Faster R-CNN + VGG16	96.26	98.01	97.13	98.36
	Faster R-CNN + ResNet50	99.87	99.90	98.88	99.91
cone yarn with rose red bobbin	YOLOv3 + DarkNet-53	96.18	97.26	96.72	98.08
	Faster R-CNN + VGG16	96.80	99.05	97.76	99.05
	Faster R-CNN + ResNet50	99.91	99.92	99.91	99.95
cone yarn with green twill bobbin	YOLOv3 + DarkNet-53	96.39	97.67	97.03	98.25
	Faster R-CNN + VGG16	96.21	97.63	96.91	98.28
	Faster R-CNN + ResNet50	99.94	99.97	99.95	99.98

Five representative images are selected from the 550 cone yarn test dataset, which are images of a normal cone yarn, cone yarn with yarn defects, cone yarn with a covered bobbin, dirty cone yarn, and cone yarn with a worn bobbin.

Here, we select five representative pictures from the cone yarn test dataset, which represent a normal cone yarn, cone yarn with yarn defects, cone yarn with a covered bobbin, dirty cone yarn, and cone yarn with a worn bobbin. As shown in Figures 12–16, YOLOv3 + DarkNet-53, Faster R-CNN + VGG16, and Faster R-CNN + ResNet50 algorithms are used for testing.

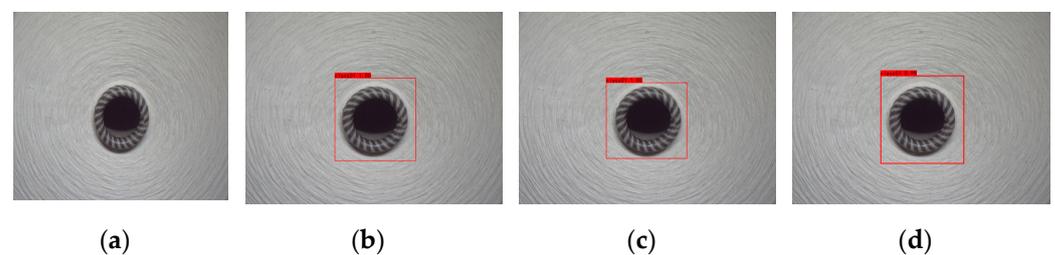


Figure 12. Identification of normal cone yarn with black twill bobbin: (a) original image; (b) YOLOv3 + DarkNet-53 algorithm recognition results; (c) Faster R-CNN + VGG16 algorithm recognition results; (d) Faster R-CNN + ResNet50 algorithm recognition results.

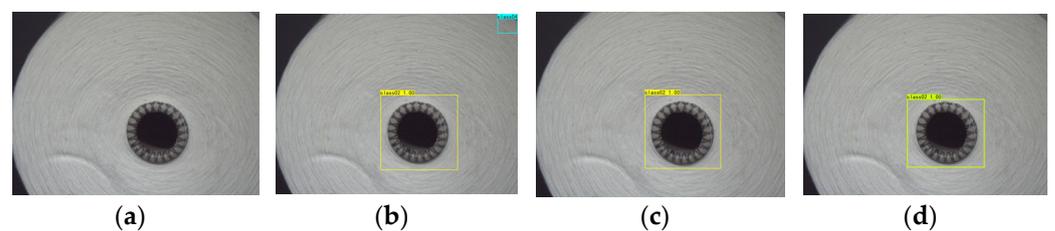


Figure 13. Identification of cone yarn with yarn defects: (a) original image; (b) YOLOv3 + DarkNet-53 algorithm recognition results; (c) Faster R-CNN + VGG16 algorithm recognition results; (d) Faster R-CNN + ResNet50 algorithm recognition results.

