

ITC 2/51 Information Technology and Control Vol. 51 / No. 2 / 2022 pp. 376-389 DOI 10.5755/j01.itc.51.2.29741	Classification for Fruit Fly Images Using Convolutional Neural Networks	
	Received 2021/09/04	Accepted after revision 2022/01/23
	 http://dx.doi.org/10.5755/j01.itc.51.2.29741	

HOW TO CITE: Chen, S., Chen, J. (2022). Classification for Fruit Fly Images Using Convolutional Neural Networks. *Information Technology and Control*, 51(2), 376-389. <http://dx.doi.org/10.5755/j01.itc.51.2.29741>

Classification for Fruit Fly Images Using Convolutional Neural Networks

Shuping Chen, Jingjin Chen

Chongqing Aerospace Polytechnic; Chongqing 40021, P.R China; e-mail:spchencq@163.com

Corresponding author: (Shuping Chen) spchencq@163.com

The discriminative information of fruit fly images often exists in a very fine region, resulting in hard extracting the desired features from the entire image level. Hence, it is a tough task to obtain the discriminative features in the fine-grained image for the classification of fruit fly images. To address this, here proposed a convolutional neural network of bilinear pooling based on feature fusion for the classification of fruit fly images. Firstly, Gaussian blur is performed on fruit fly images to reduce the detailed level of fruit fly images, which conveniently extracts high-level features. In the fruit fly images processed by Gaussian blur, the convolutional layers in the proposed model (i.e., consisting of CNN A and CNN B) are used for the features extraction from the processed images. Then, these extracted features are fused by using the matrix dot multiplication in the Bilinear layers. According to these fused features, the softmax layer performs the classification of fruit fly images. Experimental results on the 4000 image set containing different fruit fly morphology show that the propose method not only outperforms the state-of-the-art methods in the classification accuracy of fruit fly images, but also the precision, recall and score are above 94.87% on testing set and validation set. We find that image feature fusion has positive effects on promoting the accuracy of image classification. We also demonstrate that using feature connection operation after performing the matrix point multiplication operation is beneficial to feature fusion, instead of using feature connection operation directly. In addition, our findings indicates that convolutional neural networks easily obtain the desired features from the images performed by Gaussian blur.

KEYWORDS: Convolution neural networks, feature fusion, fruit fly, Gaussian blur, image classification.

1. Introduction

Bayberry is one of characteristic fruits in Chongqing, China. However, in recent years, bayberry fruit flies have eaten a lot of bayberry fruits. Especially, bayberry fruit flies flourish in June and July in the Chongqing region, so that the insect fruit rate is as high as 70%. As such, bayberry fruit flies are considered as a dangerous pest since they can cause potentially huge economic losses to the bayberry industry in Chongqing. Indeed, the classification of fruit flies is a meaningful link of quarantine works [5], [10]. Hence, the classification of fruit fly images is a research hot spot.

Currently, collecting image features of the fruit fly more used the artificial markers, whereas, in order to high accurately label features in the fruit fly wings, chest and back, this process needs to maintain the image sharpness using Camera. In addition, the feature labeling by artificial and traditional manners is low fault-tolerant, as well as, the process cumbersome and high cost.

The classification of fruit fly images is a kind of image classification, indeed, the research of image classification includes the studies both image features extraction and classification algorithms, such as support vector machine based classifier [3] and bag of visual word [15], automatic image analysis based on artificial intelligence techniques [17], etc. While image features extraction suffers from the gap between the low-level image features represented by machines and the high-level semantic information perceived by humans.

Recently, deep networks have earned good results in the field of image classification, such as, the VGG in [20], the GoogleNet in [7] and DenseNet in [23]. While for multi-label image classification, the deep networks in [19] and in [25] are used and gain advanced classification results. Since deep networks can auto-learn image features from massive images [26], thereby avoiding manual feature extraction, deep networks gain high accuracy and efficiency for image classification. Especially, convolutional neural networks (CNNs) show outstanding ability on images [32], [31], e.g., the [29] and the [30]. Certainly, CNNs can be also used a classifier, for instance, the [18] applies CNNs as a classifier for image processing. In addition, CNNs is also widely used in remote sensing image

classification, for instance, in [6, 11, 37, 2, 35, 4, 9, 34, 36, 26], CNNs perform image classification according to the schemes of patch-based or pixel-to-pixel, which achieves pixel-level image classification (i.e., semantic segmentation), so as to gain a high classification precision. Whereas, using the pixel-to-pixel scheme needs to mark tedious image annotations. Indeed, it is a challenge that labels tedious image annotations for multi-classification issues. Usually, the pixel-to-pixel scheme is more beneficial for binary classification issues than for multi-classification ones.

In the field of agriculture, deep networks are used for identification and classification of crop pest, and detection of various insects, etc. While for the classification of fruit flies, their chest and back features are often used as the learning goals of deep networks, e.g., after the local binary patterns (LBP) operator extracts the features of real fly wings and mid-thoracic back plane [27], the AdaBoost algorithm [27] can complete the classification of fruit flies, which gains a good classification accuracy and robustness. However, the classifier constructed by the AdaBoost algorithm is complex so that the classification complexity of the extracted features is increased. In addition, according to the extracted local stripe features for the chest and back of fruit flies, using the BP neural network [33] achieves the classification of fruit flies. Similarly, through filtering and locking the edge of the fruit fly images, after extracting the shape feature parameters in the locked area, the BP neural network in [28] realizes the classification of fruit flies. The classification accuracy in [33] and in [28] relies on the extracted features, so the extracted features determines the classification results of fruit flies. Moreover, according to the calculation results on the Euclidean distance between the points in wings of fruit flies [16], using a support vector machine (SVM) can also identify the classification of fruit fly images. Obviously, the selection about the points in wings of fruit flies is critical because of affecting the calculation of distance. Indeed, the selection of points is a tough issue in itself. By extracting the middle-level features on the vein of the wings, Maced [12] et al. identify fruit flies using Multi-layer perceptron (MLP) and SVM, respectively, and the obtained average accuracy is

88.9% and 87.7%. As a result of acquiring the middle-level features, the classification accuracy in [12] is not sufficiently high.

Usually, the discriminative information of fruit fly images exists in a very fine region, so that it is not easy to extract the desired features of the entire image level by using deep networks directly. As such, it is a tough challenge to obtain the discriminative features in the fine-grained image for the classification of fruit fly images. To address this, this work proposes a convolutional neural network based on feature fusion for the classification fruit fly images. Through the fusion of image features, the classification accuracy of fruit fly images can be improved. As an important application, the classification of fruit fly images is essential for further preventing fruit flies from harming citrus in Chongqing city, China, and reduce economic losses. We summarize the contribution in this work as following.

- 1 A convolutional neural network based on feature fusion is proposed for image classification, obtaining the desired results on the classification of fruit fly images.
- 2 The manner of fusing image features has positive effects on promoting the accuracy of image classification. As for feature fusion, the operation of performing feature connection after performing the matrix dot multiplication to the features is better than that of performing feature connection directly.
- 3 Using Gaussian blur for performing images is more helpful for convolutional neural networks to extract high-level features from images.

