Engineering Applications of Artificial Intelligence 124 (2023) 106563

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

# A two-stream network with complementary feature fusion for pest image classification

Chao Wang [a,b], Jinrui Zhang [a,b], Jin He [a,b], Wei Luo [a,b], Xiaohui Yuan [c,*], Lichuan Gu [a,b,*]

[a] Anhui Agricultural University, No. 130, West Changjiang Road, Hefei, 230036, Anhui, China
[b] Key Laboratory of agricultural electronic commerce of the Ministry of Agriculture, No. 130, West Changjiang Road, Hefei, 230036, Anhui, China
[c] Department of Computer Science and Engineering, University of North Texas, 1155 Union Cir, Denton, 76203, TX, USA

ARTICLE INFO

*Keywords:*
Classification
Long-tailed distribution
Convolutional neural network
Vision Transformer
Fusion

## ABSTRACT

Pests are diverse and the available datasets often contain an uneven number of examples for different pests (a.k.a., the long-tail distribution). This poses a great challenge to learning-based classification methods, especially deep networks, and often leads to degraded performance, especially for the minority (tail) classes. This paper presents a deep learning integration architecture based on decoupling training and fusion learning, which integrates different models with complementary performance on pest datasets with a long-tailed distribution to improve the overall classification performance of pests. A deep neural network is designed that fuses two complementary deep learning models at the feature level, which consists of a convolution neural network (ConvNeXt) and a Swin Transformer model for decoupling training. Experiments are conducted using three datasets (d0, insect, and IP102), and evaluation on accuracy, recall, and F1-Score is reported. For the large-scale pest dataset with long-tailed distribution IP102, the accuracy achieves 76.1%, which outperforms the state-of-the-art methods. In addition, the accuracy for d0 and insect datasets are 98.5% and 92.3%, respectively.

## 1. Introduction

Pests are one of the primary causes of crop losses, which affect a broad range of crops, e.g., rice, wheat, maize, soybeans, sugarcane, etc. According to the Food and Agriculture Organization, pest-related crop losses account for approximately 40% of the world's total crop yield each year, amounting to at least 70 billion USD (IPPC Secretariat, 2021). Integrated pest management requires pest identification and classification in situ (Bollis et al., 2022) to assist decision of appropriate insecticides, as well as implement effective pest control methods to prevent it from the occurrence.

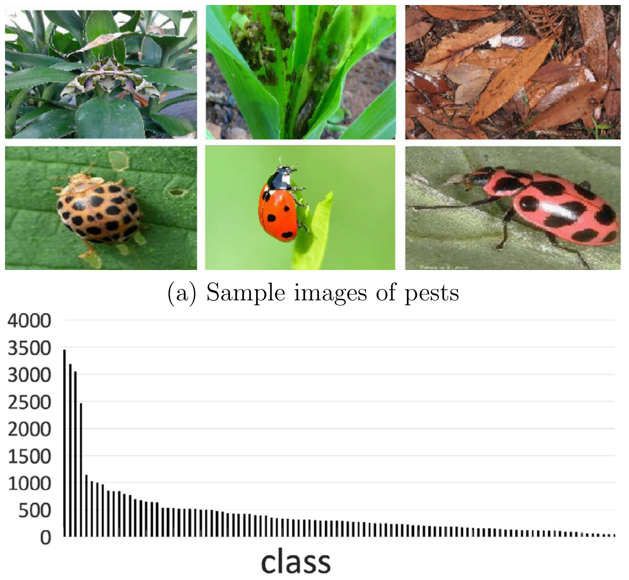
The natural habits and physiological characteristics of pests lead to the relatively similar appearance and low discrimination of some species of pests, as shown in Fig. 1(a). With the development of imaging and computer vision technologies, an automatic solution requires high-quality pest image data for machine learning model development. However, obtaining pest images in the natural environment and annotation is difficult. This leads to a long-tailed distribution (LTD) in pest datasets as shown in Fig. 1(b). When training a model using an LTD dataset, the tail category has little influence on the calculation of the loss due to the small number of examples, which leads to a bias towards the head categories. The performance is, hence, high for the head categories but much lower for the tail categories. In practice, missing some rare, yet deadly pest in the early stage of pest control could end up a disaster in a later stage.