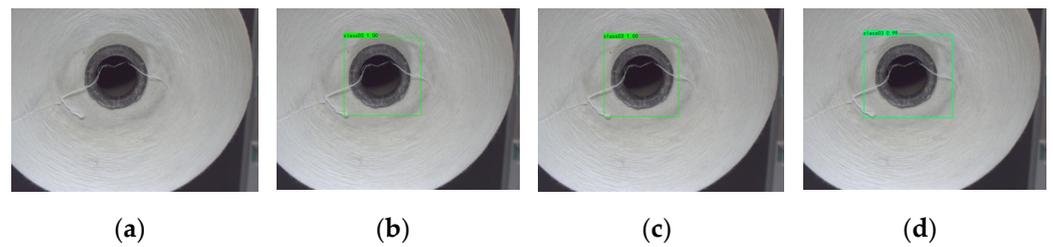


Figure 14. Identification of cone yarn with covered bobbin: (a) original image; (b) YOLOv3 + DarkNet-53 algorithm recognition results; (c) Faster R-CNN + VGG16 algorithm recognition results; (d) Faster R-CNN + ResNet50 algorithm recognition results.

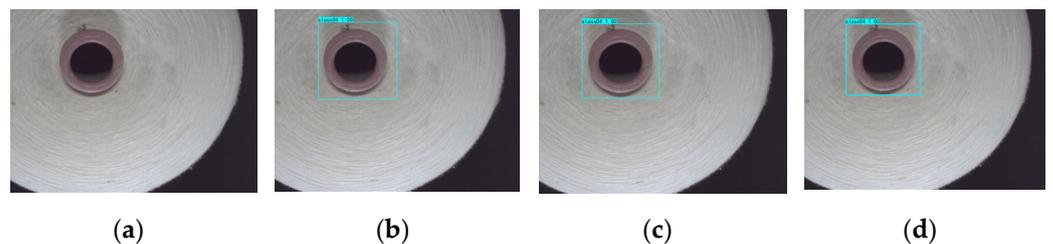


Figure 15. Identification of dirty cone yarn: (a) original image; (b) YOLOv3 + DarkNet-53 algorithm recognition results; (c) Faster R-CNN + VGG16 algorithm recognition results; (d) Faster R-CNN + ResNet50 algorithm recognition results.

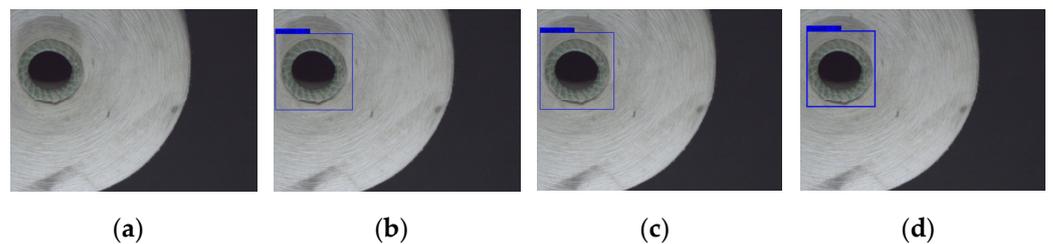


Figure 16. Identification of cone yarn with worn bobbin: (a) original image; (b) YOLOv3 + DarkNet-53 algorithm recognition results; (c) Faster R-CNN + VGG16 algorithm recognition results; (d) Faster R-CNN + ResNet50 algorithm recognition results.

As shown in Figure 12, for the normal cone yarn image, the average precision levels of the three algorithms are 98.89%, 99.01%, and 99.97%, respectively. The Faster R-CNN + ResNet50 algorithm has the highest average precision. It can accurately detect the black twill bobbin cone yarn, while the position of the bounding rectangle box is very accurate.

As shown in Figure 13, because the cone yarn has yarn defects, although the YOLOv3 + DarkNet-53 algorithm measures the cone yarn with a black fork bobbin, the average precision is only 96.17%. The YOLOv3 + DarkNet-53 algorithm still has errors in detection, and the yarn defects are mistakenly identified as cone yarn with a rose red bobbin. The average precision rate of the Faster R-CNN + VGG16 algorithm is 99.12%, while the average precision rate of the Faster R-CNN + ResNet50 algorithm is 99.94%, and there is no false detection.