2. Methodology

The overall scheme of our method is that using Gaussian blur decreases definition of fruit fly images to extract high-level features. Then, the mid-level and high-level features are extracted from the fruit fly images, respectively, and the extracted mid-level and high-level features are fused in the Bilinear layers. According to the features fused, the classification of fruit fly images is implemented in the softmax layer.

2.1. Gaussian Blur

Gaussian blur, i.e., Gaussian filter, is a linear smoothing filter widely used in image processing. In many appli-

cation, Gaussian blur can effectively reduce the image noise and image detail level [12], whose principle is that an image is convoluted using the normal distribution, then the transformation of each pixel in the image is calculated. For the n -dimensional space, the normal distribution equation is given in Equation (1).

$$G(u, \sigma) = \frac{1}{\sqrt{2\pi\sigma^{2^n}}} e^{-\frac{(x-u)^2}{2\sigma^2}}, \quad (1)$$

where u and σ are mean value and standard deviation of x , respectively. Using Equation (1), the definition in the 2-dimensional space can be derived as following.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}. \quad (2)$$

The convolution operation between an image I_{image} and a filter G is given in Equation (3).

$$C_{\text{omn}}(x, y, \sigma) = G(x, y, \sigma) \otimes I_{\text{image}}(x, y). \quad (3)$$

where the item \otimes represents the convolution operation.

2.2. Model Description

In this subsection, our motivation is to integrate information from different channels to achieve fine-grained classification. Hence, we design our model and give the manner of feature fusion.

A. Model Architecture

Traditional manners of feature fusion, such as summation or averaging, only use first-order statistics, i.e., feature concatenation. This can be regarded as direct sum mathematically. Unlike traditional manners, bilinear pooling [13] uses second-order statistical information, which is intended to use the difference in second-order information for classification when the first-order information is the same. Bilinear pooling calculates the outer product of different spatial positions, and obtains bilinear features through calculating average pooling for different spatial positions. The outer product captures the paired correlation between feature channels, and is translation invariant. Hence, bilinear pooling can be regarded as direct product, which can be used for fine-grained classification. More importantly, bilinear pooling [14] provides a stron-

ger feature representation than traditional manners, and can be optimized end-to-end. Convolution neural networks (CNNs) can extract mid-level features containing detailed information [26], which is beneficial to obtain the relations between features. Given those advantages between CNN and bilinear pooling, we propose the CNN of bilinear pooling based on feature fusion for the classification of fruit fly images.

In the proposed model, as shown in Figure 1, the convolution layers are used to extract the mid-level and high-level features. These extracted mid-level and high-level features are allocated the Bilinear layers separately. In Figure 1, after an image passes CNN A and CNN B, their outputs at each location consist of the matrix outer product and average pooled, thereby obtaining the bilinear features representation. For our model, it has two parallel Bilinear layers. After the fusion of features using the two Bilinear layers, the classification of fruit fly images is implemented in the softmax layer.

The architecture of convolutional and pooling is divided into three branches, as shown in Figure 1, and the performing each branch using different convolutions, as follows.

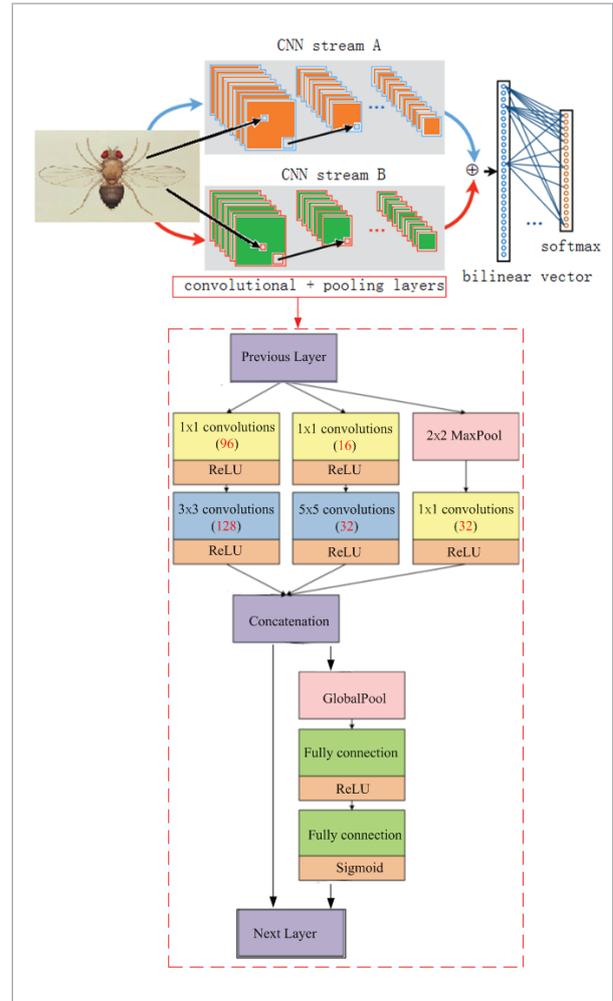
- 1 A $1 \times 1 \times 96$ convolution and a ReLU, followed by a $3 \times 3 \times 128$ convolution and a ReLU.
- 2 A $1 \times 1 \times 16$ convolution and a ReLU, followed by a $5 \times 5 \times 32$ convolution and a ReLU.
- 3 A 2×2 max pooling with stride 1, followed by a $1 \times 1 \times 32$ convolution and a ReLU.
- 4 After concatenating the above three branches, a global pooling and two fully connected layers are connected in series in turn.

B. Feature Fusion

Next, let us calculate feature fusion. Let bilinear operation denote as $B_o=(f_1, f_2, P, C)$, where f_1 and f_2 are the feature functions to two bilinear layers. $f_1=(m, p_1)$ and $f_2=(h, p_2)$, where m and h are mid-level and high-level features. p_1 and p_2 are convergent functions in bilinear layers. P and C are the pooling and classification functions, respectively. The bilinear operation and feature concatenation are calculated as following.

$$\begin{cases} m = \text{sign}(m) \odot \text{sqrt}(|m|) \\ h = \text{sign}(h) \odot \text{sqrt}(|h|) \\ f^n = m \oplus h \end{cases}, \tag{4}$$

Figure 1
Model architecture



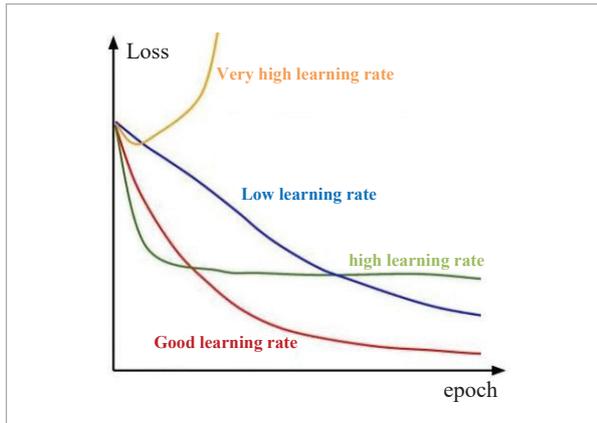
where, the \odot represents matrix dot multiplication. The item \oplus represents concatenation. Using Equation (4), the n -dimension feature vector f^n can be gotten through concatenating mid-level feature m and high-level feature h . The f^n is used as the input of the softmax layer.

