UNIVERSITY

Master's Degree in COURSE

Master's Degree Thesis

Development of AI based applications to support insect breeding experiments for the circular economy

Supervisors

Prof. STEFANO DI CARLO

Prof. ALESSANDRO SAVINO

Candidate

JINZHUO CHEN

March 2022

Summary

The title of this thesis is Development of AI based applications to support insect breeding experiments for the circular economy and the Supervisors is Prof. STEFANO DI CARLO, Prof. ALESSANDRO SAVINO, and the Candidate is JINZHUO CHEN.

The topic of this thesis stems from the need for automatic counting of insects by biological laboratory personnel, which is a project in cooperation with the University of Turin. The purpose is to promote the development of biological sciences and realize a circular economy.

This thesis takes artificial intelligence-based recognition applications as the research goal and designs a set of intelligent larvae recognition applications based on flask and machine learning, which can realize the real-time operation of mobile devices. Moreover, the thesis introduced and compared different image processing methods. It proposed larvae image segmentation based on gray threshold and edge detection methods and an image recognition model based on transfer learning. Based on the VGG-16 model, a new fully-connected layer module is designed. The VGG-16 model was migrated to the model in the trained convolution layer of the ImageNet image data set. The collected image data set was divided into a training set, testing set, and validation set. Moreover, Data augmentation is applied to rotate and flip images to expand the data set.

The experiment compared new learning methods and transfer learning methods.

Training method	Learning rate	Batch_size	Accuracy(%)
New Training	0.01	8	25
	0.01	32	25
	0.0001	8	25
	0.0001	32	25
	0.00001	8	91.67
	0.00001	32	91.67
Transfer learning only trains the fully connected layer	0.01	8	45.3
	0.01	32	47.8
	0.0001	8	95.83
	0.0001	32	96.67
	0.00001	8	95.92
	0.00001	32	95.67
Transfer learning trains all layers	0.01	8	25
	0.01	32	25
	0.0001	8	93.5
	0.0001	32	95.83
	0.00001	8	100
	0.00001	32	100

Figure 1: Experimental results

Furthermore, transfer learning methods include only trains the full connection layer method and trains all the layers method(convolution layer + full connected layer). The experimental results showed that transfer learning trains model all layers can significantly improve the recognition ability of the model. The constructed insect recognition model can achieve 100% recognition accuracy of insect species, and the average counting accuracy of insect images can reach 91.5%. Moreover, Parameter learning rate and optimizer significantly influence model recognition rate. Compared with the new learning, the two training methods of transfer learning have significantly increased accuracy, and convergence is achieved in only about 20 epochs of training.

After selecting the best model, embed the model into the Flask framework to realize the function of the system.

This project supports researchers allowing them to reduce the work and improve accuracy and provides a reference for image processing to identify tiny insects.

Acknowledgements

ACKNOWLEDGMENTS

First of all, I would like to thank my professors for raising this fascinating topic, which came about to solve counting difficulties encountered by laboratory personnel. Because the total number of larvae to be counted in the laboratory is usually in the hundreds or thousands. Currently, counting is done manually, which would take a significant amount of time given the large number of larvae involved. This paper aims to use cutting-edge image processing algorithms combined with artificial intelligence to enable a mobile application that can precisely count the number of larvae in a picture, thereby supporting researchers to reduce work and improve accuracy.

Contents

Acronyms

AI
Artificial Intelligence

CNN
Convolutional Neural Networks

HTML5
Hyper Text Markup Language

CSS
Cascading Style Sheets

JS
JavaScript

JQuery
JavaScript and Query

AJAX
Asynchronous JavaScript And XML

OpenCV
Open Source Computer Vision Library

LR
Learning Rate

FC
Fully Connected

TL

Transfer learning

WSGI

Python Web Server Gateway Interface

Chapter 1

Introduction

1.1 Research background and significance of insect recognition

Insects are the most diverse, abundant, and widely distributed animals in the animal kingdom and are closely related to human life. Data show that more than 1 million species of insects have been found, and it accounts for as high as 80% of the entire animal circle. What is more, most agricultural pests are now insects. Insect classification is the foundation of insect research and pest control. Traditional insect identification is to classify and identify insects by taxonomic experts or technicians with knowledge of insect classification. However, the existing taxonomy experts and technicians who master classification are challenging to meet the needs of current practical scenarios in terms of distribution and quantity. With the rapid development of information technology, it has become possible to replace the human brain with artificial intelligence for recognition. An artificial intelligence-based automatic identification system is required for the research.[1]

Image-based insect recognition technology is widely used and can be applied to predict and prevent plant diseases and insect pests. According to statistics, plant diseases and insect pests are not found in time every year, which causes significant losses [2]. If an image-based insect recognition system is implemented, then Agricultural workers can use the system to identify insects to detect pests and diseases quickly. Image-based insect recognition technology can also help entomologists identify insect species faster and better to achieve timely control of plant diseases and insect pests[3]. In addition, Image-based insect recognition technology involves many fields and can be widely used in customs, plant quarantine departments, forest pest control departments, etc., and has a wide range of application prospects.

The application of image processing technology in insect identification is still in the initial application stage. Because image processing technology can extract

some previously complex features in insect images, such as area, perimeter, color, etc., therefore, it has become a significant question whether these techniques can be used to realize the mobile application recognition of insects by using the feature images extracted from insects as the classification basis, that is, to realize the automatic recognition of insects.

1.2 Current Status of Research on Insect Recognition

Abroad, New Mexico State University uses an artificial neural network to classify and identify insects in the cotton ecosystem. First, the corrosion and dilation in the morphological opening operation are exploited. The original insect image uses data augmentation followed by image segmentation, i.e., reducing the insect image to its most basic elements without losing essential and useful features.

Huang Xiaoyan and Guo Yong of China City University of Science and Technology discussed a color digital image segmentation algorithm based on mathematical morphology. They propose a method for filtering and then segmenting color digital images using morphological templates and applying it to the segmentation of stored color digital images of grain damage and worms.

Lian Feiyu proposed to use the wavelet transform to compress and extract image features of stored grain pests and use support vector machine technology to classify stored grain pests so that the classifier has good classification performance. He overcomes the traditional pattern recognition technology And improves the recognition rate. Yang Hongzhen et al. Insect images are extracted based on shape and color. A radial basis neural network classifier is used to recognize and classify insect images.

1.3 Environment and equipment for image acquisition

There are various environments for image acquisition, roughly divided into two types: laboratory and natural environments. Correspondingly, in the laboratory environment, insect image acquisition is mainly performed in the laboratory, and insect images are recorded with experimental instruments or mobile devices. The natural environment is primarily farmland, forest, and other environments for collection.

1.3.1 Laboratory environment

The study noted that the laboratory environment is the primary environment for future insect identification research. The laboratory environment, background, etc., are relatively simple conducive to image processing. In addition, laboratory personnel can specify the posture of insects in detail, which is convenient for subsequent algorithm design and system development. However, the laboratory environment requirements are relatively high, such as lighting equipment, constant temperature equipment, etc. Researchers are also laborious, such as: placing insect poses, assembling image acquisition equipment, etc., while insects in the laboratory environment are usually photographed outdoors. When it is brought to the laboratory for image acquisition, the acquisition process also has certain difficulties.

Most insect images taken in laboratory settings have shadows. To eliminate shadows, choose a place with a single background color to shoot. Eliminating shadows can reduce the influence of external factors on insect image quality, obtain more insect image features and improve image recognition accuracy.

1.3.2 Natural environment

There is no fixed image acquisition device in the natural environment. Experimenters can collect larvae pictures through mobile phones or digital devices, and the primary light source is natural light. However, the background of the natural environment is more complex, and the image processing task after the acquisition is cumbersome. In addition, the poses of the larvae are diverse when collecting images in the natural environment, which increases the difficulty of subsequent algorithm design and system development[4].

1.4 Thesis organization and structure

1. Chapter 1 introduces the research background, significance, and application field of insect identification are introduced, and the current research status of this field at home and abroad is summarized. The characteristics of image acquisition in different environments are outlined. On this basis, this thesis's research route and research content are given.

2. Chapter 2 introduces the theoretical knowledge used in machine learning and web application construction in detail, including image classification, image segmentation, data augmentation, flask framework, and other technologies.

3. Chapter 3 describes the experimental steps of image segmentation and classification in detail, trains, tests, evaluates the data, and selects the best

model.

4. Chapter 4 introduces the function of the whole system, the development steps of the method, and describes the development process in detail.

5. Chapter 5 summarizes the work of this article and gives future research directions.

Chapter 2

Overview of Key Technologies of Insect Recognition System

2.1 Image Segmentation Theory and Methods

Image segmentation is an important image processing technology that has received extensive attention in theoretical research and practical applications. There are many image segmentation methods and types, and some segmentation operations can be directly applied to any image. Some images can only be used in special segmentation categories. Some algorithms require rough segmentation of the image because they require information extracted from the image. The research object of this paper is insect larvae, and the primary purpose is to realize the division of images of insect larvae.

Image segmentation decomposes an image into specific parts (regions or objects) with similar properties. It uses these parts to analyze and describe the image. An image usually contains many different regions, such as objects, environments, backgrounds, etc. In the research and application of images, people are often only interested in certain parts of the image. These "interesting" parts are often referred to as the target or foreground (the other factors become the background). They generally correspond to specific images in the image[5]. So as to recognize and analyze the target, It is necessary to segment these relevant areas. On this basis, it is possible to utilize further the target, such as feature extraction and measurement. Image segmentation refers to the technology and process of image segmentation to divide an image into various feature regions and extract objects of interest. The features here can be grayscale, color, texture, etc., and the target can correspond

to a single area or multiple areas.

Commonly used segmentation methods include the gray threshold, region growing, edge detection, and so on.

2.1.1 Gray level threshold method

The basic idea of the gray level threshold method [6]is to calculate one or more gray thresholds based on the gray features of the image. And then compare the gray value of each pixel in the image with the threshold. Finally, sort the pixels into appropriate categories based on the comparison results. Therefore, the most critical step of this method is to solve the optimal gray threshold according to a specific criterion function.

The threshold method is especially suitable for images in which the target and the background occupy different gray-level ranges.

If the image has only two categories of target and background, then only one threshold needs to be selected for segmentation, and this method becomes a single-threshold segmentation. However, if there are multiple targets in the image to be extracted, a single-threshold segmentation will appear crops. In this case, multiple thresholds need to be selected to separate each target. This segmentation method correspondingly becomes multi-threshold segmentation.

Figure 2.1

The advantages and disadvantages of the gray level threshold method:
advantages: Simple calculation and high efficiency
disadvantages: It only considers the characteristics of the gray value of the pixel itself. Generally, it does not consider spatial characteristics, so it is more sensitive to noise and not robust. Threshold selection
The threshold segmentation method can be a classic method in image segmentation. It uses the difference in grayscale between the target to be extracted and the background in the image and divides the pixel level into several categories by setting a threshold to achieve the separation of the target and the background.

General process: Determine whether the pixel in the image belongs to the target area or the background area by judging whether the feature attribute of each pixel in the image meets the threshold requirements to convert a grayscale image into a binary image.

Expressed in mathematical expressions, the original image f(x,y), T can be set as the threshold, and the following formula is satisfied when the image is segmented:

$$g(x,y) = \begin{cases} 1, & f(x,y) \geq T \\ 0, & f(x,y) < T \end{cases}$$

The threshold segmentation method is simple to calculate. It can always use closed and connected boundaries to define non-overlapping regions, and images with a strong contrast between the target and the background can get a better segmentation effect.