As shown in Figure 14, for the target image with the covered bobbin, the average precision rates of the three algorithms for the cone yarn with pure black bobbin are 97.18%, 98.36%, and 99.91%, respectively. Faster R-CNN + ResNet50 still has high average precision.

As shown in Figure 15, for dirty cone yarn, the average precision rates of the three algorithms for detecting this type of cone yarn are 98.08%, 99.05%, and 99.95%, respectively. The Faster R-CNN + ResNet50 algorithm accurately detects the cone yarn with the rose red bobbin.

As shown in Figure 16, the average precision rates for YOLOv3 + DarkNet-53 and Faster R-CNN + VGG16 algorithms are 98.25% and 98.28%, respectively, for the cone yarn with a worn bobbin. The above two algorithms are both lower than the average precision of the Faster R-CNN algorithm by 1.7 percentage points.

For a very small number of bobbins with severe wear, there are instances of misclassification or no classification, as shown in Figure 17. Cone yarn is not identified, as shown in Figure 18. The Faster R-CNN + ResNet50 detection model detects very few severely worn bobbins. There are problems with misclassification and unrecognized types. However, it is an abnormal for a bobbin to be severely worn. This phenomenon is completely avoidable.

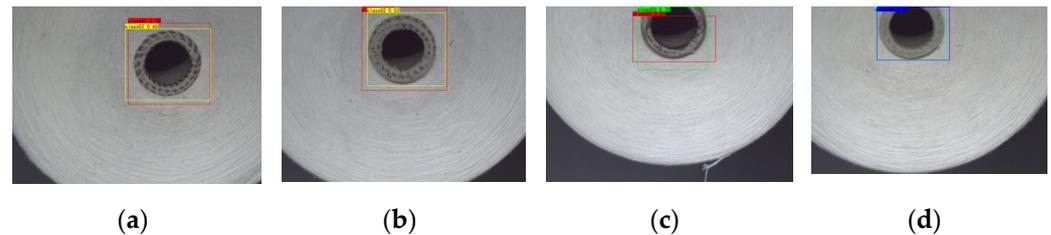


Figure 17. Incorrectly classified cone yarns: (a) cone yarn with black twill bobbin; (b) cone yarn with black fork bobbin; (c) cone yarn with pure black bobbin; (d) cone yarn with green twill bobbin.

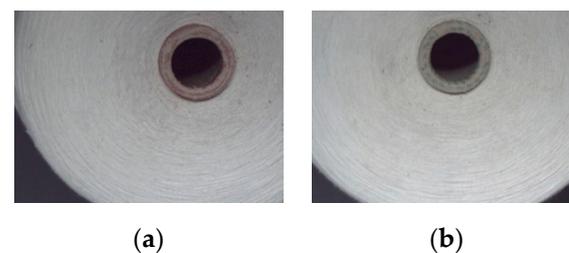


Figure 18. Unidentified cone yarn types: (a) cone yarn with the rose red bobbin; (b) cone yarn with green twill bobbin.

To sum up, the detection model based on Faster R-CNN + ResNet50 shows high accuracy for the five cone yarn test images, which reflects the strong generalization ability of the model. At the same time, the average precision rates for the cone yarn with yarn defects, the cone yarn with a covered bobbin, and the cone yarn with a worn bobbin are greater than 99.91%. This shows that the Faster R-CNN + ResNet50 algorithm has better robustness.

4.3. Detection Speed and Precision Rates of the Different Recognition Methods

As shown in Table 3, in terms of the recognition speed, the processing speed for YOLOv3 + DarkNet-53 reaches $45.73 \text{ image}\cdot\text{s}^{-1}$. The Faster R-CNN + VGG16 model shows decreased results compared to YOLOv3 + DarkNet-53, processing $14.26 \text{ image}\cdot\text{s}^{-1}$, which is higher than the detection speed for Faster R-CNN + ResNet50 with $4.79 \text{ image}\cdot\text{s}^{-1}$. However, in terms of the detection accuracy, the average precision rate of YOLOv3 + DarkNet-53 is the lowest among the three methods at 97.71%. The final average precision achieved by the Faster R-CNN + VGG16 method is 98.76%. Faster R-CNN + ResNet50 has an average precision rate of 99.95%, the highest among the three methods.