C. Hyper Parameters and Model Training

In the training process of neural networks, hyper parameters, such as batch size, base learning rate, etc., often need to be determined, and training epoch needs to be determined, in order to ensure that networks have converged to a suitable value, also to prevent over-fitting. Hence, we carefully study following hyper parameters.

Figure 2

Relation of learning rate, loss and epoch



- 1 Learning rate and training epoch. During the training process, if loss starts to decrease quickly, and it is obvious fluctuation in the training later. This may be caused by the high learning rate. In addition, if there is a slight upward arc in the loss, there may be over-fitting, so training epoch should be reduced. The relation between learning rate, loss and epoch is as shown in Figure 2, which indicates that loss relies on good learning rate and a proper epoch. Consequently, we use cross-validation to determine learning rate and training epoch, i.e., learning rate, denoted as L , $L \in \Omega$, and training epoch, denoted as E , is determined by varying from $E=e_1$ to e_2 with a step of Δe . Let $\Omega = \{1e-2, 1e-3, 1e-4, 1e-5, 1e-6, 1e-7\}$, $e_1=100$, $e_2=1000$, and $\Delta e=50$.
- 2 Batch size. If batch size is too large, local optimum may occur. Oppositely, a small batch size introduces greater randomness, so as to difficulty achieve convergence. Usually, within a certain range, the larger the batch size is, the more accurate the descending

direction is determined, and the smaller the training shock will be. Indeed, batch size increases to a certain extent, whose determined descending direction has basically no longer changed. Hoffer [8] et al. indicate that the performance degradation of a large batch size is because training time is not long enough. Smith [21], [22] et al. show that for a fixed learning rate, there is an optimal batch size that can maximize test accuracy, moreover, batch size is positively correlated with learning rate and a training set scale. Respecting batch size, epoch and accuracy, etc., Figure 3 [8], [21], [22] displays their relation.

Figure 3 shows that batch size is too small, resulting in difficultly converging within 200 epochs. As batch size increases, the speed of processing the same amount of data increases. Since the final convergence accuracy falls into different local extremes, batch size is increased to a certain size, thus achieving the best final convergence accuracy. Using Figure 3 a reference, we dynamically adjust batch size according to the change relations between epoch and precision during training.

- 3 Convergence judgment. Existing methods are hardly to precisely determine with certainty whether neural networks have been fully trained and reached the optimal value, but it is still possible to evaluate whether networks have converged using some manners, so as to help us stop training at a suitable position (i.e., get better good results, meanwhile has no occurred over-fitting). One of the methods is to monitor the loss of a training set and a testing set, i.e., obtaining training curves. When the training loss and the testing loss are maintained in a relatively stable state and the gap between the two is almost unchanged, networks are considered to be basically trained.

Figure 3

Batch size [8], [21], [22]

Batch Size	5000	2000	1000	500	256	100	50	20	10	5	2	1
Total Epoches	200	200	200	200	200	200	200	200	200	200	200	200
Total Iterations	1999	4999	9999	19999	38999	99999	199999	499999	999999	1999999	cannot converge	
Time of 200 Epoches	1	1.068	1.16	1.38	1.75	3.016	5.027	8.513	13.773	24.055		
Achieve 0.99 Accuracy at Epoch	-	-	135	78	41	45	24	9	9	-		
Time of Achieve 0.99 Accuracy	-	-	2.12	1.48	1	1.874	1.7	1.082	1.729	-		
Best Validation Score	0.015	0.011	0.01	0.01	0.01	0.009	0.0098	0.0084	0.01	0.032		
Best Score Achieved at Epoch	182	170	198	100	93	111	38	49	51	17		
Best Test Score	0.014	0.01	0.01	0.01	0.01	0.008	0.0083	0.0088	0.008	0.0262		
Final Test Error (200 epoches)	0.0134	0.01	0.01	0.01	0.01	0.009	0.0082	0.0088	0.008	0.0662		

- 4 Activation function. The activation functions of convolutional layers and fully connection layers are given in Figure 1, i.e., ReLU and Sigmoid are used as their activation functions.
- 5 Model training. Table 1 displays the overall algorithm. In the algorithm 1, some parameters are firstly initialized, and the division of sample is given, in step 1 to step 5. Sample sets are divide into three parts, including training set, *Train_set*,

Table 1

Algorithm 1

```

1 Initialization parameters,  $Q$ ,
 $\Omega = \{1e-2, 1e-3, 1e-4, 1e-5, 1e-6, 1e-7\}$ 
 $batch\_size=64$ ,
 $e_1=100, e_2=1000, \Delta e =50$ , etc;
2 Input sample dataset  $X$ ;
3  $Train\_set$  is obtained by random selecting 80% of  $X$ ;
4  $Test\_set$  is obtained by 10% of  $X$ ;
5 Let  $Val\_set = X - Train\_set - Test\_set$ ;
6 for  $E = e_1$  to  $e_2$  with step  $\Delta e$  do:
7   foreach  $L$  in  $\Omega$ 
8     Use training set  $Train\_set$  to train NN;
9     Calculate training accuracy
       TrainAcc = NN( $Train\_set; E; L; batch\_size$ );
10    Adjust  $batch\_size$  according to TrainAcc;
11    Test NN using testing set  $Test\_set$ ;
12    Calculate testing accuracy
       TestAcc = NN( $Test\_set; E; L; batch\_size$ );
13   end foreach
14 end for
15 Get the optimal parameter value
   Opt( $E; L; batch\_size$ ) = arg max(TestAcc)
16 for  $q=1$  to  $Q$  do:
17   Train NN using  $Train\_set$  and Opt( $E; L; batch\_size$ );
18   Calculate training accuracy
       Train_Acc( $q$ ) =
         NN( $Train\_set; Opt(E; L; batch\_size)$ );
19   Validate NN using validation set  $Val\_set$ ;
20   Calculate validation accuracy
       Val_Acc( $q$ ) = NN( $Val\_set$ );
21 end for
22 Calculate average training accuracy
   AvgTrAcc =  $(\sum_{q=1}^{q=Q} Train\_Acc(q)) / Q$ 
23 Calculate average validation accuracy
   AvgValAcc =  $(\sum_{q=1}^{q=Q} Val\_Acc(q)) / Q$ 
24 Output AvgTrAcc, AvgValAcc;
```

testing set *Test_set*, and validation set *Val_set*. For batch size, we refer to the value in Figure 3, let initial value be equal to 64, i.e., $batch_size=64$. Then, the parameter cross-validation is presented in step 6 to step 15, in order to gain the optimal values for $E, L, batch_size$, denoted as $Opt(E; L; batch_size)$. In addition, NN represents our model. The procedure between step 16 and step 21 shows that NN is trained using training set and $Opt(E; L; batch_size)$. Once NN is well trained, NN is verified by validation set *Val_set*. Finally, the average training and testing accuracy are sent out, as shown in step 22 and step 24.