The following are several optimal threshold selection methods:

(1)Use histogram:
This method uses the histogram to analyze and select a better threshold according to the relationship between the peaks and troughs of the histogram. This method is more accurate, but only for images with a target and a background, and the contrast between the two is obvious, and the histogram is the most valuable kind of bimodal.

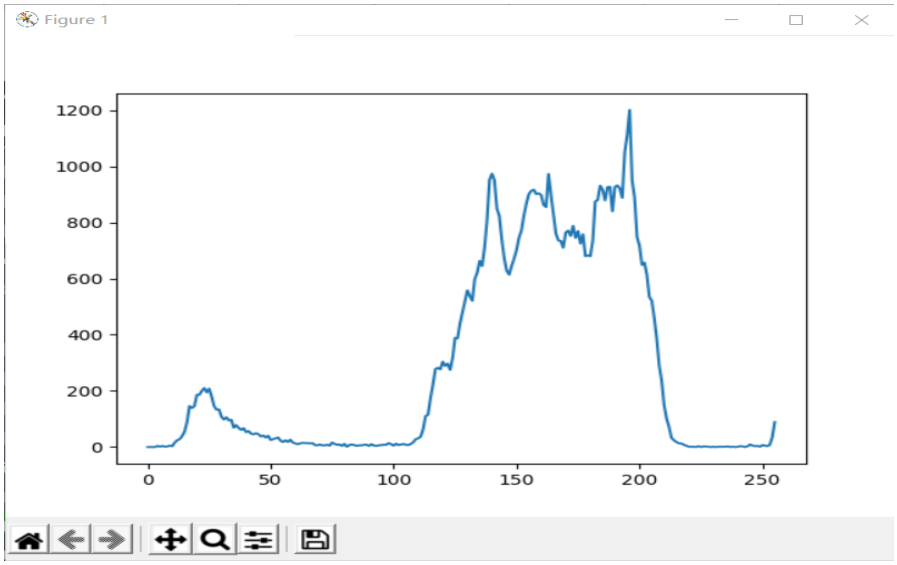

Figure 2.2: Histogram

(2)Maximum Between-Class Variance Method (OTSU):
OTSU is a method of automatically determining the threshold using the maximum variance between classes. It is a global binary algorithm, which divides the image into two parts, the foreground, and the background, according to the gray-scale characteristics of the image. When the optimal threshold is taken, the difference between the two parts should be the largest. The standard used in the OTSU algorithm to measure the difference is the more common maximum between-class variance.

Let T be the segmentation threshold between the foreground and the background, the proportion of the number of front sights in the image is w_0, and the average gray is u_0; the proportion of background points in the image is w1, the average gray is u_1, and the total average gray of the image is u, the foreground and background The variance g of the image has:

$$u=w_0\times u_0+w_1\times u_1$$

$$g=w_0\times(u_0-u)^2+w1\times(u_1-u)^2$$

Combine the above equations to get:

$$g=w_0\times w1\times(u_0-u)^2$$

or

$$g=w_0 \div (1-w_0) \times (u_0-u)^2$$

When the variance g is the largest, it can be considered that the difference between the foreground and the background is the largest, and the gray level T is the best threshold. The between-class variance method is susceptible to noise and the size of the target. It only produces a better segmentation effect for images with single-peak between-class variance. When the size ratio between the target and the background is very different (for example, affected by uneven illumination, reflection, or complex background), the between-class variance criterion function may show double peaks or multiple peaks, and the effect is not good at this time.

(3)Adaptive threshold method:

The above maximum between-class variance threshold segmentation method uses an equal threshold for each pixel on the image during the segmentation process. However, in actual situations, when the illumination is uneven, there is a sudden noise, or the background changes significantly, there will be no suitable single threshold for the whole image segmentation. If a single threshold is still used to process each pixel, the target may be changed. Moreover, the background area is wrongly divided. The idea of adaptive threshold segmentation is to set a different threshold for each pixel in the image.

A more straightforward adaptive threshold selection method is to determine a domain window centered on itself for each pixel, find the maximum and minimum values of the pixels in the window, take the average of the two as the threshold.The average value of all pixels within is used as the threshold.

2.1.2 Region growing method

The region growing method is a sequential segmentation method for extracting regions or entities from an image according to the gray level, the uniformity of the texture. The contrast and area, shape, size, and other criteria of the same background, the properties are roughly the same adjacent pixels of are grouped to form a segmented area.

Region growth starts from a set of seed pixels representing different growth areas, merges the eligible pixels in the seed pixel neighborhood into the growth area represented by the seed pixels, and continues with the newly added pixels as new seed pixels. In the merging process, until no new pixels that meet the conditions are found, the key to this method is to select a suitable initial seed pixel and a reasonable growth criterion.

Three problems that the region growing algorithm needs to solve:

(1) Select or determine a group of seed pixels that can correctly represent the

desired area;

(2) Determine the criteria for including adjacent pixels in the growth process;

(3) Specify the conditions or rules to stop the growth process.

Advantages and disadvantages of region growing algorithm:
(1) The effect of segmentation of complex images is good.

(2) The algorithm is complex and the amount of calculation is large.

2.1.3 Contour detection method

The contour detection method is used to obtain the contours of objects in the image. The basic idea is to detect the image first. The edge points in the image are then connected into contours according to a specific strategy to form a segmented area. Generally, a curve is used to simulate contour tracking or edge point connection techniques to find the object's boundary.

According to the image processing technology, edge detection technology can usually be divided into serial edge detection and parallel edge detection. Serial edge detection determines whether the current pixel is a point on the detection edge, which depends on the verification result of the previous pixel. Parallel edge detection is whether a pixel belongs to the noble point of the detected edge or not, depending on the pixel and neighboring pixels.

2.2 Introduction to OpenCV

The full name of OpenCV is: Open Source Computer Vision Library.[7] Founded by Intel in 1999, it is now powered by Willow Garage. and OpenCV implements many common algorithms in image processing and computer vision. There are many fields of application: Human-Computer Interaction, object recognition, Image segmentation, face recognition, Action recognition, motion tracking, robot, motion analysis, machine vision, Structural Analysis, Safe car driving, and so on.

Based on the above functional requirements, OpenCV also includes the following machine learning libraries: Boosting, decision tree, GBDT, EM algorithm (expectation-maximization), KNN, Naive Bayes classification, artificial neural network, random forest, support vector machine, OpenCV provides methods including image reading and saving, pixel manipulation, adding noise, RGB separation, and merging, histogram, morphological operations: dilation and erosion, filtering, Sobel edge detection operator, Canny operator, and so on.

OpenCV is an open-source computer vision and machine learning library that provides C++, C, Python, Java interfaces and supports Linux, Mac, Android, and Windows systems. OpenCV provides many functions to implement computer vision algorithms, from basic filtering to advanced object detection. Therefore, this experiment uses OpenCV in the Python environment to realize the identification and counting of larvae.

2.3 Image Recognition Theory

2.3.1 Image Recognition related technical introduction

Fundamental

The basic principle of image recognition technology based on artificial intelligence in practical application is to make full use of computer technology to process pictures and effectively extract valuable picture information. The principle and composition of remote image recognition technology itself are relatively simple[8]. It is embodied explicitly that after the completion of person-to-person vision, it can be regarded as image recognition technology. The obtained information can be systematically analyzed based on the primary impression. Since image recognition technology and the principle of computer processing data algorithms have consistent characteristics, computer technology can help extract simple image data information. However, when the amount of information is too large, the recognition effect of the technology will be affected. In identifying technical analysis, we should explore the best innovative way to improve image processing quality and efficiency.

Technical advantages

Artificial intelligence [9]image recognition technology has significant advantages in the aspect of convenience and intelligence. The most prominent advantage of image recognition technology is its application quality in the development of science and technology. From the perspective of intelligence, the difference between artificial intelligence image recognition technology and traditional image processing technology is obvious, especially in image processing. In particular, the function of face unlocking and intelligent recognition of image processing is similar[10]. After completing face unlocking, you can use this method as the primary unlocking method. Energy can also analyze and save itself. On this basis, [11]it is easy to conduct analysis based on image recognition, along with the rational use of image recognition technology, so that people can obtain high-quality services in life and work. Although people may not process complex images, they can still achieve their goals. Face-sweeping, unlocking, and card punching have improved people's

lifestyles. Owing to the rapid development of society, the popular features of image recognition technology have gradually become prominent.

2.3.2 Convolution Neural Networks introduction

In deep learning, the convolutional neural networ[12] is a pipelined multi-processing layer network model, which includes multiple convolutional layers, pooling layers, and fully connected layers. The model's training is to reverse the error-driven by the loss function. They are propagating to each layer of the network and updating the parameter weights. The test of the model is that the original data is mapped to the output through all the trained intermediate layers. The function of the convolutional layer is to extract the features of the image, and each layer contains many convolutional layers. The kernel, the convolution kernel, and the image function to obtain the local information of the image, and the calculation formula of the convolution layer is

$$z_{i,j}^{l} = \sum_{m=0}^{H} \sum_{n=0}^{W} f_{m,n} x_{i+m,j+n}^{l}$$

$$x_{i+m,j+n}^{l} - input\ tensor\ fo\ convolutional\ layer\ l$$

$$f_{m,n} - Convolution\ kernel\ of\ dimension\ H \times W$$

$$z_{i,j}^{l} - The\ output\ tensor\ of\ the\ convolutional\ layer\ l$$

$$i, j - thecoordinatesofthetensor$$

$$m, n - the\ coordinate\ value\ of\ the\ convolution\ kernel$$

The pooling layer uses the pooling function to compress and reduce the dimension of the feature map and has translation invariance to the input. The commonly used pooling functions include draw pooling, maximum pooling, and random pooling. If the maximum pooling function and2 $\times$ 2 pooling window, the calculation formula of the pooling layer is

$$f_{pool} = Max(s_{i,j}, s_{i+1}, s_{i,j+1}, s_{i+1,j+1})$$

$$\mathrm{f}_{pool} - The\ result\ after\ pooling$$
$$\mathrm{s}_{i,j} - element\ at\ position\ (i,j)\ on\ the\ feature\ map\ tensor$$

The fully connected layer[13] is to reduce the dimension and tile the high-dimensional feature data extracted by the convolution layer and the pooling layer and then perform the nonlinear transformation, and finally, input the result into the classifier for classification.

VGG16 model

VGG16 is a kind of CNN model. vgg16 has 16 layers, 13 convolution layers, and three fully-connected layers. One pooling is used after the first two convolution kernels of 64 convolution kernels. The second one is After two convolutions with 128 convolution kernels, pooling is used, and after three convolutions with 512 convolution kernels are repeated twice, pooling is performed again. Finally, three full connections are performed. Attached is the official vgg16 network structure diagram:

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 2.3: VGG16 architecture

2.3.3 Transfer Learning introduction

With the emergence of more and more machine learning application scenarios, the existing well-performing supervised learning requires a large amount of labeled data. Labeling data is a boring and expensive task, so transfer learning is receiving more and more attention. Combined with the size of our dataset, transfer learning is the best way to achieve the classification of this topic[14].

Transfer learning is to make a convolutional neural network model trained on a task suitable for a new task by simple adjustment[15]. The convolutional layer of the trained convolutional neural network can extract image features and the extracted feature vector is then input into the structure. A simple, fully connected layer can

achieve better recognition and classification, so the feature vector extracted by the convolution layer can be used as a more compact and expressive vector of the image. The trained convolution layer plus the fully connected layer suitable for the new task will form a new network model.