Table 3. Detection speed and precision rates of the different recognition methods.

Algorithm	Number of Images	Total Detection Time/ms	Average Detection Time/ms	Detection Speed $/(\text{image}\cdot\text{s}^{-1})$	mAP %
YOLOv3 + DarkNet-53	550	12028.37	21.87	45.73	97.71
Faster R-CNN + VGG16	550	38576.89	70.14	14.26	98.76
Faster R-CNN + ResNet50	550	114819.01	208.76	4.79	99.95

Therefore, the Faster RCNN + ResNet50 algorithm has more advantages than the YOLOv3 + DarkNet53 algorithm and the Faster R-CNN + VGG16 algorithm, and its detection precision is higher. Although the Faster RCNN + ResNet50 algorithm has the slowest detection speed, it can fully detect the targets in real time.

5. Conclusions

In this study, an image recognition method for cone yarn based on Faster R-CNN + ResNet50 was proposed. We selected five pictures of cone yarn samples, namely with a black twill bobbin, black fork bobbin, pure black bobbin, rose red bobbin, and green twill bobbin, in order to establish a database. Based on the convolutional neural network model, the recognition rates for the five kinds of cone yarn were analyzed. The YOLOv3 + DarkNet-53, Faster R-CNN + VGG16, and Faster R-CNN + ResNet50 algorithms were used for comparative experiments. The results showed that the Faster R-CNN + ResNet50 algorithm had the highest average precision rates for the five types of cone yarn, namely with a black twill bobbin, black fork bobbin, pure black bobbin, rose red bobbin, and green twill bobbin. For the above five types of cone yarn, the Faster R-CNN + ResNet50 algorithm AP gave precision rates of 99.97%, 99.94%, 99.91%, 99.95%, and 99.98%, with an mAP of 99.95%. Compared with the YOLOv3 + DarkNet-53 algorithm, the mAP for the Faster R-CNN + ResNet50 algorithm was 2.24% higher, while the single image detection time increased by 186.96 ms. Compared with the Faster R-CNN + VGG16 algorithm, the mAP for the Faster R-CNN + ResNet50 algorithm was 1.19% higher, while the single image detection time was reduced by 138.63 ms. Although the Faster R-CNN + ResNet50 algorithm had a low detection speed, it was fully capable of real-time detection of targets. Faster R-CNN + ResNet50 had the highest average detection precision, while the F1 coefficients were all higher than 99.91%. Faster R-CNN + ResNet50 showed good recognition precision and the best overall performance. For cone yarns with yarn defects, the Faster R-CNN + ResNet50 algorithm could correctly identify the target without missing and avoiding incorrect detection, with an average precision rate greater than 99.94%. The YOLOv3 + DarkNet-53 algorithm could detect cone yarn with yarn defects, but false detection occurred. Although Faster R-CNN + VGG16 detected cone yarn with a green twill bobbin, the average recognition precision was 1.70 percentage points lower than the Faster R-CNN + ResNet50 algorithm for the cone yarn with a worn bobbin.

The Faster R-CNN + ResNet50 identification algorithm designed in this paper had higher average accuracy and a faster detection speed. It accurately identified the different types of yarn with high average precision. A neural network was used to automatically extract deep-level abstract features to make up for the shortcomings of manual feature extraction. The recognition algorithm of the Faster R-CNN + ResNet50 model has practical significance for improving the identification efficiency of cone yarn types and controlling yarn quality.

The detection method proposed in this paper was based on the need to identify a large amount of actual sample data and manually label the targets according to the standard for model training. Under actual production conditions, the color and shape of the bobbin markers can be matched arbitrarily. Therefore, there are many types of bobbin markers. However, the type of new cone yarn can only be identified after the training of recognition model. In response to this problem, semantic labels will be added to the dataset later to enrich the sample data. In addition, it is still necessary to simplify the model network and reduce the weight, calculation costs, and resource consumption. However, it is possible to carry out embedded development research while ensuring high precision, and these recognition algorithms can be deployed in embedded devices. The image recognition information can be transmitted to the execution structure (manipulator or pneumatic device) in production applications.