3. Experimental Settings

3.1. Dataset

We selected 6 fruit flies with different morphology to form an image set as show in Figure 4, including 420 pictures of morphology1, 430 pictures of morphology2, 400 pictures of morphology3, 420 pictures of morphology4, 430 pictures of morphology5, and 500 pictures of morphology6. For the 2600 pictures, 1400 pictures of them are randomly selected to be enlarged by rotating, reversing and adjusting the brightness and darkness. Then, the experimental image set consists of the 2600 pictures and 1400 pictures, i.e., the experimental image set is 4000 pictures. 80% of pictures for each sample are randomly selected as the training set, and 10% of pictures for each sample are used as the testing set. The remaining 10% is used as the validation set. The experimental image set description is listed in Table 2.

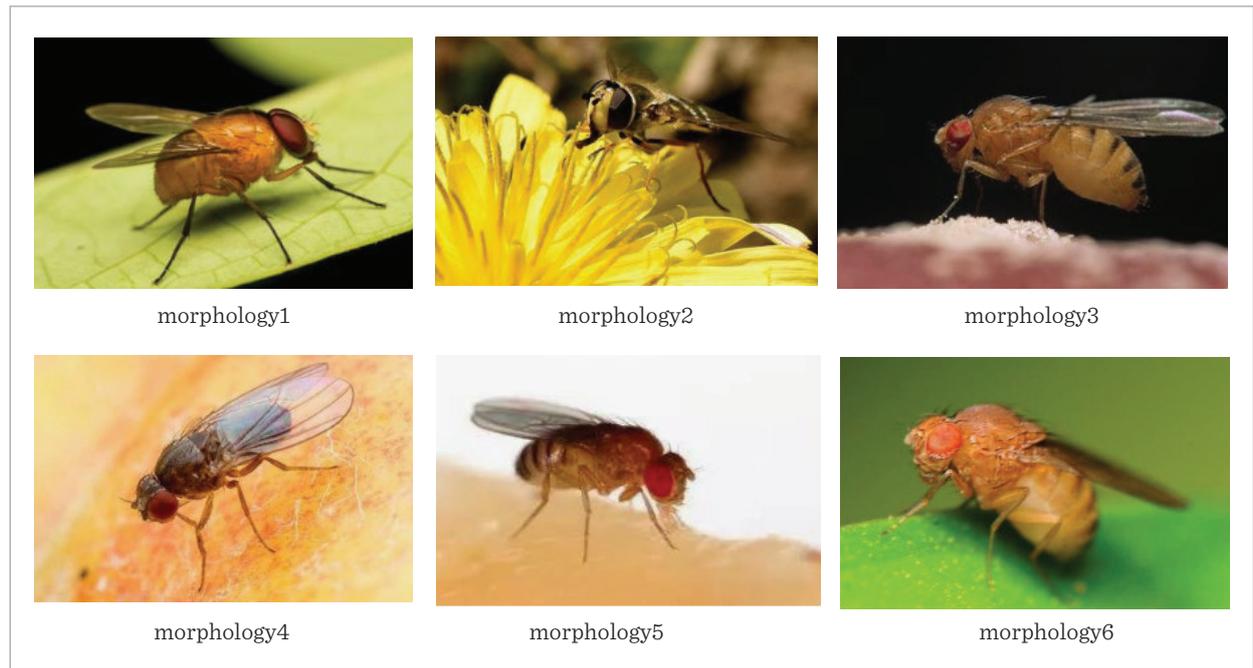
Table 2

Experimental image set

Category	Training	Testing	Validation
Morphology 1	336	42	42
Morphology 2	344	43	43
Morphology 3	320	40	40
Morphology 4	336	42	42
Morphology 5	344	43	43
Morphology 6	400	50	50

Figure 4

Image set of fruit fly. The dimension of image is 300×200



3.2. Assessment Metrics

Each image in the experimental image set has four conditions, including True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). Three assessment indicators are considered in this work in order to verify the ability for our method, i.e., precision, recall rate and score, as following.

$$\begin{cases} \text{Precision} = \frac{TP}{TP + FP} \\ \text{Recall} = \frac{TP}{TP + FN} \\ \text{Score} = \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \end{cases} \quad (5)$$

3.3. Comparison Methods

In order to objectively assess the classification ability of our method, four mainstream methods are used for comparison, i.e., AlexNet in the [17], DenseNet in the [23], CNN in the [25] and CNN the [24]. In addition, to address a fair conclusion, as for the selected four comparison methods, their optimal parameters observed in the corresponding literature are used. Unless otherwise stated, all experiments run on the same GPU, and using the same environment.

4. Results

In this section, entire experimental results are exhibited, including parameter validation, validation of feature fusion, and classification accuracy. Together, the results show that the propose method outperforms the state-of-the-art methods in term of the classification accuracy of fruit fly images. The precision, recall rate and score of the proposed method are above 94.87% on testing set and validation set. Section down below detailed these results.

4.1. Parameter Validation

To achieve the optimal performance of our model, four group of Gaussian matrix is considered to perform image blur, including (10,10), (20, 20), (30, 30), (50, 50).

The results in Figure 5 show that training and testing accuracy are highest on the Gaussian matrix (10, 10) when learning rate is equal to 0.0001, i.e., training and testing accuracy reaches 98.27%, 97.11%, respectively. As such, $1e-4$ is chosen as learning rate to train our model. The final choice is to set the Gaussian matrix as (10, 10), and sample images are blurred. Figure 6 displays the degree of blurring.

Figure 5

Results of different Gaussian matrix. Average is obtained using experiment 100 times independently