With little training, the new network model can handle new classification and recognition tasks.[16] Transfer learning first keeps the structure of the convolutional layer of the model unchanged, then loads the trained weights and parameters into the convolutional layer, then designs a fully connected layer suitable for the new task, and replaces the original fully connected layer with the newly designed fully connected layer. Connect the layer, form a new convolutional network model with the previous convolutional layer, and train the new model with the new image dataset. There are two ways to train the new model. One way is to freeze the convolutional layer and train only the fully connected layer, and another way is to train all layers of the network.

2.3.4 Data Augmentation introduction

Data augmentation is mainly used to prevent overfitting and is adopted when the dataset is small[17]. Although a two-layer network can theoretically fit all distributions, it is not easy to learn. Therefore, in practice, we usually increase the depth and breadth of the neural network, thereby enhancing its learning ability and facilitating the fitting of the distribution of training data. In convolutional neural networks, it has been experimentally proved that depth is more important than breadth.

However, with the deepening of the neural network, the parameters that need to be learned also increase, easily leading to overfitting. When the data set is small, too many parameters will fit all its characteristics. Overfitting is the distribution of a neural network so that it highly fits the training data[18]. However, it has a low accuracy rate for the test data and lacks generalization ability.

Therefore, in this case, data augmentation helps prevent overfitting.

Common data augmentation methods
1) Random rotation.
Random rotation is generally a random rotation of the input image.
2) Random cropping
Random cropping involves randomly cutting a part of the input image.
3) Gaussian noise.
Gaussian noise refers to randomly adding a small amount of noise to the image, which is effective in preventing overfitting and makes the neural network unable to fit all the features of the input image.

2.3.5 Model evalution schemes

There are different ways to train and test machine learning models. Although these ways can provide a performance score as an output, they do not generalize the results either in the training phase or the testing phase[19]. Nevertheless, it may depend on the specific data platform used to obtain that result. This section introduces and evaluates standard methods for ML models in increasing order of model complexity and model generalization ability.

Model evaluation with held-out test set

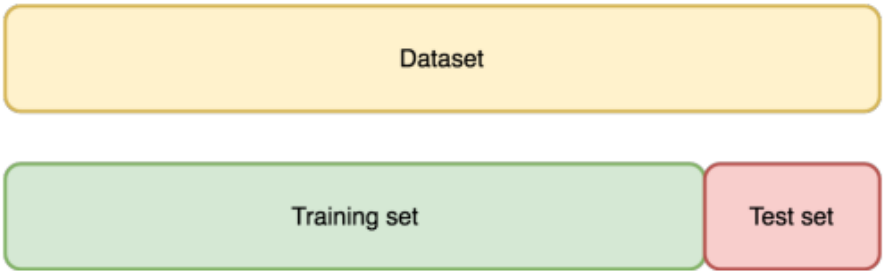

Figure 2.4: Model evaluation with held-out test set schema

This method splits the dataset into training and test sets. The first part is used to train models. The second part estimates the trained model's performance (such as accuracy) and generalization error. This approach has a severe risk of overfitting the reserved test set since hyper-parameters can be tuned until the estimator performs optimally. Furthermore, this method has the disadvantage of the high generalization error and high variance on the test set since model performance depends on the specific data that make up the training and test splits. So, do not take this approach.

Model evaluation with held-out validation and test sets

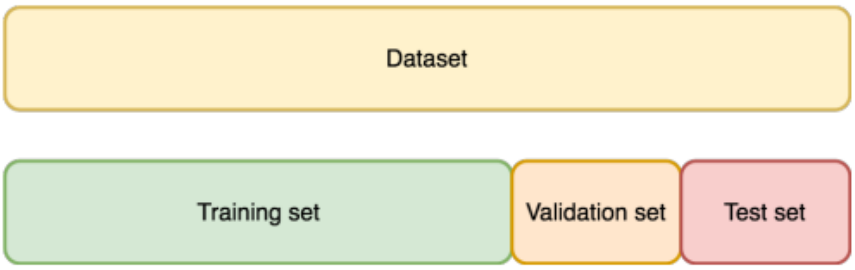

Figure 2.5: Model evaluation with held-out validation and test sets schema

Owing to the problem of overfitting, another part of the dataset is divided into the validation set. Training is performed on the training set based on the hyper-parameter tuning results obtained from the validation set. When the model appears to be tuned correctly, it is trained on the union of the training and validation sets using the best hyper-parameters. Finally, model evaluation is performed on the test set.

2.4 Flask related technical theory

2.4.1 Flask framework

Flask is a lightweight web application framework written in Python. It is based on Werkzeug WSGI (Python Web Server Gateway Interface)and is an interface between a Python application or framework and a Web server, which is widely accepted.

This thesis uses the Flask framework[20],which is Similar to frameworks such as Django, Tornado, etc. Flask was chosen for development because the following reasons:

1. The main development language is Python in the follow-up research on insect recognition algorithms based on machine learning and artificial intelligence. The whole system adopts a unified development language, and it is easy to develop and maintain.

2. The industry recognizes flask because of its flexible, lightweight, and efficient characteristics. It also has some open source libraries based on Werkzeug and Jinja2, as well as built-in servers and unit tests, adapts to RESTful, supports secure cookies, has complete official documents, and is easy to learn and master[9].

3.Flask has a flexible Jinja2 template engine, which improves the reuse rate of the front-end code. Moreover, development efficiency facilitates later development and maintenance.

Flask has two main dependencies: routing, debugging, and the Web Server Gateway Interface (WSGI) subsystem provided by Werkzeug; Jinja2 [21]provides the board system. Both Werkzeug and Jinja2 were developed by the core developers of Flask [22].

The Flask framework does not support various functions such as database access, validation of web forms, and user authentication. These functions are implemented as extension components and integrated with the Flask framework. Developers can create corresponding extensions according to the needs of a project. This is the exact opposite of large frameworks, which make most of the decisions themselves, making it challenging to change scenarios flexibly.

All Flask programs require Flask objects. The web server uses the WSGI protocol to forward all requests received from the client to this instance object, and the program instance is the object of the Flask class. The creation code is as follows:

```
1 from flask import Flask
2 app = Flask(__name__)
```

The constructor of the Flask class usually has one parameter, which needs to be specified manually, that is, the name of the program's main module or package. The parameter is set to the "___name___" variable in most cases.

The web browser sends the request to the web server, which immediately sends the request to the Flask object, but the Flask object needs to know the function or code corresponding to each request. In this case, routing is needed to save the URL request mapping of Python functions or codes. In Flask, the app. route decorator is provided to register the decorated function as a route. The decorator declares the route as follows:

```
@app. route(    /    )
def index():
return     <h1></h1>
```

The index() function is registered as the root directory address in the example. When the server domain name or IP address is accessed in the browser, the server will execute the index() function, referred to as the response.

2.4.2 Front-end framework

Bootstrap, an open-source framework developed by Twitter, provides components to create beautiful, collaborative tune web pages, which are compatible with all web browsers. Bootstrap is a client-side framework.

The server only needs to provide HTML responses that reference the Cascading Style Sheets (CSS) and JavaScript files in Bootstrap and instantiate the required components in the HTML, CSS, and JavaScript code[23]. With Flask, Only need to install the extension package called Flask-Bootstrap to fully realize the function of Bootstrap.

The initialization code for Flask-Bootstrap is:

```
from flask_bootstrap import Bootstrap
#. . .
Bootstrap = Bootstrap(app)
```

where app is the object instantiated by Flask.

block name	instruction
doc	the whole file
html_attribs	<html> tag attributes
html	Content in the <html> tag
head	Content in the <head> tag
title	Content in the <title> tag
metas	A set of <meta> tags
style	Cascading Style Sheet Definition
body_attribs	<body> tag attributes
navbar	User Defined Navigation Bar
content	User-defined page content
scripts	JavaScript declaration at the bottom of the document

Figure 2.6: Blocks defined in the base template

Applying Bootstrap in Flask is very convenient, as the basic component function blocks and styles are defined. Now, the class attribute of the element is assigned to the class defined by Flask-Bootstrap when in use, such as , the navbar block defines a simple navigation bar. Enter When localhost:8000 is entered in a web browser, a simple Flask is finished.

2.4.3 Front-end and Back-end interactions

The server backend and web frontend are built based on the Flask framework and some components such as Flask-Bootstrap. The interaction between the frontend and backend is mainly based on the routing in Flask. For example, when a user clicks on the home page in the navigation, the front end sends a request, the background responds, and the interface of returning to the home page is presented on the browser side. The process is responded to within the system.

On the front end, data is sent, and resources are requested from the server through the form provided in Flask-Bootstrap.On the back end, the route is decorated through the Flask framework decorator app. route(). When accessing the URL address, call the corresponding route is called through the route function or code that returns the desired result.[24]

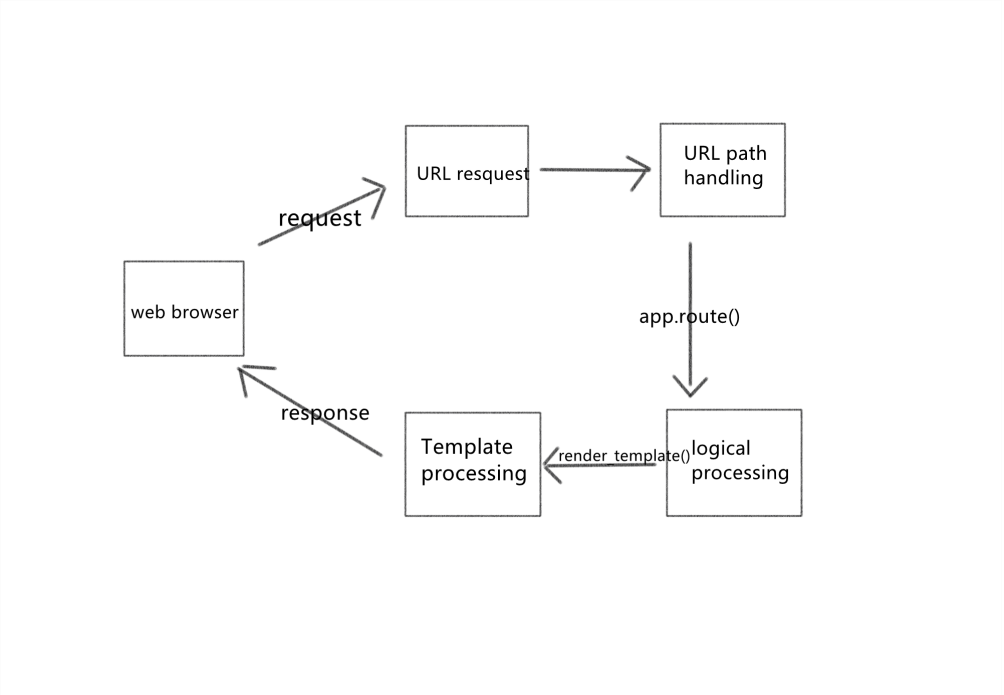

Figure 2.7: System internal response process

2.4.4 Template Engine

A template contains the response text, which contains dynamic variables. The specific values of dynamic variables are only in the context of the request. Substitute the real value for the variable, and then get the final response string, this process is called rendering, and the Jinja2 template engine is designed to render the template[22].