Author Contributions: Conceptualization, H.Z. and J.L.; methodology, H.Z., J.N. and J.G.; software, H.Z. and L.Y.; validation, S.Y. and Y.P.; formal analysis, H.Z.; investigation, H.Z. and S.Y.; resources, H.Z.; data curation, H.Z.; writing—original draft preparation, H.Z.; writing—review and editing, H.Z., J.L. and J.N.; visualization, K.W.; supervision, H.Z.; project administration, H.Z.; funding acquisition, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Scientific and Technological Research Projects of Xinjiang Production and Construction Corps (No. 2019AB014).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

- Liu, X.; Liu, J. Monitoring system for yarn miscount based on radio frequency identification. *Shanghai Text. Technol.* **2021**, *49*, 20–24. [[CrossRef](#)]
- Guo, M.; Han, C.; Lu, Y.; Gao, W. Talking about the current situation of intelligent development of spinning process. *Cotton Text. Technol.* **2020**, *48*, 81–84.
- Chen, P.; Li, P.; Qin, M. Design of control system for cheese automatic packaging production line. *Packag. Eng.* **2021**, *42*, 282–286. [[CrossRef](#)]
- Ozkaya, Y.A.; Acar, M.; Jackson, M. Digital image processing and illumination techniques for yarn characterization. *J. Electron. Imaging* **2005**, *14*, 023001. [[CrossRef](#)]
- Zhang, F.; Zhang, T.S.; Ji, Y.L. Research on Color Sorting Algorithm of Spinning Tube Based on Machine Vision. *J. Xi'an Polytech. Univ.* **2018**, *32*, 560–566.
- Yang, L.Z.; Zhou, F.Y. Machine vision-based wool yarn clustering method. *Wool Text. J.* **2017**, *45*, 85–88.
- Jiang, H.H.; Wang, P.F.; Zhang, Z.; Mao, W.H.; Zhao, B.Q.P. Rapid identification of weeds in corn fields based on convolutional network and hash code. *J. Agric. Mach.* **2018**, *49*, 30–38.
- Wang, A.; Zhang, W.; Wei, X. A review on weed detection using ground-based machine vision and image processing techniques. *Comput. Electron. Agric.* **2019**, *158*, 226–240. [[CrossRef](#)]
- Fu, L.S.; Feng, Y.L.; Tola, E. Image recognition method of field multi-cluster kiwifruit based on convolutional neural network. *Chin. J. Agric. Eng.* **2018**, *34*, 205–211.
- Wang, L.; Zhang, H. Application of Faster R-CNN model in vehicle detection. *J. Comput. Appl.* **2018**, *38*, 666.
- Wan, S.; Goudos, S. Faster R-CNN for Multi-class Fruit Detection using a Robotic Vision System. *Comput. Netw.* **2019**, *168*, 107036. [[CrossRef](#)]
- Fu, L.; Majeed, Y.; Zhang, X.; Karkee, M.; Zhang, Q. Faster R-CNN-based apple detection in dense-foliage fruiting-wall trees using RGB and depth features for robotic harvesting—ScienceDirect. *Biosyst. Eng.* **2020**, *197*, 245–256. [[CrossRef](#)]
- Dai, X.; Hu, J.; Zhang, H.; Shitu, A.; Luo, C.; Osman, A.; Sfarra, S.; Duan, Y. Multi-Task Faster R-CNN for Nighttime Pedestrian Detection and Distance Estimation. *Infrared Phys. Technol.* **2021**, *115*, 103694. [[CrossRef](#)]
- Hu, Y.; Luo, D.Y.; Hua, K.; Lu, H.M.; Zhang, X.G. A review and discussion on deep learning. *J. Intell. Syst.* **2019**, *14*, 19.
- Li, Y.; Nie, J.; Chao, X. Do we really need deep CNN for plant diseases identification? *Comput. Electron. Agric.* **2020**, *178*, 105803. [[CrossRef](#)]
- Li, Y.; Chao, X. ANN-Based Continual Classification in Agriculture. *Agriculture* **2020**, *10*, 178. [[CrossRef](#)]
- Ba, G. Image Classification Algorithm Based on Convolutional Neural Network. *Comput. Inf. Technol.* **2020**, *28*, 3.
- Cao, X.; Yao, J.; Xu, Z.; Meng, D. Hyperspectral image classification with convolutional neural network and active learning. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4604–4616. [[CrossRef](#)]
- Jiang, Q.; Tan, D.; Li, Y.; Ji, S.; Cai, C.; Zheng, Q. Object detection and classification of metal polishing shaft surface defects based on convolutional neural network deep learning. *Appl. Sci.* **2020**, *10*, 87. [[CrossRef](#)]
- Kim, Y.; Kim, D. A CNN-based 3D human pose estimation based on projection of depth and ridge data. *Pattern Recognit.* **2020**, *106*, 107462. [[CrossRef](#)]
- Sultana, F.; Sufian, A.; Dutta, P. Evolution of image segmentation using deep convolutional neural network: A survey. *Knowl.-Based Syst.* **2020**, *201*, 106062. [[CrossRef](#)]
- Gao, G.; Yu, Y.; Yang, J.; Qi, G.J.; Yang, M. Hierarchical deep cnn feature set-based representation learning for robust cross-resolution face recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *10*, 87. [[CrossRef](#)]
- Khan, A.; Sohail, A.; Zahoora, U.; Qureshi, A.S. A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev.* **2020**, *53*, 5455–5516. [[CrossRef](#)]