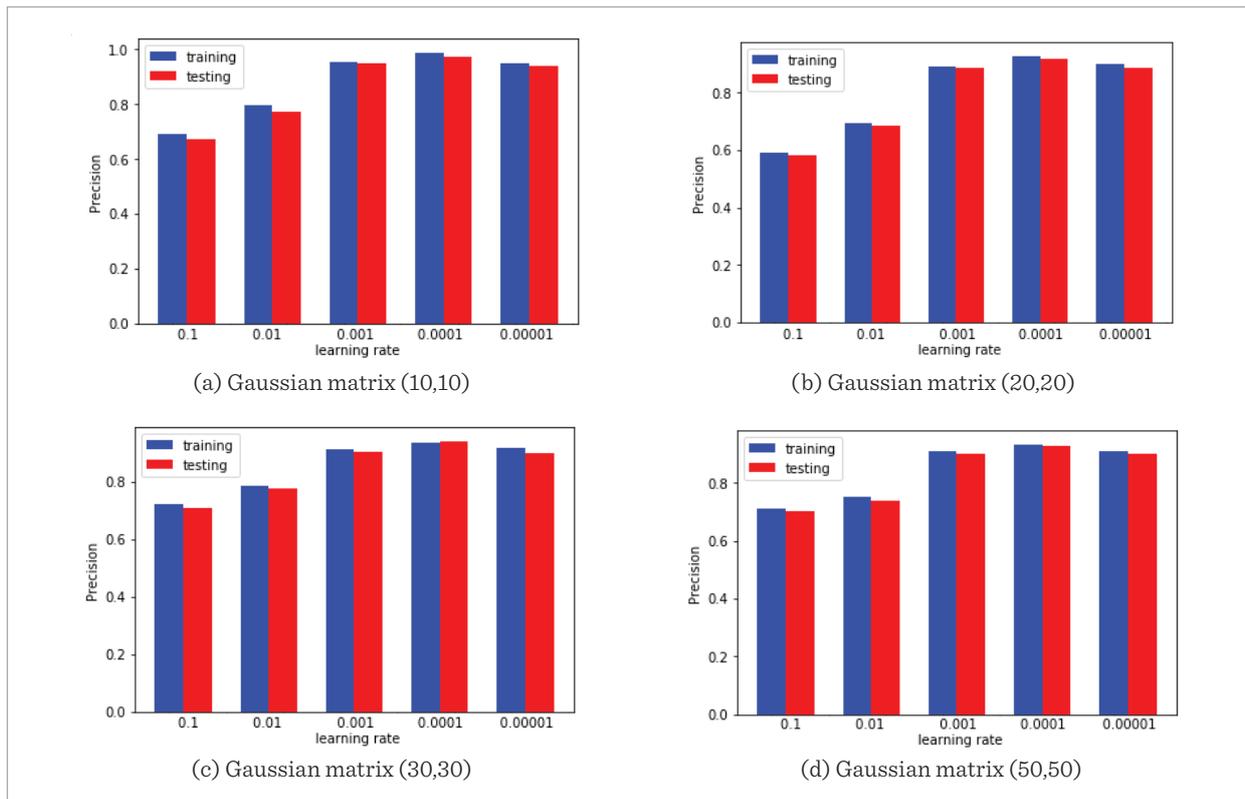
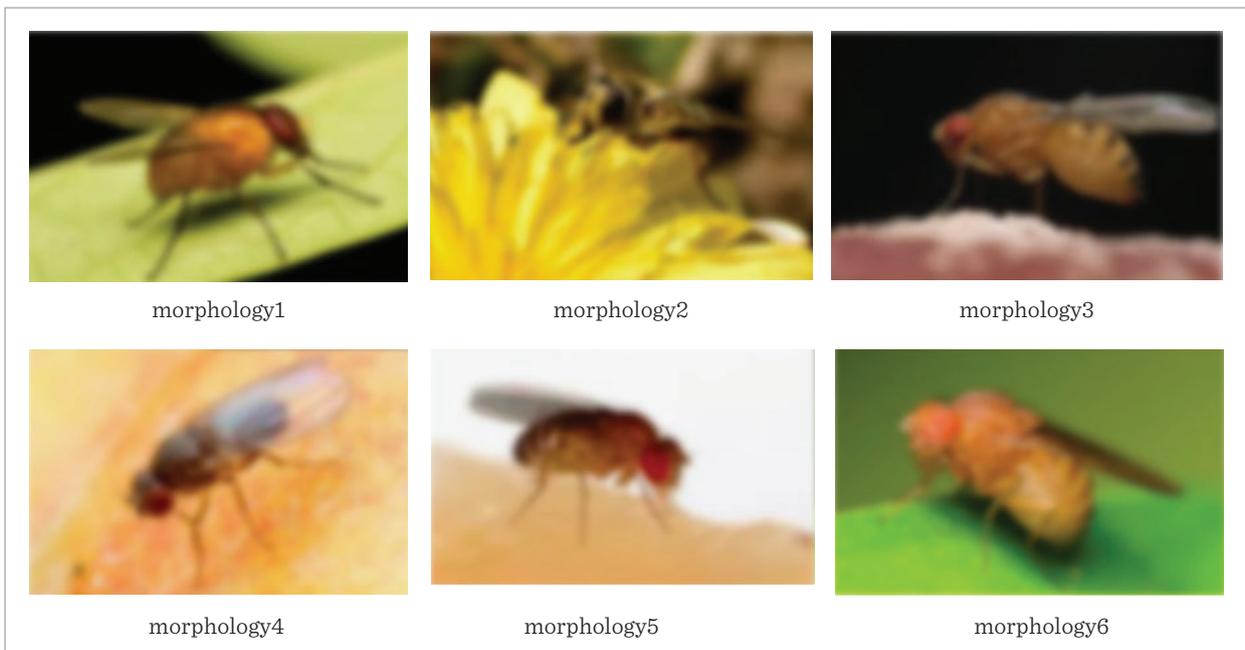


Figure 6

Results using Gaussian blur. The dimension of image is 300×200



4.2. Feature Fusion Validation

The Bilinear layers in the proposed model can fuse the features extracted by the convolution layers, so we need to verify the effectiveness of feature fusion. Based on this, in this subsection, two sets of experiments were implemented, i.e., the first set uses feature fusion in the Bilinear layers, whereas, the second set does not use feature fusion in the Bilinear layers. The designed scheme respecting the effectiveness validation of feature fusion is listed in Table 3.

Table 3

Designed scheme to the effectiveness validation of feature fusion

	Conv 1	Conv 2	Conv 3	Conv 4	Conv 5	Bilinear
First set	√				√	Use
		√			√	Use
			√		√	Use
				√	√	Use
Second set	√				√	Not use
		√			√	Not use
			√		√	Not use
				√	√	Not use

The verified results on feature fusion are presented in Table 4. It can be seen that the best classification results are obtained through using the features extracted by the Conv3 layer and Conv5 layer and fused by the

Table 4

Results on the effectiveness validation of feature fusion

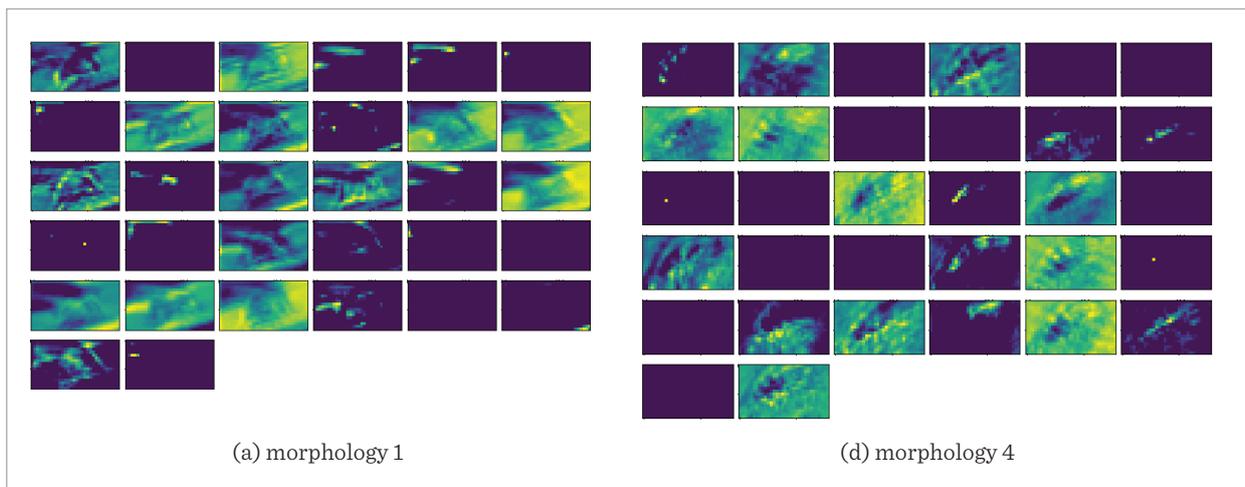
		precision	recall	score
Feature fusion	Conv1+Conv5 +Bilinear	89.22%	88.86%	89.02%
	Conv2+Conv5 +Bilinear	93.17%	93.56%	93.11%
	Conv3+Conv5 +Bilinear	94.55%	95.05%	94.51%
	Conv4+ Conv5+Bilinear	87.30%	87.10%	87.17%
Not fusion	Conv1+ Conv5	84.67%	85.16%	85.88%
	Conv2+ Conv5	86.82%	86.22%	86.98%
	Conv3+ Conv5	88.05%	88.18%	88.07%
	Conv4+ Conv5	81.73%	81.72%	81.68%