By default, rendered templates are stored in the templates subfolder of the Flask program folder and end in HTML format. In Flask, the render_template function is provided to integrate the Jinja2 template engine into the program. In the template, name A structure represents a dynamic variable, a special kind of placeholder.

Another powerful feature of the Jinja2 template engine is template inheritance, improving the code reuse rate. As long as you first create a base template base.html, put all the required basic elements in the base template. And then, subsequent templates only need to inherit the base template. The inheriting template needs to add the extended directive in the first line of the latter template to declare the need to inherit the template.

Chapter 3

Training and Testing with real data

3.1 The software tools and libraries

The software tools and libraries used for dealing with ML are listed in the following picture.

Tool/Library Name	Formal Library Name	Version	Purpose
Python		3.6.3	Programming language
Pycharm		2017.3.4	Code Editor
Scikit-Learn	scikit-learn	1.0.2	Machine Learning Python library
	scikit-image	0.15.0	
TensorFlow	tensorFlow	1.13.1	Keras dependency
Matplotlib	matplotlib	3.5.1	Library for figures
Numpy	numpy	1.18.1	Math library
opencv_python	opencv_python	3.4.2.16	Cross-platform computer vision software library
Keras	keras	2.2.4	Neural Networks Python library
BeautifulSoup	beautifulSoup	4.10.0	Python library for extracting data

Figure 3.1: List of software tools and libraries used for dealing with ML

3.2 Dataset creation

As introduced in chapter1, the Real Dataset is acquired with a series of laboratory photos made with the University of Turin (Dipartimento di Scienze Agrarie, Forestali e Alimentari) through Dr. Sara Bellezza Oddon.

3.3 Selection and Implementation of Image Segmentation Algorithms

3.3.1 Image processing

Read larva pictures from mobile devices, The saving type of larva pictures is a binary file, and the input and output of the image are uint8 types. Furthermore, resize the pictures.

3.3.2 Binarized image

According to figure 3.2, after comparing different threshold methods, the binarization operation method of the target image is to select the best threshold through the Maximum Between-Class Variance Method(OTSU). Binarize images convert the grayscale image into a black and white image. It is, moreover, dividing the gray image with 256 brightness levels through threshold segmentation to obtain a binary that can reflect the overall local characteristics of the image. Among them, in the grayscale image, the pixel value more significant than the threshold is set to 255, displayed as white; the pixel value less than the threshold is set to 0, displayed as black.

Method	Image	Accuracy (%)
Histogram method	image1	71.5
	image2	77.8
	image3	80.2
OTSU method	image1	91.5
	image2	92.7
	image3	94.2
Adaptive method	image1	89.6
	image2	88.7
	image3	90.1

Figure 3.2: Threshold selection methods

Figure 3.3: Initial image

Figure 3.4: Binarized image

3.3.3 Image opening operation

This experiment uses the open operation to process the binary image to deal with the noise that appears. The opening operation is because the input image is eroded by structuring elements and then expanded [11]. The open operation can convert other pixels of the image into the background, which has a good effect on the separation of the target and the removal of background noise (salt and pepper noise) and can play the role of low-pass filtering.

Figure 3.5: Image opening operation

3.3.4 Image contour detection

The edge detection method of the target image is significant in image analysis, and it is widely used in regional shape extraction, target region recognition, and image segmentation, like processing operations. The edge detection of the image is conducive to the extraction of target pixel features in the image recognition process. In an image, each pixel on the boundary of an object has a gradient of pixel value around the pixel. The change direction and size of the grayscale are used to express the characteristics of the change of the grayscale value of the image. [25].

he pixel value of the image has a significant difference on edge, and the derivative of the pixel difference has a local maximum. Therefore, the edge of the image can be obtained by obtaining the maximum local value of the pixel difference and the

gradient direction.

Since the gradient modulus operator has the characteristics of isotropy and displacement invariance, it can be applied to the edge detection of the image, and the direction of the pixel value change, that is, the direction of the boundary, can be determined by the tangent function of the pixel. The inverse function of is obtained.

In the experiment, the gradient modulus operator is represented by a differential operator, and the gradient is obtained through the convolution function of the differential operator.

Differential operators mainly include Canny operator, Roberts operator, Prewitt operator, Soble operator, and Isotropic Soble operator, this research uses Canny operator for convolution. The canny operator works best in stepped edge detection. The implementation steps of the Canny operator are as follows bottom:
(1) Convolve the image to eliminate noise, and use a 2D Gaussian filter template for convolution.
(2) Use the Sobel operator to find the derivative Gx and Gy of the gray value.
(3) Use the values of Gx and Gy to find the direction of the gradient change.
(4) Calculate the direction of the edge. It is necessary to divide the gradient direction of the image edge into 0°, 45°, 90°, 135°, and record the pixel value in each gradient direction.
(5) Iterate the pixel value calculation. Image pixel The gray value of g is g, where, in the gradient direction, as long as the gray value g of the pixel is less than or equal to any adjacent pixel value, the gray value g of the pixel is equal to 0, that is, the pixel is not an edge pixel .
(6) Image edge detection. Determine the edge of the image by the size of all pixels and the two thresholds g1 and g2 in the gray histogram, where g1>g2. If the difference between the pixel and the threshold g1 is greater than 0, then this pixel is the edge of the image; if the difference between the pixel and the threshold g2 is less than 0, then the pixel is not an edge; if the pixel value is between g1 and g2, then the pixel around the If the other pixel values are all pixels smaller than the threshold, then this pixel is an edge pixel, otherwise it is not.

Figure 3.6: Image edge detection

3.3.5 Gaussian filter denoising

When acquiring images of larvae, due to the influence of imaging equipment, environment, illumination, and shooting background, certain noises are likely to be generated during the imaging process. The noise in the image affects the viewing of the image and has a significant impact on the processing of the image, reducing the accuracy of the image processing result. Therefore, Gaussian filter denoising was used to process the larvae image to eliminate image noise.

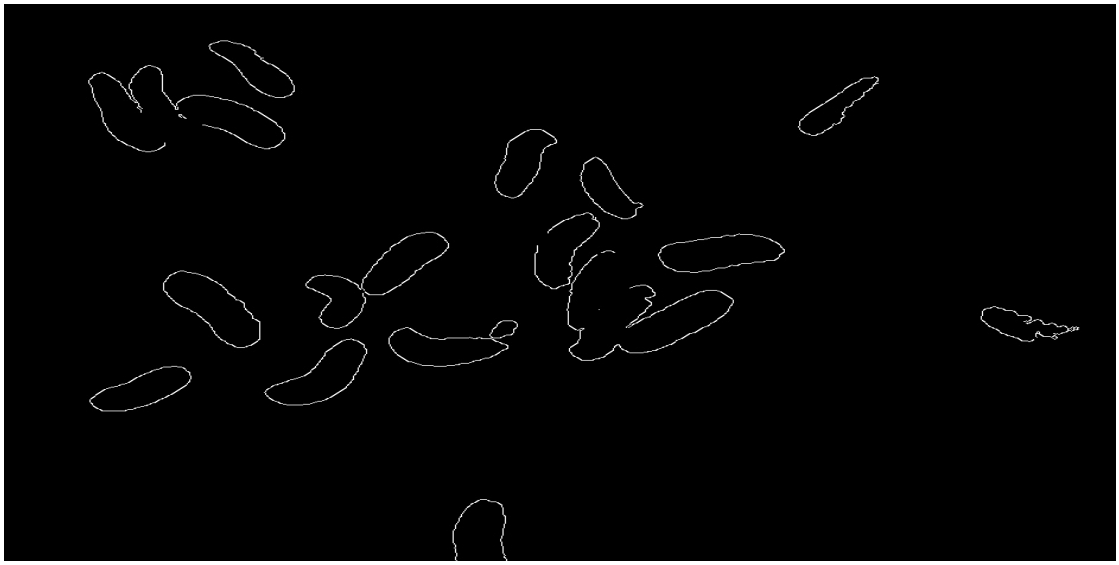

Figure 3.7: Gaussian filter denoising

3.3.6 Get image outline

After the contour detection algorithm is completed, we get the target contour [25].

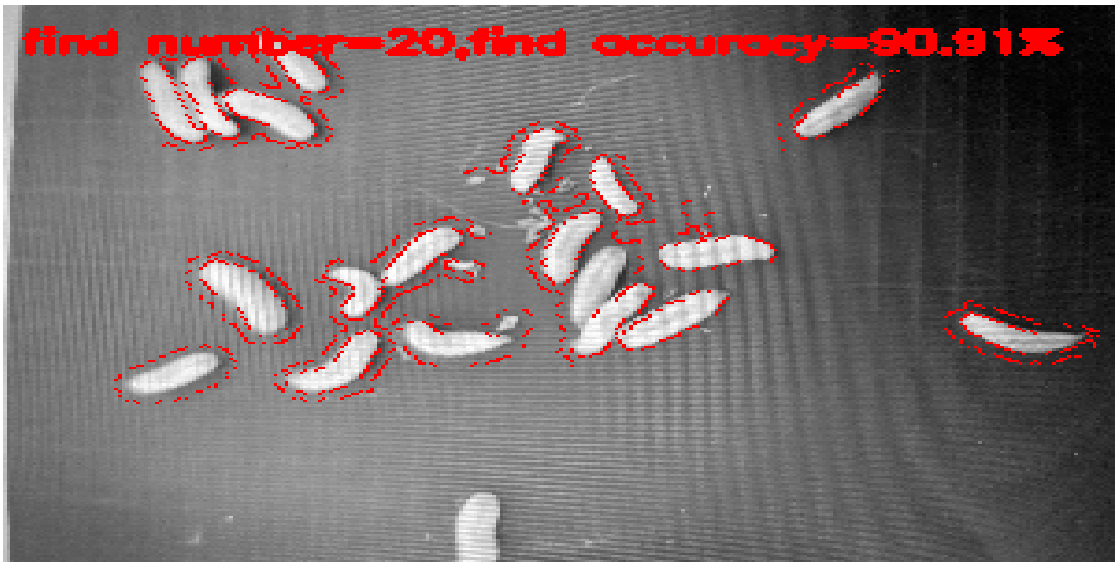

Figure 3.8: Get image outline

3.3.7 Counting result judgment

Count the results of different types of larvae obtained automatically, and use the number of larvae counted manually as a control to calculate the recognition rate(REFR) to evaluate the effect of counting larvae.

Randomly take five photos, the following figure counts the number of larvae by manual counting and automatic counting.