Many pest classification and recognition methods (Setiawan et al., 2022; Liu et al., 2022b) fail to address the LTD issue. A small number of samples for those tail classes poses a challenge in training a fair model and finding a proper decision boundary for the tail classes (Yuan et al., 2018; Yang et al., 2022). The existing solutions are mainly built from two aspects: model and data. Methods (Setiawan et al., 2022; Zhang et al., 2022; Yu et al., 2022) extend the existing models (such as convolution neural networks (CNNs), Vision Transformers, and mask self-supervised learning) by supplementing training data with augmentation and pretraining technology. Other methods (Yang et al., 2021; Sambasivam and Opiyo, 2021) use class rebalancing and information augmentation. LTD is commonly seen in real-world applications, where some classes have much fewer examples due to the less frequent occurrence. One popular strategy is to balance the data for all classes via data augmentation, which, however, could lead to degraded performance when the trained model is applied to the real data.

Recent deep CNN methods use convolutions and pooling operations to extract local features of different scales (Geirhos et al., 2018; Naseer et al., 2021). But the feature scope is limited. An alternative network structure leverages a Transformer that enables the extraction of global

* Corresponding authors.
*E-mail address:* xiaohui.yuan@unt.edu (X. Yuan).

https://doi.org/10.1016/j.engappai.2023.106563
Received 26 September 2022; Received in revised form 15 April 2023; Accepted 29 May 2023
Available online xxxx
0952-1976/© 2023 Elsevier Ltd. All rights reserved.



(a) Sample images of pests

(b) Distribution of the pest images in the IP102 data set.

**Fig. 1.** Sample pest images (a) and their distribution in the IP102 data set (b).

features. The question is if merging these complementary deep features helps improve the classification of pests and overcome the difficulty induced by LTD. This motivates us to explore the integration architecture of feature extraction and fusion to improve the classification performance of the tail classes in LTD pest datasets and hence improve the overall performance of pest classification.

Such a fusion model learns different types of features of pest images with LTD. Designing a network architecture to integrate complementary deep-learning models requires investigation and is the focus of this study. Vision Transformers perform well for the head classes, but the performance degrades significantly for the tail classes, whereas CNNs much better performance for the tail classes. In our evaluation, we also demonstrate the complementarity of inductive bias of CNNs and Vision Transformers. The former learns the local detail features of pest images, while the latter learns the global features. Such a combination of the two models addresses the LTD problem.

This paper aims at improving the classification accuracy of pest images in natural environments by developing a deep fusion network that integrates two backbone networks for extracting and fusing complementary features. The contributions of this paper include:

1. A deep network that fuses complementary features for pest classification. The network employs a training strategy that decouples the training and feature fusion processes to reduce training costs and improve classification performance. This strategy allows us to leverage the backbones with different network architectures to fit the application scenarios.
2. We demonstrate the complementarity of the image features extracted by convolutional networks (e.g., CNNs) and transformer networks (e.g., Vision Transformer) on data sets with LTD.

The rest of this paper is organized as follows: Section 2 reviews the related work. Section 3 describes our proposed fusion-based deep network. Section 4 discusses the experimental results. Section 5 concludes this paper with a summary and future work.

## 2. Related work

Class rebalancing is a common technique for learning from long-tailed training data (Zhang et al., 2021). As the training progresses, if more instances are sampled from a class, the sampling probability of that class will gradually decrease in the subsequent sampling process. Sambasivam and Opiyo (2021) used SMOTE (Chawla et al., 2002) (Synthetic Minority Over-sampling Technique) to solve the problem of unbalanced cassava pest datasets. This approach generates an even distribution of all classes. Alternatively, Yang et al. (2021) simultaneously applied instance-balanced and reversed samplings to generate attention maps and data augmentation is used to solve the LTD problems. The sampling-based methods assume that the real data distribution is mostly even. In cases such as pest control, the number of pests varies over the season, which results in a decline in performance. Because the sampling strategy manipulates data distribution, oversampling introduces limited varieties for model learning (Yang et al., 2022), whereas under-sampling leads to a waste of examples and possible overfitting. In addition, the sampling rate is difficult to decide, making the sampling strategy inapplicable in many real-world applications.

Information augmentation is also often used to deal with imbalanced data, which includes transfer learning and data augmentation (Zhang et al., 2021). Transfer learning leverages pretraining models developed from large-scale datasets and fine-tunes the model using the dataset with an LTD. Peng and Wang (2022) used the ensemble model with pretraining on ImageNet21k (Deng et al., 2009) to classify pests and found that using pretraining parameters can improve the recognition performance of the model compared to directly training on the classification dataset. Mallick et al. (2022) realized the automatic identification and classification of soybean diseases and pests using transfer learning. The data augmentation methods use new data to supplement the dataset to solve the problem of insufficient training data for the model. Nanni et al. (2022, 2020) improved the recognition accuracy of a large-scale pest dataset with an LTD by integrating different data augmentation methods. Kusrini et al. (2020) optimized CNNs in mango pest identification utilizing data augmentation. Setiawan et al. (2022) improved the performance of the lightweight network mobilenet (Howard et al., 2017) on a large-scale pest dataset with an LTD using the Cut Mix (Yun et al., 2019) augmentation and sparse coding. Data augmentation helps learn invariant features. However, without prior knowledge of the data distribution, it is difficult to make intelligent augmentation. Instead, fusing complementary features allows learning invariant features and eliminate the impact of the individual model's induction bias.