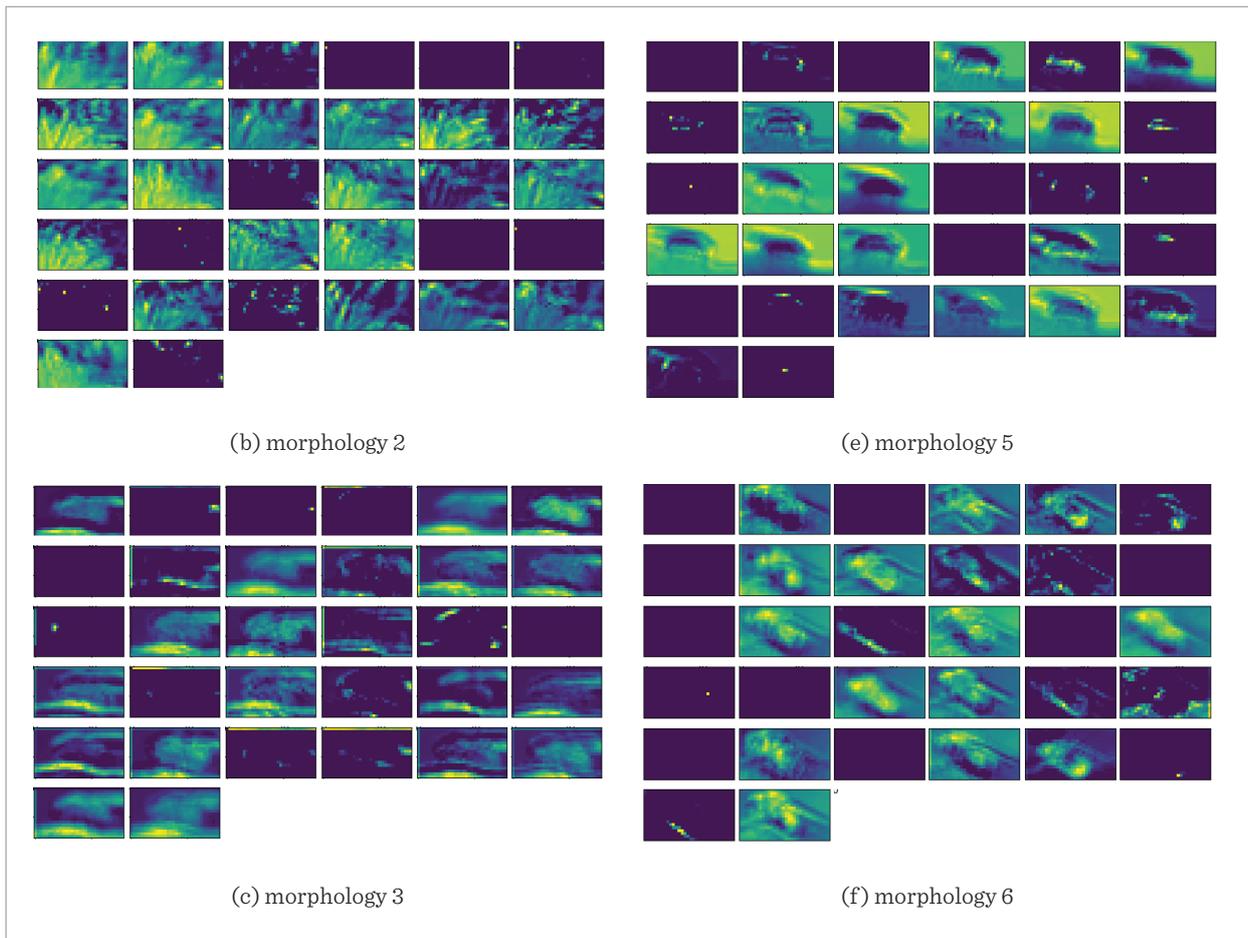
Bilinear layers. Obviously, the results obtained by not using feature fusion are not as good as that of using feature fusion. Together, these results demonstrates that this manner of feature fused can promote the classification ability for the proposed model.

In training process, convolution layers extract different types of features of the image region, and pooling operation fuses these features. As the convolutional operations are superimposed, the deep features obtained from each layer are transitioned to from generalized features, including edges, textures, etc, to high-level semantic representations, e.g., wings, head, and back. The convolution features obtained the fifth layer are visualized in Figure 7.

Figure 7

Visualization of convolution features of the fifth layer



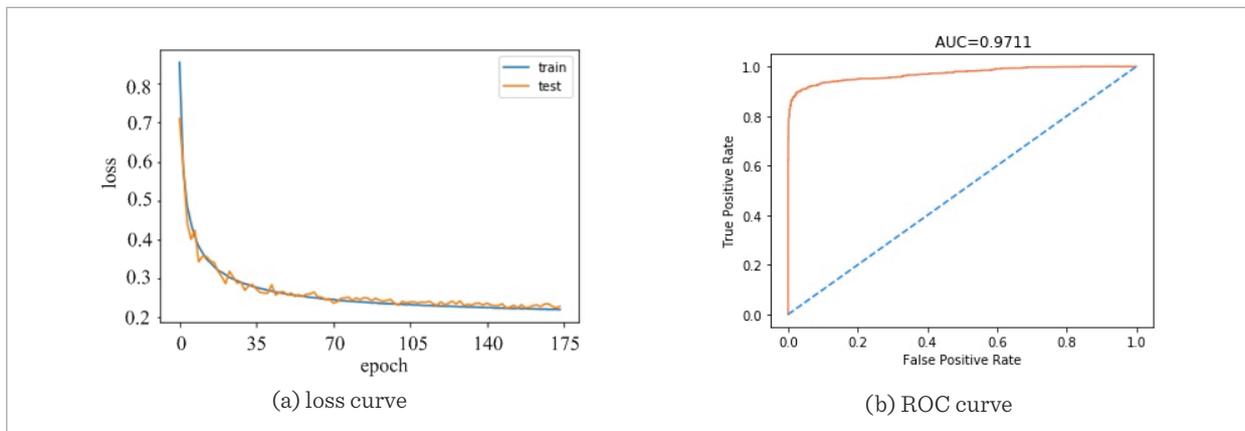


The loss in process of training and testing is presented in Figure 8 (a). It can be seen that the proposed model converged after nearly 175 epochs, and the loss

fell to 0.22. Figure 8 (b) displays the Receiver Operating Characteristic curve (ROC) and the corresponding area under the curve (AUC) on the testing results.