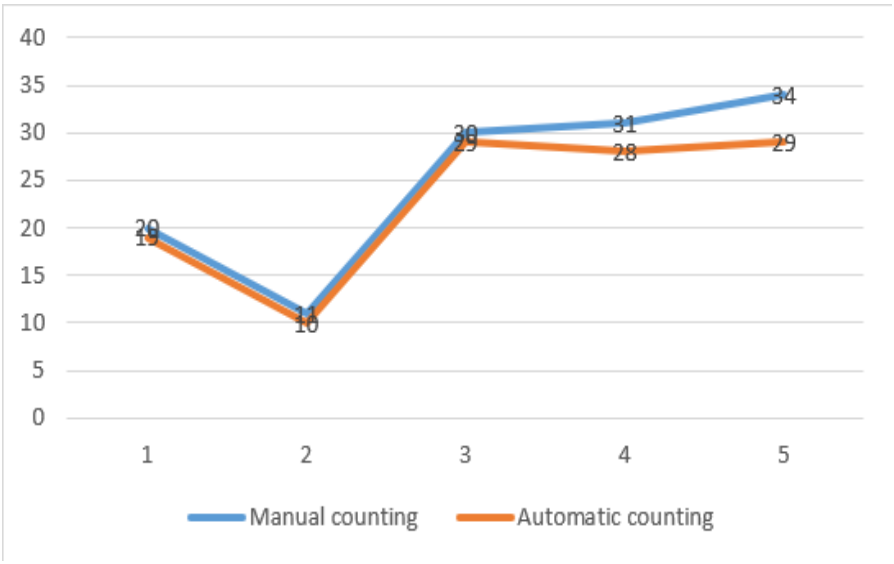

Figure 3.9

$$REFR = (1 - |Nm - Na| \div Nm) \times 100\% \tag{3.1}$$

REFR represents the correct recognition rate, Nm represents the result of manual recognition, and Na represents the result of image recognition.
The average recognition rate(REFR)=91.5%

3.4 Selection and Implementation of Image Recognition Algorithms

3.4.1 Real Dataset preprocessing

The dataset is divided into three parts, namely training set, testing set, and validation set. The training set is used to train the neural network model, and then the validation set is used to verify the model's validity. The model with the best effect is selected until we get a satisfactory model. Finally, when the model "passes" the validation set, we use the test set to test the final effect of the model, evaluate the accuracy of the model, the error, etc. and then resize all images in the directory (overwrite the original file), and then load the pictures in the folder into an array form and form a file label.

3.4.2 Data augmentation

Extend the training set images by flipping them horizontally or vertically.The number of pictures has been increased from the original ten to forty.

3.4.3 Training and candidate models

Considering the performance of the hardware equipment and the training effect,without a large enough data set, it is difficult to train the network model to the ideal classification effect, so the method of transfer learning can be used to achieve small dataset classification task for larvae.

In this thesis, the VGG 16 network model is used for New learning and transfer learning.[26]The model is shown in the figure 3.10, which includes 13 convolutional layers, 5 pooling layers and 3 fully connected layers. There are many model parameters. Figure 3.11 is the flow chart of New learning.

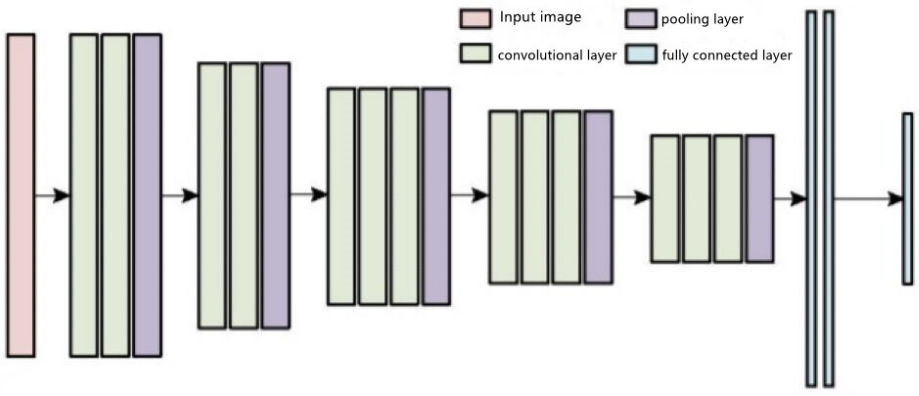

Figure 3.10: Convolutional neural network model

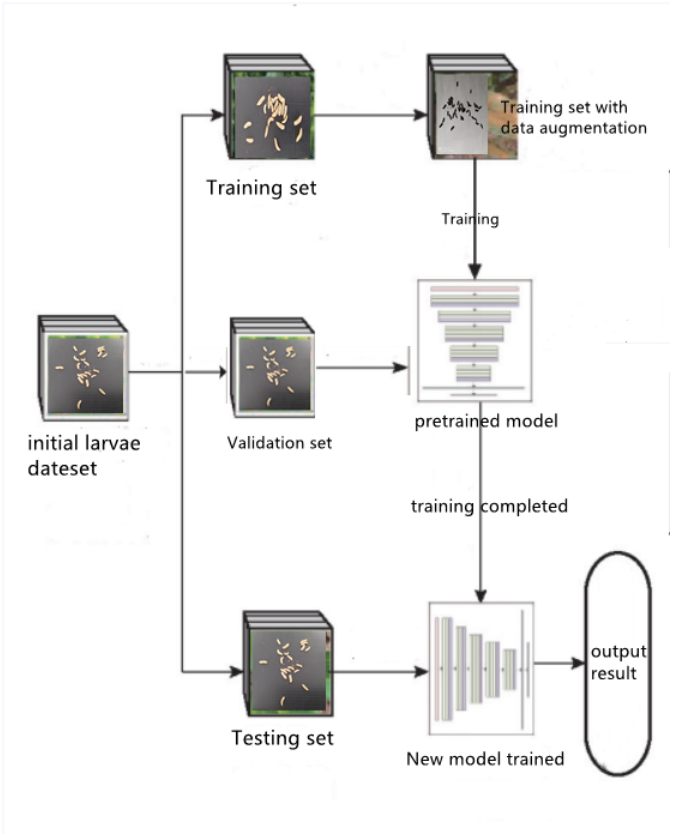

Figure 3.11: Flow chart of insects image recognition method based on New learning

VGG 16 is trained on the ImageNet dataset, The number of larvae images is as high as more than 1 million. However, the images of larvae do not have a large enough dataset at present, and it is difficult to train the network model to the ideal classification effect. Therefore, the transfer learning method can be used to realize the classification task of small datasets of larvae. Preserve the structure before the last convolution module, and then redesign the fully connected module. The improved fully connected module is shown in Figure 3.12.

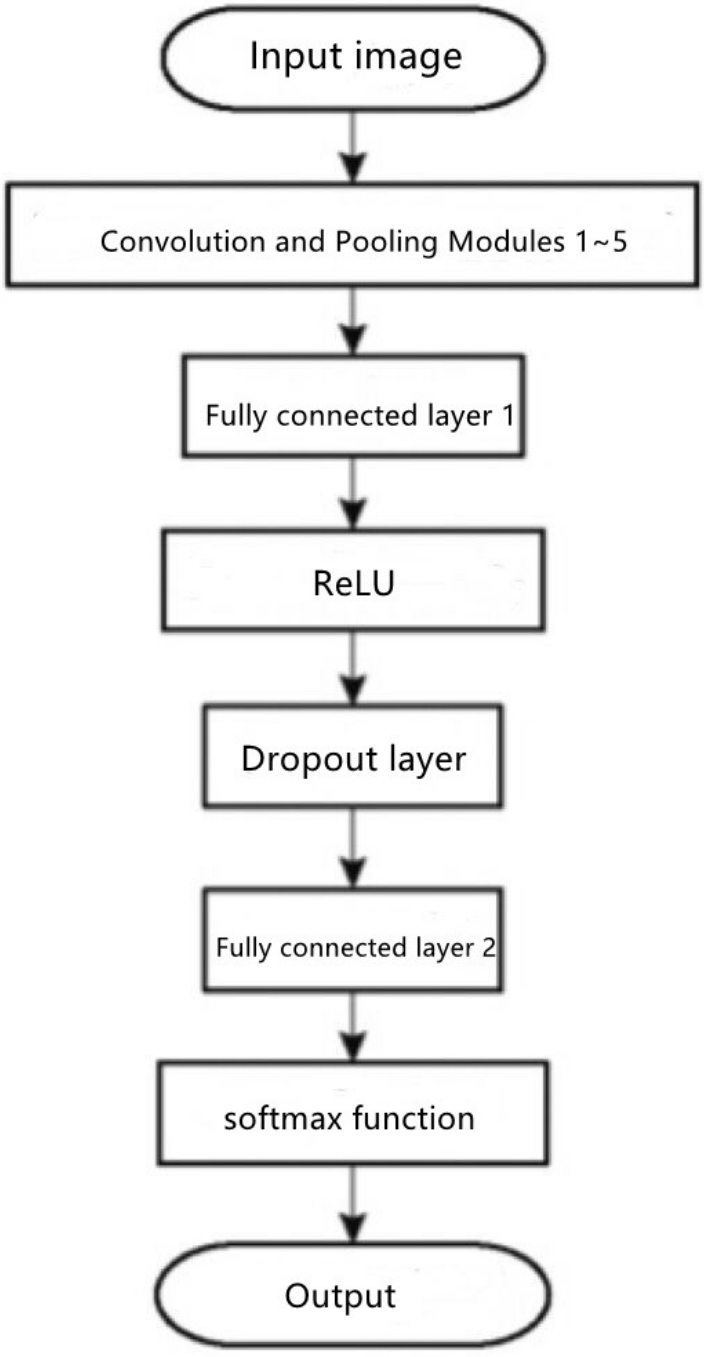

Figure 3.12: Flow chart of improved full connection layer model

The input image can be expanded into a vector through the operations of convolution and pooling modules, and the dimension is reduced to 1×256 through the fully connected layer 1. Then enter the nonlinear activation function, The functions include sigmoid function, tanh function and ReLU function, and the model adopts the ReLU activation function. Compared with the previous two functions, the ReLU function has the characteristics of simple calculation and fast convergence speed. its calculation formula is as follows:

$$f_{ReLU} = \begin{cases} 0 & (t < 0) \\ t & (t \geq 0) \end{cases}$$

$$f_{ReLU} - Relu\ function$$

$$t - Relu\ independent\ variable$$

Then enter the Dropout layer [16] , the Dropout layer is in each training process of the network, according to a certain probability, a part of the neurons weight is temporarily set to 0, which can alleviate the cooperative adaptation between neurons, reduce the dependence between neurons, and avoid overfitting of the network. Then enter the fully connected layer 2 further reduces the dimension of the vector to 1×3,Finally, the softmax function is used to calculate the final classification probability, and its calculation formula is as follows:

$$p_r = \frac{exp(v_r)}{\sum_{k=1}^{3} exp(v_k)}$$

$$v_r - the\ rth\ component\ in\ the\ vector$$

$$p_r - the\ classification\ probability\ of\ the\ rth\ component$$

$$k - the\ serial\ number\ of\ the\ classification$$

The fully connected layer module of VGG 16 is replaced by the newly designed fully connected layer to form a new network model, and then the VGG 16 is used. The weights and parameters of the trained convolutional layer are loaded into the convolutional layer of the newly constructed model, and finally the collected larvae images are used to the new model is trained, and the trained new model can detect and recognize larvae images.The specific process is shown in Figure 3.13.