24. Wu, L.; Liu, Z.; Bera, T.; Ding, H.; Langley, D.A.; Jenkins-Barnes, A.; Furlanello, C.; Maggio, V.; Tong, W.; Xu, J. A deep learning model to recognize food contaminating beetle species based on elytra fragments. *Comput. Electron. Agric.* **2019**, *166*, 105002. [CrossRef]
25. Nie, J.; Wang, N.; Li, J.; Wang, K.; Wang, H. Meta-learning prediction of physical and chemical properties of magnetized water and fertilizer based on LSTM. *Plant Methods* **2021**, *17*, 119. [CrossRef] [PubMed]
26. Li, Y.; Yang, J. Few-shot cotton pest recognition and terminal realization. *Comput. Electron. Agric.* **2020**, *169*, 105240. [CrossRef]
27. Li, Y.; Yang, J. Meta-learning baselines and database for few-shot classification in agriculture. *Comput. Electron. Agric.* **2021**, *182*, 106055. [CrossRef]
28. Li, Y.; Chao, X. Semi-supervised few-shot learning approach for plant diseases recognition. *Plant Methods* **2021**, *17*, 68. [CrossRef] [PubMed]
29. Yang, Y.; Zhang, Z.; Mao, W.; Li, Y.; Lv, C. Radar target recognition based on few-shot learning. *Multimed. Syst.* **2021**, 1–11. Available online: <https://link.springer.53yu.com/article/10.1007/s00530-021-00832-3> (accessed on 20 March 2022). [CrossRef]
30. Li, Y.; Yang, J.; Wen, J. Entropy-based redundancy analysis and information screening. *Digit. Commun. Netw.* **2021**. [CrossRef]
31. Li, Y.; Chao, X. Toward Sustainability: Trade-Off Between Data Quality and Quantity in Crop Pest Recognition. *Front. Plant Sci.* **2021**, *12*, 811241. [CrossRef] [PubMed]
32. Li, Y.; Chao, X. Distance-Entropy: An effective indicator for selecting informative data. *Front. Plant Sci.* **2022**, *12*, 3167. [CrossRef] [PubMed]
33. Li, Y.; Chao, X.; Ercisli, S. Disturbed-entropy: A simple data quality assessment approach. *ICT Express* **2022**. [CrossRef]
34. Nie, J.; Li, Y.; She, S.; Chao, X. Magnetic shielding analysis for arrayed Eddy current testing. *J. Magn.* **2019**, *24*, 328–332. [CrossRef]
35. Li, Y.; Sheng, X.; Lian, M.; Wang, Y. Influence of tilt angle on eddy current displacement measurement. *Nondestruct. Test. Eval.* **2016**, *31*, 289–302. [CrossRef]
36. Qu, J.; Su, C.; Zhang, Z.; Razi, A. Dilated convolution and feature fusion SSD network for small object detection in remote sensing images. *IEEE Access* **2020**, *8*, 82832–82843. [CrossRef]
37. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single Shot Multibox Detector. In *Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016*; Springer: Cham, Switzerland, 2016; pp. 21–37.
38. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *IEEE* **2016**, 779–788. Available online: https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html (accessed on 20 March 2022).
39. Li, J.H.; Lin, L.J.; Tian, K. Detection of leaf diseases of balsam pear in the field based on improved faster R-CNN. *Trans. Chin. Soc. Agricult. Eng.* **2020**, *36*, 179–185.