Figure 8

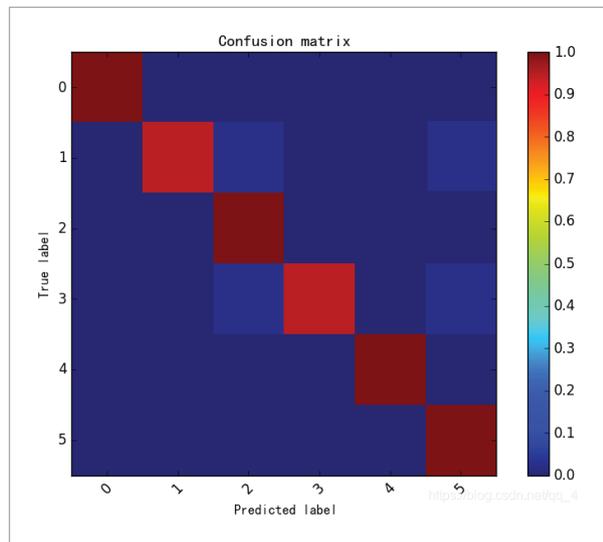
Loss curve and ROC curve



In addition, in order to further analyze the testing results, we also gave the confusion matrix, as shown in Figure 9. The results on the confusion matrix show that the classification accuracy for the image of morphology 2 (i.e., label 1 in Figure 9) and morphology 4 (i.e., label 3 in Figure 9) is lower, while the classification accuracy on the images of other morphology is 1.

Figure 9

The confusion matrix



4.3. Classification Precision Comparison

The results in Table 5 show that our method outperforms the four competitors for the classification results on fruit fly images. It can be observed in Table 5 that the classification accuracy of our method is equal to 97.11% on testing set and 94.87% on validation set, respectively. This implies that the convolutional layers can extract the advanced features from fruit fly images for training on classification tasks. More importantly, this also further demonstrates that the desired classification results can be obtained using the manner of fusing the mid-level and high-level features in the classification of fruit fly images.

From the above multiple experiments, some results can be obtained, as following

- 1 For the classification of images, using the manner of feature fused can achieve a advanced classification accuracy than that of not using feature fused.

Table 5

Precision on testing and validation set. Experiment 100 times independently

Method	Category	Precision	
		Testing	Validation
Our method	6	97.11 ± 1.12%	94.87%
Method in [24]	6	94.11 ± 1.77%	90.55%
Method in [25]	6	95.07 ± 1.55%	90.27%
Method in [23]	6	90.06 ± 2.36%	87.66%
Method in [17]	6	84.64 ± 3.29%	84.66%

- 2 In feature fusion, after performing the matrix dot multiplication operation on features, it is a better choice to perform feature connection than performing feature connection directly.
- 3 Using Gaussian blur to perform images, convolutional neural networks are easy to extract high-level features from images.

5. Conclusion

In this work, to address this issue for the classification of fruit fly images, the CNN of bilinear pooling based on feature fusion was proposed. In our method, Gaussian blur is performed on fruit fly images to reduce the detailed level of fruit fly images, which conveniently extracts high-level features. Then, the features are extracted using the convolutional layers from the fruit fly images processed by Gaussian blur, using the matrix dot multiplication achieves feature fusion in the Bilinear layers. Finally, the softmax layer completes the classification of fruit fly images according to the fused features. Experimental results show that the proposed method is better than the competitors in the classification precision of fruit fly images. We find that the feature fusion of images has a positive effect on promoting the accuracy of image classification. In term of feature fusion, we also indicate that this is a better choice to perform feature connection after performing the matrix dot multiplication operation on features, instead of performing feature connection directly.

These image sets used in this paper is based on different morphology of fruit flies, so the proposed model extracts features according to the morphology of fruit

flies, and then classifies the images of fruit fly. However, as for classifying fruit fly images with natural background, currently, the proposed model is weak in this aspect. Therefore, in future work, we will look at focusing on solving the issue how to effectively classify fruit fly images with natural background.

Data availability

All authors agree with availability of data and material in this work.

Declarations

Competing interests. The authors have no conflicts of interest to declare that are relevant to the content of this article.

Authors' Contributions

Shuping Chen designed research and wrote the paper. Jingjin Chen performed research and analyzed experiment results.

References

1. Ariyan, Z. Improve CAPTCHA's Security Using Gaussian Blur Filter. arXiv:1410.4441, 2014.
2. Cheng, G., Yang, C., Yao, X., Guo, L., Han, J. When Deep Learning Meets Metric Learning: Remote Sensing Image Scene Classification via Learning Discriminative CNNs. *IEEE Transactions on Geoscience and Remote Sensing*, 2018, 56(5), 2811-2821. <https://doi.org/10.1109/TGRS.2017.2783902>
3. Christopher, J. C. B. A Tutorial on Support Vector Machines for Pattern Recognition. *ACM Trans. on Data Mining and Knowledge Discovery*, 1998, 2(2), 121-167.
4. Du, P., Li, E., Xia, J., Samat, A., Bai, X. Feature and Model Level Fusion of Pretrained CNN for Remote Sensing Scene Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019, 12(8), 2600-2611. <https://doi.org/10.1109/JSTARS.2018.2878037>
5. Fang, Y., Li, Z. H., Qin, M. The Potential Economic Impact of the Pumpkin Industry Caused by *Bactrocera Tau* (Walker). *Plant Quarantine*, 2015, 29(3), 28-33.
6. Girshick, R., Donahue, J., Darrell, T., Malik, J. Region-based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016, 38 (1), 142-158. <https://doi.org/10.1109/TPAMI.2015.2437384>
7. He, K., Zhang, X., Ren, S., Sun, J. Delving Deep Into Rectifiers: Surpassing Human-Level Performance on Imagenet Classification. In: *Proceedings of the International Conference on Computer Vision*, 2015, 1026-1034. <https://doi.org/10.1109/ICCV.2015.123>
8. Hoffer, E., Hubara, I., Soudry, D. Train Longer, Generalize Better: Closing the Generalization Gap in Large Batch Training of Neural Networks. *Advances in Neural Information Processing Systems*, 2017, 1731-1741.
9. Hu, F., Xia, G., Yang, W., Zhang, L. Mining Deep Semantic Representations for Scene Classification of High-Resolution Remote Sensing Imagery. *IEEE Transactions on Big Data*, 2019, 6(3), 522-536. <https://doi.org/10.1109/TBDATA.2019.2916880>
10. Huang, Z., Guo, Q. X., Wu, Q. M., Huang, K. H. Morphology. Hazards and China-invading Risk of *Bactrocera Correcta*. *Acta Agricult*, 2014, 26(4), 61-63.
11. Kussul, N., Lavreniuk, M., Skakun, S., Shelestov, A. Deep Learning Classification of Land Cover and Crop Types Using Remote Sensing Data. *IEEE Geoscience and Remote Sensing Letters*, 2017, 14(5), 778-782. <https://doi.org/10.1109/LGRS.2017.2681128>
12. Leonardo, M. M., Avila, S., Zucchi, R. A., Faria, F. A. Mid-level Image Representation for Fruit Fly Identification (Diptera: Tephritidae). In *IEEE Proceedings of 13th International Conference on e-Science (e-Science) (e-Sci.)*, 2017, 202-207.
13. Lin, T.-Y., Chowdhury, A. R., Maji, S. Bilinear CNN Models for Fine-Grained Visual Recognition. *ICCV 2015*, 2015, 1449-1457. <https://doi.org/10.1109/ICCV.2015.170>
14. Lin, T.-Y., Chowdhury, A. R., Maji, S. Bilinear Convolutional Neural Networks for Fine-Grained Visual Recognition. *TPAMI*, 2018, 40(6), 1309-1322. <https://doi.org/10.1109/TPAMI.2017.2723400>
15. Penatti, O. A. B., Silva, F. B., Valle, E., Gouet-Brunet, V., Torres, R. D. S. Visual Word Spatial Arrangement for Image Retrieval and Classification. *ACM Transactions on Pattern Recognition*, 2014, 47(2), 705-720. <https://doi.org/10.1016/j.patcog.2013.08.012>
16. Perre, P., Faria, F. A., Jorge, L. R., Rocha, A., Torres, R. S., Souza-Filho, M. F., Lewinsohn, T. M., Zucchi, R. A. Toward an Automated Identification of *Anastrepha* Fruit