To deal with the LTD problem, recent efforts focus on representation learning, classifier design, decoupled training, and ensemble learning (Zhang et al., 2021). CNNs play an important role in the task of pest classification. Wang et al. (2021) improved CNNs by using a capsule network and adding an attention module to optimize the learning ability of the model and realized the fine-grained identification of pests. Wei et al. (2022) used dilated convolution to increase the receptive field of the model and fused the deep and shallow features through skip connections. Yu et al. (2022) used the fruit fly optimization algorithm to optimize the deep learning model for tomato pest identification. There are many lightweight models (Zhou and Su, 2020; Li et al., 2021; Zhang et al., 2022; Chan et al., 2015; Aiadi et al., 2022), some of which are used for the classification and identification of pest data with LTD, and most recent transformer methods (Liu et al., 2021; Touvron et al., 2021) applied to the classification of pest data with LTD. Bollis et al. (2022) used an attention mechanism and activation map to achieve weakly supervised location and classification of pests. Liu et al. (2022b) used VIT masked self-supervised learning (He et al., 2021) (MAE). These methods improve the model's representation power to address the LTD problem.

Although progress has been made with the aforementioned single backbone models, the performance of such a single backbone network with specific inductive biases for the tail classes of LTD data varies greatly. For example, there is a large difference in recognition accuracy between the Vision Transformer head and tail classes, while the CNNs are more balanced in terms of accuracy for each class. To take advantage of models of diverse learning bias, ensemble methods have been proposed. Ayan et al. (2020) used a genetic algorithm to integrate different CNNs for crop pest classification. Khanramaki et al. (2021) designed a voting mechanism for the results of multiple CNN feedback to determine the final pest classification. Xia et al. (2022) used CNNs and Vision Transformer as large-scale and small-scale feature extractors, respectively, and applied voting to achieve the final classification.

Compared with other methods, ensemble-based methods generally obtain better overall classification performance of pest data with an LTD. However, most current ensemble-based methods use the integration interaction of multiple models to collect the features of each location of the LTD, which still falls into the class rebalancing concept. Therefore, this ensemble solution leads to the problem of complex hyperparameter settings and high calculation costs of backbones. A single backbone in the ensemble cannot learn the multitype features of the dataset with an LTD, thus failing to ensure optimal performance. In addition, the accumulated training stage hinders decoupling training integration with the existing well-designed models and makes the ensemble solution unable to replace backbones flexibly.

Given these problems in the current classification methods of pests with LTDs, we propose an ensemble method of deep learning models with complementarity for the classification of pests with LTD, for example, the CNN model with better classification performance in the tail class and the Swin Transformer model with better classification performance in the head classes. In the proposed method, the classification performance of backbones in the ensemble, as well as the inductive bias of backbones, are complementary. The use of backbones in the ensemble is more flexible: they can learn all data samples independently, each achieving its best performance. The proposed method designs a deep neural network to fuse and re-learn the features extracted by backbones, which has a more robust generalization performance and robustness than the ensemble voting mechanism. At the same time, the decoupling design training strategy enables the proposed method to obtain better classification performance while saving computing resources.

## 3. Methodology

### 3.1. System model

The architecture of the proposed FNSTC model is illustrated in Fig. 2(a). It consists of a two-stream network: the backbone part and the fusion module (a multilayer residual block, MIX-Block for short). The backbone part of FNSTC includes two types of backbone models: a CNN model and a Vision Transformer model. As one of the backbones in FNSTC, Swin Transformer is one of the Vision Transformer models, while ConvNeXt (a CNN model) is another backbone in FNSTC. The MIX-Block module fuses the output features from the two backbones for pest classification tasks.

The workflow of FNSTC is presented in Fig. 3. In step 3, the training of the two backbones is conducted independently. There is no interaction between backbones during the training process to achieve the best performance of each backbone. In step 5, the MIX-Block is trained using the output features from the backbones, and there is no direct interaction with the backbones during the training process. Therefore, compared with the joint interactive training method, the decoupling training method adopted in this paper has lower complexity and has certain advantages in saving computing resources.