40. Girshick, R. Fast R-CNN. *Comput. Sci.* **2015**, 1440–1448. Available online: https://openaccess.thecvf.com/content_iccv_2015/html/Girshick_Fast_R-CNN_ICCV_2015_paper.html (accessed on 20 March 2022).
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef]
42. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Processing Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
43. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017*; pp. 6517–6525.
44. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
45. Luo, H.L.; Chen, H.K. A Review of Object Detection Based on Deep Learning. *J. Electron. Eng.* **2020**, *48*, 10.
46. Li, M.; Zhang, Z.; Lei, L.; Wang, X.; Guo, X. Agricultural greenhouses detection in high-resolution satellite images based on convolutional neural networks: Comparison of faster r-cnn, yolo v3 and ssd. *Sensors* **2020**, *20*, 4938. [CrossRef] [PubMed]
47. Han, C.; Gao, G.; Zhang, Y. Real-time small traffic sign detection with revised faster-RCNN. *Multimed. Tools Appl.* **2018**, *78*, 13263–13278. [CrossRef]
48. Hahn, S.; Choi, H. Understanding dropout as an optimization trick. *J. Neurocomputing* **2020**, *398*, 64–70. [CrossRef]
49. Sun, Z.; Zhang, C.L.; Ge, L.Z.; Zhang, M.; Li, W.; Tan, Y.Z.; College of Engineering, China Agricultural University. Image detection method of field broccoli seedlings based on Faster R-CNN. *J. Agric. Mach.* **2019**, *50*, 6.
50. Quan, L.; Feng, H.; Lv, Y.; Wang, Q.; Zhang, C.; Liu, J.; Yuan, Z. Maize seedling detection under different growth stages and complex field environments based on an improved Faster R-CNN. *Biosyst. Eng.* **2019**, *184*, 1–23. [CrossRef]
51. Liu, S.; Tian, G.; Xu, Y. A novel scene classification model combining ResNet based transfer learning and data augmentation with a filter. *Neurocomputing* **2019**, *338*, 191–206. [CrossRef]
52. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016*; pp. 770–778.
53. Uijlings, J.R.; Van De Sande, K.E.; Gevers, T.; Smeulders, A.W. Selective search for object recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171. [CrossRef]
54. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the International Conference on Machine Learning (PMLR), Lille, France, 6–11 July 2015*; pp. 448–456.

-
55. Wang, S.H.; Muhammad, K.; Hong, J.; Sangaiah, A.K.; Zhang, Y.D. Alcoholism identification via convolutional neural network based on parametric ReLU, dropout, and batch normalization. *Neural Comput. Appl.* **2020**, *32*, 665–680. [[CrossRef](#)]
 56. Garbin, C.; Zhu, X.; Marques, O. Dropout vs. batch normalization: An empirical study of their impact to deep learning. *Multimed. Tools Appl.* **2020**, *79*, 12777–12815. [[CrossRef](#)]
 57. Jung, Y. Multiple predicting K-fold cross-validation for model selection. *J. Nonparametric Stat.* **2018**, *30*, 197–215. [[CrossRef](#)]