- Flies in the Fraterculus Group (Diptera, Tephritidae). *Neotropical Entomology*, 2016, 45(5), 554-558. <https://doi.org/10.1007/s13744-016-0403-0>
17. Połap, D. Analysis of Skin Marks Through the Use of Intelligent Things. *IEEE Access*, 2019, 7, 149355-149363. <https://doi.org/10.1109/ACCESS.2019.2947354>
 18. Połap, D., Włodarczyk-Sielicka, M., Wawrzyniak, N. Automatic Ship Classification for a Riverside Monitoring System Using a Cascade of Artificial Intelligence Techniques in Cluding Penalties and Rewards. *ISATransactions*, 2021, 1-8. <https://doi.org/10.1016/j.isatra.2021.04.003>
 19. Qu, Y., Li, L., Shen, F., Lu, C., Wu, Y., Xie, Y., Tao, D. C. Joint Hierarchical Category Structure Learning and Large-Scale Image Classification. *IEEE Transactions on Image Processing*, 2017, 99, 1-1.
 20. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In: *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, 779-788. <https://doi.org/10.1109/CVPR.2016.91>
 21. Smith, S. L., Kindermans, P. J., Ying, C., et al. Don't Decay the Learning Rate, Increase the Batch Size. *arXiv:1711.00489*, 2017.
 22. Smith, S. L., Le, Q. V. A Bayesian Perspective on Generalization and Stochastic Gradient Descent. *arXiv:1710.06451*, 2017.
 23. Wan, Z., Yuxiang, Z., Gong, X., Boyang Yu, Z. DenseNet Model with RAdam Optimization Algorithm for Cancer Image Classification. *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, 2021, 15-17. <https://doi.org/10.1109/ICCECE51280.2021.9342268>
 24. Wang, J., Zheng, Y., Wang, M., Shen, Q., Huang, J. Object-Scale Adaptive Convolutional Neural Networks for High-Spatial Resolution Remote Sensing Image Classification. *IEEE Journal of Selection Topics in Applied Earth Observations and Remote Sensing*, 2021, 283-299. <https://doi.org/10.1109/JSTARS.2020.3041859>
 25. Wei, Y., Wei, X., Lin, M., Huang, J. S., Ni, B. B., Dong, J., Zhao, Y., Yan, S. C. HCP: A Flexible CNN Framework for Multi-Label Image Classification. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2015, 38(9), 1901-1907. <https://doi.org/10.1109/TPAMI.2015.2491929>
 26. Zeiler, M. D., Taylor, G. W., Fergus, R. Adaptive Deconvolutional Networks for Mid and High Level Feature Learning. *Proceedings of the IEEE International Conference on Computer Vision*, 2013, 2018-2025.
 27. Zhang, L., Chen, X. L., Hou, X. W., Liu, C. L., Fan, L. M., Wang, X. J. Construction and Testing of Automated Fruit Fly Identification System-Bactrocera Macquart (diptera:Tephritidae). *Acta Entomologica Sinica*, 2011, 54(2), 184-196.
 28. Zhenand, L., Zhongyi, D. Image Recognition Algorithm for Fruit Flies Based on Bp Neural Network. *Trans. Chin. Soc. Agricult. Mach.*, 2017, 48(S1), 129-135.
 29. Zheng, Q., Yang, M., Tian, X., Jiang, J., Wang, D. A full Stage Data Augmentation Method in Deep Convolutional Neural Network for Natural Image Classification. *Discrete Dynamics in Nature and Society*, 2020, 1-11. <https://doi.org/10.1155/2020/4706576>
 30. Zheng, Q., Zhao, P., Li, Y., Wang, H., Yang, Y. Spectrum Interference-Based Two-Level Data Augmentation Method in Deep Learning for Automatic Modulation Classification. *Neural Computing & Applications*, 2020, 33, 7723-7745. <https://doi.org/10.1007/s00521-020-05514-1>
 31. Zheng, Q., Zhao, P., Zhang, D., Wang, H. MR-DCAE: Manifold Regularization-Based Deep Convolutional Auto-encoder for Unauthorized Broadcasting Identification. *International Journal of Intelligent Systems*, 2021, 1-35. <https://doi.org/10.1002/int.22586>
 32. Zheng, Q., Yang, M., Yang, J., Zhang, Q., Zhang, X. Improvement of Generalization Ability of Deep CNN via Implicit Regularization in Two-Stage Training Process. *IEEE Access*, 2018, 6, 15844-15869. <https://doi.org/10.1109/ACCESS.2018.2810849>
 33. Zhongyi, D., Zhen, L., Tiansheng, H. Taxonomy of Fruit Flies Based on Mathematical Morphology. *Journal of Northwest University (Natural Science Edition)*, 2019, 7, 1-7.
 34. Zhou, K., Ming, D., Lv, X., Fang, J., Wang, M. CNN-based Land Cover Classification Combining Stratified Segmentation and Fusion of Point Cloud and Very High-Spatial Resolution Remote Sensing Image Data. *Remote Sensing*, 2019, 11(17). <https://doi.org/10.3390/rs11172065>
 35. Zhou, Q., Zhou, Y., Zhang, L., Li, D. Adaptive Deep Sparse Semantic Modeling Framework for High Spatial Resolution Image Scene Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 2018, 56(10), 6180-6195.

36. Zhou, S., Xue, Z., Du, P. Semisupervised Stacked Autoencoder with Cotraining for hyperspectral Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 2019, 57(6), 3813-3826. <https://doi.org/10.1109/TGRS.2018.2888485>
37. Zhu, X., Tuia, D., Mou, L., Xia, G., Zhang, L., Xu, F. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geoscience and Remote Sensing Magazine*, 2017, 5(4), 8-36. <https://doi.org/10.1109/MGRS.2017.2762307>



This article is an Open Access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 (CC BY 4.0) License (<http://creativecommons.org/licenses/by/4.0/>).