### 3.2. Backbone networks

Our method employs the ConvNeXt model, which uses a larger convolutional kernel than ResNet. Max pooling is changed to downsampling in two linear layers, with the introduction of nonoverlapping convolutional blocks to enhance the extraction of global information. Another modification is the replacement of batch normalization with layer normalization. A series of the above tuning operations result in a better performance of ConvNeXt in the image classification task.

The Swin Transformer model is used as the complementary backbone, which broke down each image into patches of equal sizes. However, unlike VIT, the Swin Transformer implements the attention mechanism within the windows of patches. These patches are rearranged within the windows, and the window size is progressively increased to obtain global information. Their workflows as backbones of FNSTC are as follows:

1. Separately train ConvNeXt and Swin Transformer using the same training dataset.
2. Delete the last fully connected layer of ConvNeXt and Swin Transformer and take the means for the output eigenvectors of each model.
3. Connect the modified ConvNeXt and Swin Transformer as the backbones to MIX-Block.

### 3.3. MIX-Block

The available methods combining the Transformer model with the CNN model include AlterNet (Park and Kim, 2022), CvT (Wu et al., 2021), Convformer (Wei et al., 2022), and VITDet (Li et al., 2022). However these methods stack the convolution blocks and multi-head Attentions (MSAs), and their training cost is high. The Conformer model (Peng et al., 2021) adopts another combination method, in which many binding structures perform feature conversion between each layer of both backbones. However, these binding structures also imply high training costs and complex expansion. Utilizing the strong representation ability of the multilayer neural network, we design a multilayer residual block (MIX-Block) to fuse the features extracted by both backbones. Compared to some ingeniously designed combination models, MIX-Block has a simpler network structure, and the pretraining model of backbones can be used more conveniently.

MIX-Block has seven layers, including two downsampling layers, with the number of dimensions halved, two feature merging modules, and one classification layer, as shown in Fig. 2(b). Each feature merging module consists of two linear layers with dimensions being invariant. The seven linear layers can be divided into three stages working in turn. In the first stage, MIX-1 contains a downsampling layer, two activation functions, and a feature merging module. In the second stage, MIX-2 has the same construction as MIX-1. The classification results are obtained at the third stage at the classification layer. The downsampling layer in MIX-1 is used to compress the spliced features and filter out the repeated features extracted by the two backbones. The downsampling layer in MIX-2 further compresses and filters the initial fusion features from MIX-1 to reduce the computational cost. Feature merging module in MIX-1 and MIX-2 uses a two-layer linear layer to learn more complex deep features. The fused features are compressed and learned by two modules with the same structure, MIX-1, and MIX-2, which can obtain deeper image feature information. The subsequent experiments also verify the effectiveness and rationality of this design.