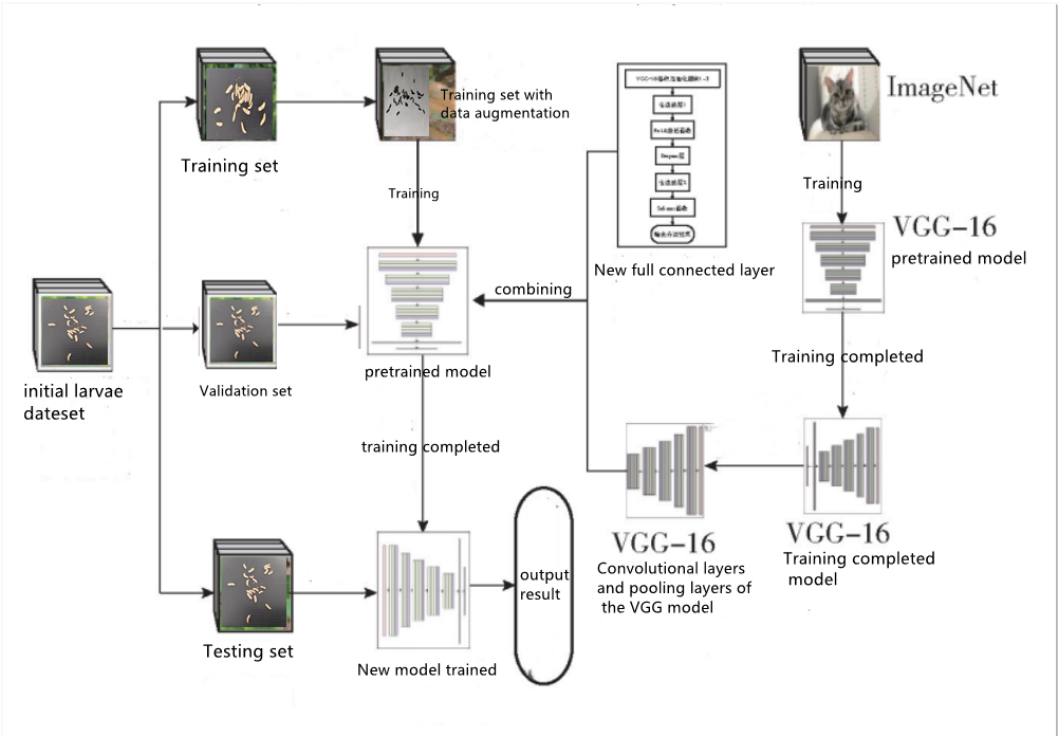

Figure 3.13: Flow chart of insects image recognition method based on transfer learning

3.4.4 Testing, performance evaluations and the best real model

Under different influence factors, the network model is trained, including New learning and two different transfer learning training methods (freezing the convolutional layer, only training the fully connected layer and all layers). There are many different parameters and compositions in the model:

• **Learning rate**: It determines whether the objective function can converge to the local minimum and when it converges to the minimum.

• **Batch_size**: the number of data samples captured in one training.

• **Epoch**: All data is sent to the network, and a forward calculation adds back. propagation process is completed.

• **Loss Function**: A function that maps the value of a random event or its related random variables to non-negative real numbers to represent the "risk" or "loss."

• **Optimizer**: The optimizer is to guide the parameters of the loss function (objective function) to update the appropriate size in the correct direction in deep learning backpropagation so that the updated parameters make the value of the

loss function (objective function) continuously approach the global minimum.

During the experiments, set different parameter values to compare the results. different learning rate (0.01, 0.0001,0.00001); different Batch_size(8,32); different optimizer: Adam,Adadelta,Adagrad,Nadam,SGD; Loss Function: cross entropy; epochs=100. and a total of 18 groups of trials were conducted.
The table below shows the accuracy of different model tests.

Table 3.1: Accuracy of model training and testing under different factors

Training method	**Learning rate**	**Batch_size**	**Accuracy(%)**
New Training	0.01	8	25
	0.01	32	25
	0.0001	8	25
	0.0001	32	25
	0.00001	8	91.67
	0.00001	32	91.67
Transfer learning only trains the fully connected layer	0.01	8	45.3
	0.01	32	47.8
	0.0001	8	95.83
	0.0001	32	96.67
	0.00001	8	95.92
	0.00001	32	95.67
Transfer learning trains all layers	0.01	8	25
	0.01	32	25
	0.0001	8	93.5
	0.0001	32	95.83
	0.00001	8	100
	0.00001	32	100

The effect of learning rate on the model

The learning rate is 0.01 on the initial training of the original VGG 16 with the Image dataset. For these test samples, when the learning rate is set to 0.01, it can be seen from Figure 4.4: only 'Transfer learning only trains FC layer' has better accuracy, the accuracy of 'New Training' and 'transfer learning trains all layers' is 25%. When the learning rate is set to 0.0001, the accuracy is still very low. However, when the learning rate is set to 0.00001, all three training methods can achieve better results. The accuracy has become 100% on Transfer learning trains all layers. So, excessive learning rate fails to train appropriately.

The effect of Batch size on the model

For these test samples, it can be seen from Table 3.1 that for new learning, the accuracy does not change when batch_size is changed. For the transfer learning method, the result when batch_size is equal to 8 is usually only slightly lower than the result when batch_size is equal to 32, so increasing the batch_size has little effect on the accuracy.

The effect of learning method on the model

During training, each epoch trains the training set, verifies the validation set, and then selects the model according to the best validation set accuracy.

The training accuracy and validation accuracy of the three training methods at the learning rate of 0.00001 with the different batch sizes is shown in Figures 3.14-3.17. It can be seen that the new Learning has the slowest convergence speed and large fluctuations. The convergence speed of the Learning only trains fully connected layer is faster than that of the new Learning and finally tends to be stable. Transfer learning has the fastest convergence speed for training all layers. The accuracy rate reached 100% in a few training epochs, indicating that transfer learning shortens the model convergence time.

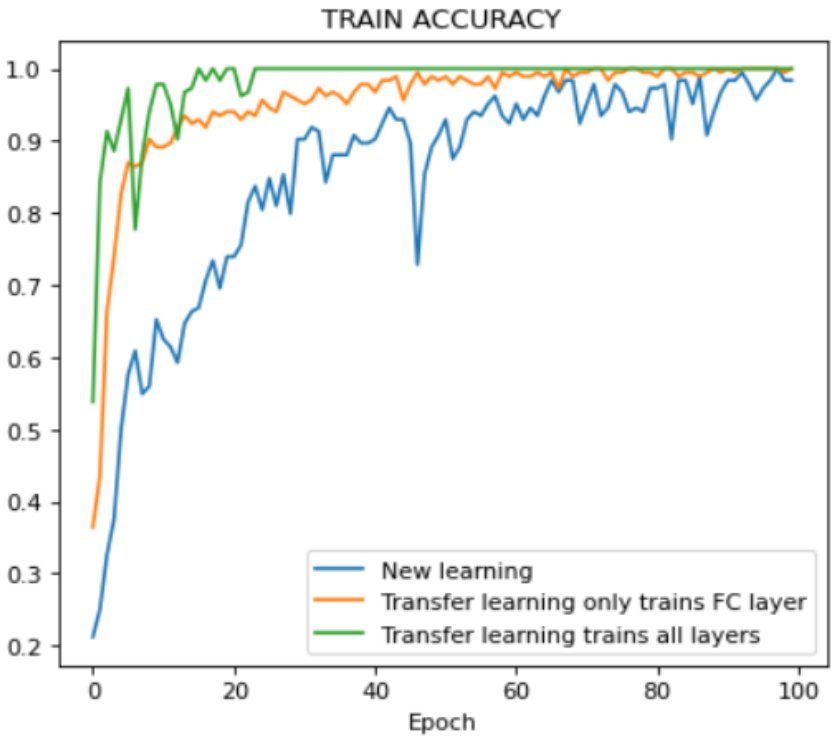

Figure 3.14: lr=0.00001,batch_size=8,epoch=100

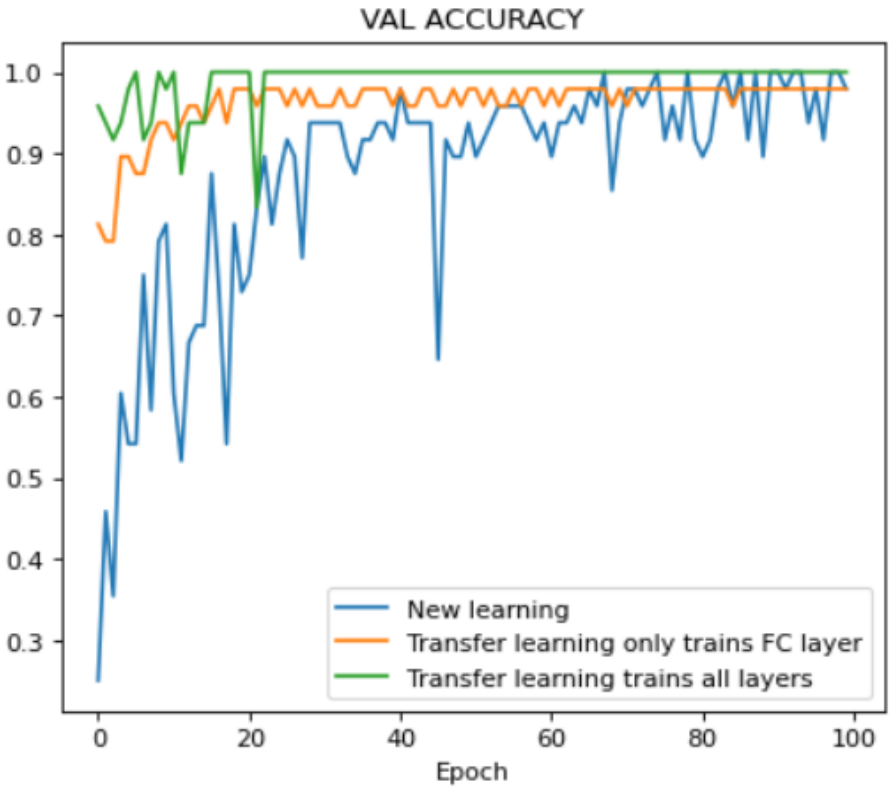

Figure 3.15: lr=0.00001,batch_size=8,epoch=100

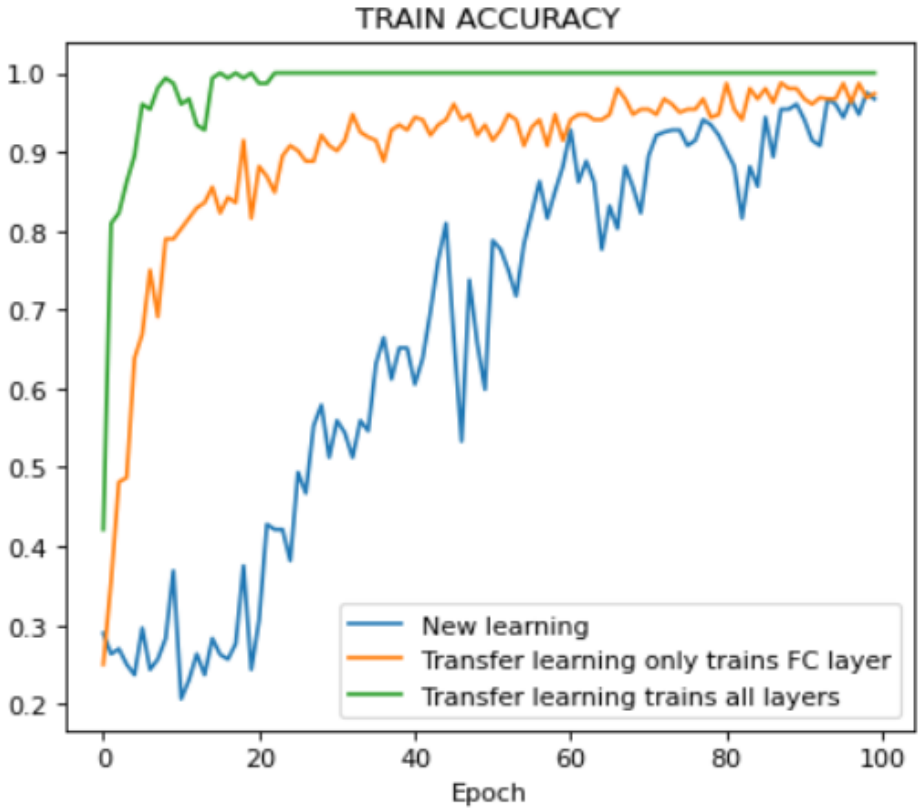

Figure 3.16: lr=0.00001,batch_size=32,epoch=100

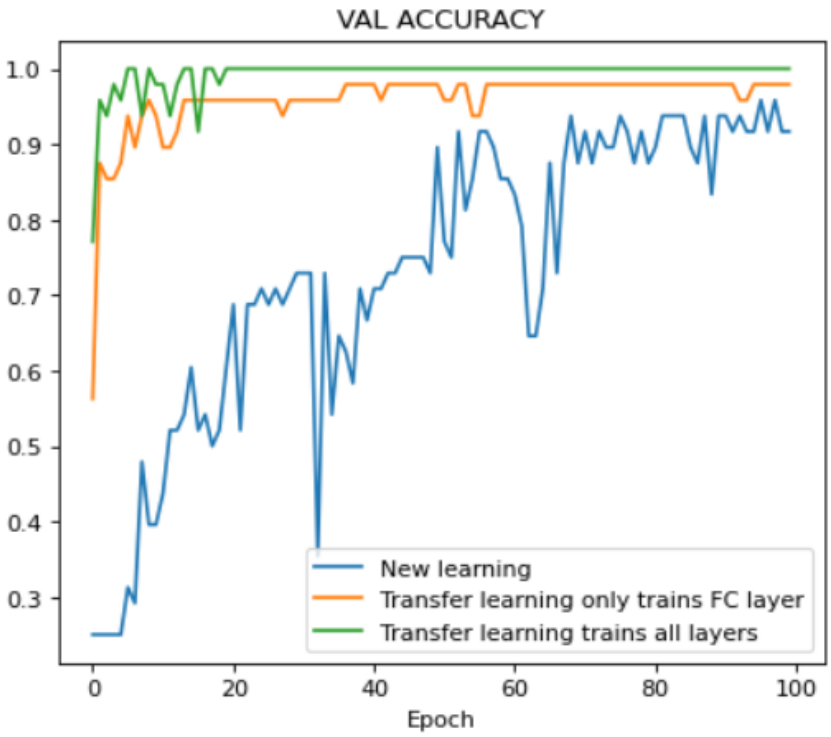

Figure 3.17: lr=0.00001,batch_size=32,epoch=100

It also can be seen from figures 3.14-3.17 that the lowest verification accuracy is the new learning, and the accuracy of transfer learning methods has become almost 100%. Only training and changing the fully connected layer module can not achieve the desired effect, and transfer learning trains all layers can significantly improve the accuracy.

The effect of optimizer on the model

After comparing the effects of different parameters on the recognition rate, combing the previous experiment result, set Batch_size=32, learning rate=0.0001, and other parameters remain unchanged and comparing the effect of different optimizers on training results. It is shown in table 3.2

From table 3.2, we can see that different optimizers have a more significant impact on accuracy. The Adam optimizer has the best effect no matter which training method is used. The Adadelta optimizer has the worst effect, so we still choose the Adam optimizer in the model.

Table 3.2: The effect of optimizer on the model

Training method	**Optimizer name**	**Accuracy(%)**
New Training	Adam	91.67
	Adadelta	25
	Adagrad	43.75
	Nadam	47.92
	SGD	25
Transfer learning only trains the fully conntected layer	Adam	95.67
	Adadelta	25
	Adagrad	75
	Nadam	91.67
	SGD	54.17
Transfer learning trains all layers	Adam	100
	Adadelta	47.92
	Adagrad	95.83
	Nadam	95.83
	SGD	81.93

3.5 Conclusion

In conclusion, compared with new learning, transfer learning shortens the training time, the recognition performance of the model was improved, and it had better recognition accuracy in larvae image recognition. In particular, the transfer learning trains all layers, and the final accuracy rate can reach 100%.

In new learning and transfer learning, the training effect of the learning rate 0.01 is not ideal. In new learning, a large learning rate will cause the loss function value to oscillate or the gradient will explode, and the training effect cannot be achieved. In transfer learning, a large learning rate will lead the parameter update is too fast, and the weight information originally trained in the transfer learning is destroyed.

Chapter 4

Implementation of Insect Recognition System

4.1 System functions

(1)This application recognizes the insect pictures taken by the user and counts the number of larvae.
(2)This application recognizes the insect pictures already in the user's photo album and counts the number of the larvae.

4.2 Implementation of the system

4.2.1 Development Environment and System Components

The entire system was developed using Python 3.6. The Flask framework and Flask extension package required by the entire system are installed through pip install. The components that need to be installed are: Flask, Flask-Script, Flask-Bootstrap, etc., and Flask uses version 0.12.2, Flask-Script uses version 2.0.5, Flask-Bootstrap uses version 3.3.7. Moreover, Development Software is Pycharm that uses version 2017.2.3.

4.2.2 Architecture design

The client is mainly responsible for the user interface and front-end interaction, providing the user with an operator interface to select the recognized images on the interface and obtain the recognition and counting results. The server is mainly responsible for responding to the request sent by the client, making function calls, and returning the corresponding results.

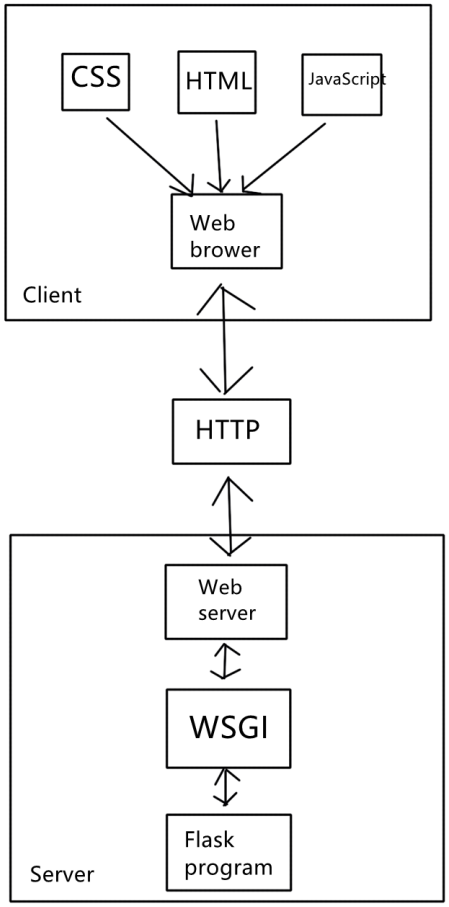

Figure 4.1: Architecture design

4.2.3 Module design

Combined with the system's functions, the system modules are designed in detail. The initialization module is designed for the general application of the flask framework, and the recognition module is designed to realize the system's core functions.

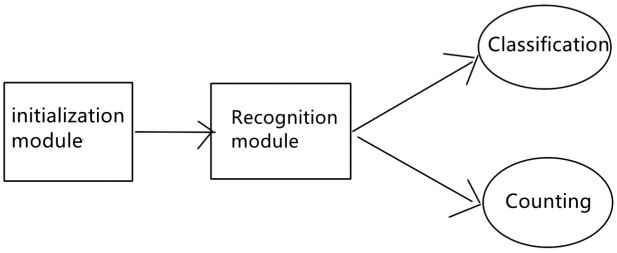

Figure 4.2: System modules

4.2.4 Flask-based backend implementation

1.Initialize module implementation
Execute pip install flask in Python 3.6, and create flask project. This module is mainly responsible for creating and configuring Flask objects, and the system will use some components for initialization.

```python
#encoding: UTF-8
from flask import Flask

app = Flask(__name__)


@app.route('/')
def hello_world():
    return 'Hello World!'


if __name__ == '__main__':
    app.run()
```

Figure 4.3: initialization

2.Function module implementation.
Connect with the route to post requests to upload files. And then save the image path. Moreover, import trained classifier and counting files to classify and count.

```python
sys.path.append('templates')
from skimage import io
from flask import Flask, render_template, request
import json
import time
from flask_cors import *
import spider
from train_model import classifier
from cross_fold import kflod_classifier
from count import count_plus

# Define the Classifier Interface
classifier = classifier
```

Figure 4.4: import python files

```
# post request to upload files
@app.route("/uploadfile",methods=['POST'])
def uploadfile():
    f = request.files['the_file']
    imgName = (time.strftime('%Y%m%d%H%M%S',time.localtime(time.time()))+f.filename

    img_path = 'static/upload/' + imgName

    for filename in os.listdir('static/upload/'):
        os.remove('static/upload/'+filename)


    # save image
    f.save(img_path)

    # recognize and classify images
    class_name = classifier(img_path)

    # Counting the number of targets
    count_value,res_img = count_plus(img_path)

    # Recognition result image path
    result_path = 'static/upload/'+'res'+imgName
    cv2.imwrite(result_path,res_img)
```

Figure 4.5: Function module implementation

4.2.5 Front-end implementation

Flask front-end template uses Jinja2 template engine. In the program, the specific template is rendered by the render_template method, and the template will calculate the variables or functions in the inherited template to form a completed HTML page.

```
@app.route("/")
def shibie():
    return render_template('shibie.html')
```

Figure 4.6

As shown in Figure 4.7, the page uses the Bootstrap framework to beautify the page layout and interface elements, and the action attributes in the form are connected to the routing app.route('/shibie'). Click the icon to send a request to the server.

Figure 4.7: Main interface design

The square icon on the main page provides the main functions. Click the icon, as shown in figure 4.8. You can choose to take a photo right now or choose an

image from an album or files.

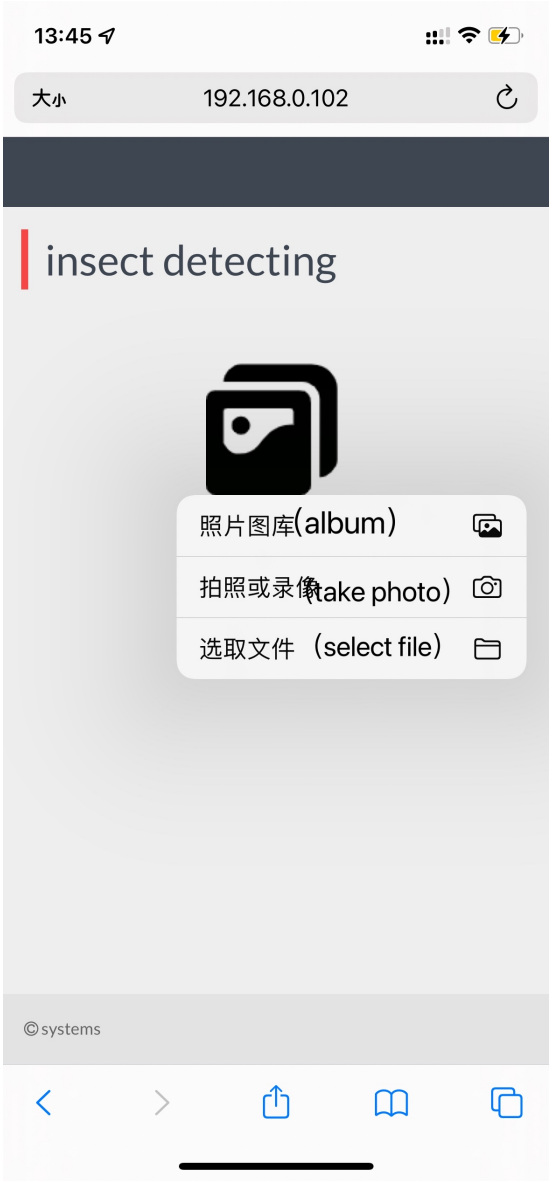

Figure 4.8: After clicking the icon page

As shown in Figure 4.9,After uploading a picture, the action attributes in the form are connected to the route app.route("/uploadfile",methods=['POST']).and server return results.

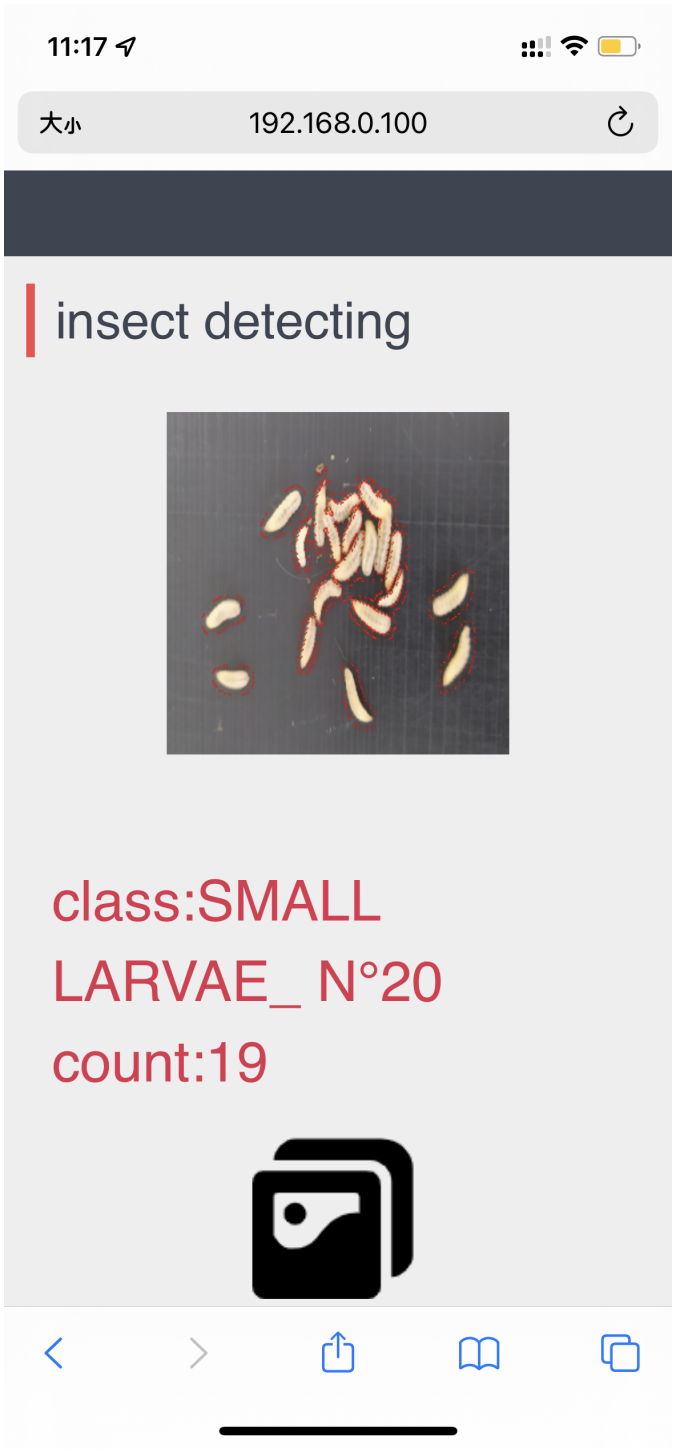

Figure 4.9: Result page

Click the icon again, and you can restart to take photos or choose an image.

4.3 System advantages

1. Lower cost

Because WebApp only needs page development, there is no limitation of development language, and there is no limitation of client and server, which will greatly reduce the cost.

2. Low barriers to use

WebApp can be cross-platform, which saves download and installation time, and does not take up mobile phone memory and mobile phone traffic.

3. High mobility

The development of WebApp applications only needs to be packaged into the Android installation package and the ios installation package, and there is no need to develop them separately for the two systems.

4.4 Conclusion

This chapter mainly introduces the system functions, development process, advantages, and the attached core code.

Chapter 5

Conclusion

5.1 Thesis Summary

The main research contents of this thesis are the development status and trend of artificial intelligence and insect recognition at home and abroad. The background and significance of this research are analyzed. At the same time, the function and development of larvae recognition applications were analyzed. Relevant research directions have been developed. Moreover, design the overall framework and functional modules.

5.1.1 Technical Summary

Image segmentation technology: This thesis uses image segmentation techniques for the primary image processing of the system. The larvae region was segmented by grayscale threshold and image contour detection algorithms. The insect object itself is extracted, and the area segmentation of the insect object and the background is realized.

Image recognition technology: This thesis applies artificial intelligence technologies such as convolutional neural networks and transfer learning to insect image recognition, compares different models' recognition accuracy, and selects the optimal model through experimental analysis. Through experimental analysis, transfer learning has high feedback efficiency and accuracy, which effectively improves the recognition performance of the system.

Data augmentation technology: essential data augmentation is used to transform the image horizontally or vertically to expand the data set, which plays a vital role in improving the recognition rate of the model.

System design and implementation: the flask framework is used to build web applications. In the future learning process, we can also try to use frameworks

such as Django to compare the speed, adaptability, and flexibility of different frameworks.

5.2 Future work

In the comprehensive and sustainable development of biological science and technology, the importance of image recognition and other related technologies is becoming more and more apparent.Vigorously developing artificial intelligence technology can provide more convenience for further research in biological sciences.

In the future, other transfer learning models(like MobileNet, VGG19) can be chosen for a more detailed comparison. Moreover, diversified data augmentation methods can be tried, such as cutting part of the image stretching the image. And with the rapid development of modern technology, taking pictures with mobile phones has become more convenient and fast, which will provide a more convenient way to collect images in the natural environment and help the expansion of data sets. This way will be the primary source of images contained in the field of insect image recognition in the future. However, some problems in the images, such as incomplete pictures of individual insects and overlapping and covering of individual insects, will also be another research direction of image processing.

Bibliography

[1] Yang hongwei and Zhangyun. «Application Progress of Computer Vision Technology in Insect Recognition». In: *Bioinformatics* 3.3 (2005), p. 4 (cit. on p. 9).

[2] Zhu leqing and Zhang zhen. «Image recognition of insects based on sparse coding and SCG BPNN». In: *Acta Entomologica Sinica* 56.11 (2013), pp. 1335–1341 (cit. on p. 9).

[3] Andrzej Cichocki. «Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications». In: *John Wiley Sons, Inc.* (2002) (cit. on p. 9).

[4] M. Nixon and A. Aguado. «Feature Extraction Image Processing». In: *feature extraction image processing for computer vision* (2002) (cit. on p. 11).

[5] H. Kaut and R. Singh. «A Review on Image Segmentation Techniques for Future Research Study». In: *International Journal of Engineering Trends and Technology* 35.11 (2016), pp. 504–505 (cit. on p. 13).

[6] Han Siqi and Wang Lei. «A Survey of Threshold Methods for Image Segmentation». In: *Systems Engineering and Electronic Technology* 24.6 (2002), p. 5 (cit. on p. 14).

[7] G. Bradski and A. Daebler. «Learning OpenCV. Computer vision with OpenCV library». In: *University of Arizona Usa Since* (2008) (cit. on p. 18).

[8] H. Nakano. «Image recognition method and apparatus». In: *US* (1998) (cit. on p. 19).

[9] Zheng Nanning. «Information processing of cognitive processes and novel artificial intelligence systems». In: *Chinese Basic Science* 8 (2000), p. 10 (cit. on pp. 19, 26).

[10] M. Suzaki. «Eye image recognition method eye image selection method and system therefor». In: *Us Patent Us B1* (2001) (cit. on p. 19).

[11] Chen Ru, Deng Zelin, Liu Ding, and Song Zhi. «Using artificial intelligence system to predict the outcome of patients with massive cerebral infarction». In: *China Health Statistics* 31.1 (2014), p. 4 (cit. on p. 19).

[12] C. Roletscheck and T. Watzka. «detection and classification of acoustic scenes and events 2018 challenge using an evolutionary approach to explore convolutional neural networks for acoustic scene classification technical report». In: *Springer Berlin Heidelberg* (2019) (cit. on p. 20).

[13] Li Qingtao, Ren Yuchi, Wang Yuan, Li Linsong, and Chi Zhenye. «Research on abnormal line loss diagnosis method based on artificial neural network fully connected layer optimization». In: *Electrical application* 39.4 (2020), p. 7 (cit. on p. 21).

[14] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz. «Pruning Convolutional Neural Networks for Resource Efficient Transfer Learning». In: (2016) (cit. on p. 22).

[15] D Cook, K. D. Feuz, and N. C. Krishnan. «Transfer learning for activity recognition: a survey». In: *KnowledgeInformation Systems* (2013) (cit. on p. 22).

[16] Lian Xiaoqin, Cheng Kaiyuan, An Sa, Wu Yelan, and Guan Wenyang. «Fruit Image Classification Based on Deep Learning and Transfer Learning». In: *Measurement and Control Technology* 38.6 (2019), p. 4 (cit. on pp. 23, 43).

[17] Wing H. Wong. «The Calculation of Posterior Distribution by Data Augmentation». In: *Journal of the American Statistical Association* 82.398 (1987), p. 528 (cit. on p. 23).

[18] Meng David A. Van Dyk. «Seeking efficient data augmentation schemes via conditional and marginal augmentation». In: *Biometrika* 86.2 (1999), pp. 301–320 (cit. on p. 23).

[19] Abdurrahim Yilmaz, Fatih Goktay, Rahmetullah Varol, Gulsum Gencoglan, and Huseyin Uvet. «Deep Convolutional Neural Networks for Onychomycosis Detection». In: (2021) (cit. on p. 24).

[20] M. Grinberg. «Flask web development : developing web applications with Python». In: *O'Reilly Media, Inc.* (2018) (cit. on p. 26).

[21] G. Dwyer, S. Aggarwal, and J. Stouffer. «Flask : building Python web services : unleash the full potential of the Flask web framework by creating small to large and powerful web applications : a course three modules». In: (2017) (cit. on p. 26).

[22] Shi Baokun, Li Xin, Wang Shuxian, Fan Xiaohan, and Zhang Zhenzhen. «Flask-based Python web development». In: *Digital world* 3 (2020), p. 2 (cit. on pp. 26, 29).

[23] Wang han and Honglei. «Design and Implementation of Intelligent Insect Recognition APP Based on Image Recognition». In: *software* 041.001 (2020), pp. 118–120 (cit. on p. 27).

[24] Qiu Daoyin, Zhang Hongtao, Liu Xinyu, and Liu Yannan. «Field pest detection system based on machine vision». In: *Journal of Agricultural Machinery* 38.1 (2007), pp. 120–122 (cit. on p. 28).

[25] C. Wen, D. E. Guyer, and W. Li. «Local feature-based identification and classification for orchard insects». In: *Biosystems Engineering* 104.3 (2009), pp. 299–307 (cit. on pp. 33, 37).

[26] Shi Xiangbin, Fang Xuejian, Zhang Deyuan, and Guo Zhongqiang. «Image Classification Based on Deep Learning Hybrid Model Transfer Learning». In: *Journal of System Simulation* 28.1 (2016), p. 8 (cit. on p. 39).