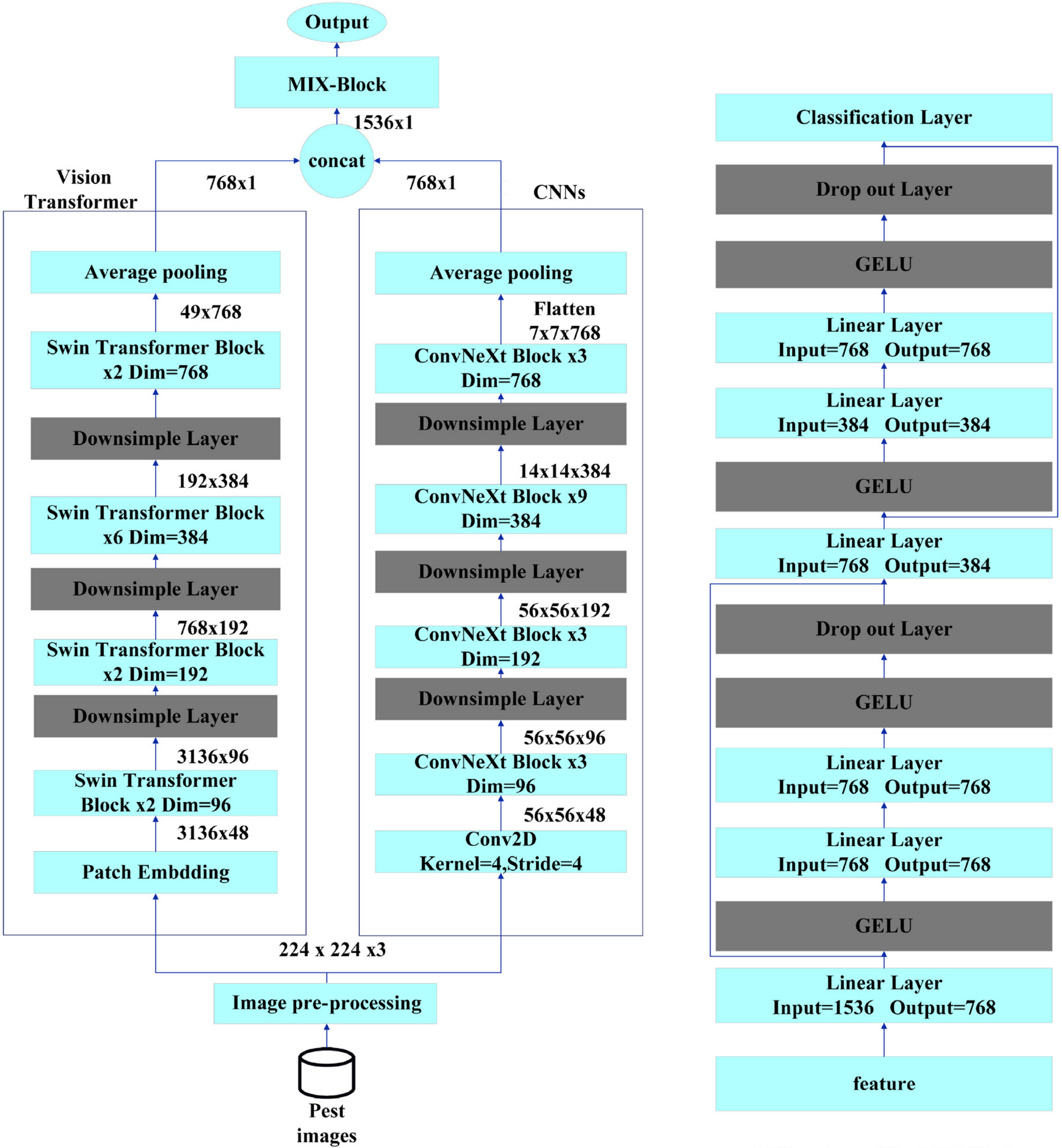
The specific algorithm flowchart and corresponding input and output data are illustrated in Fig. 4. The activation function is used in the construction of MIX-Block. In particular, the skip connection structure is also added to prevent overfitting and improve training efficiency. Introducing skip connections can break the network symmetry and improve the model's overall expression ability (Shang et al., 2016).

The addition of the activation function increases the nonlinear factors of the model and prevents overfitting. MIX-Block uses a Gaussian error linear unit (GELU) (Hendrycks and Gimpel, 2016) as the activation function as follows:

$$GELU\left(x\right) = x\Phi\left(x\right) \tag{1}$$

$$x\Phi\left(x\right) = x \cdot \frac{1}{2}[1 + erf(x/\sqrt{2})] \tag{2}$$

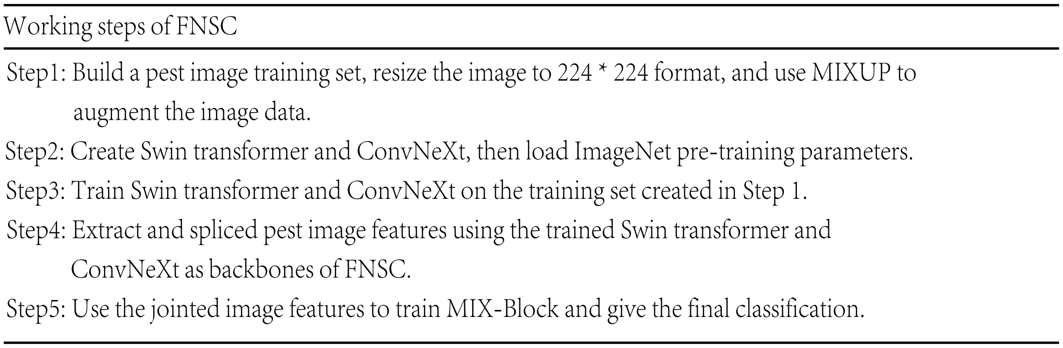
where $\Phi(x)$ is a normal distribution of random parameters.



(a)Structure of the FNSC

(b)Structure of the MIX-Block

**Fig. 2.** The architecture of our proposed network.



Working steps of FNSC

Step1: Build a pest image training set, resize the image to 224 * 224 format, and use MIXUP to augment the image data.

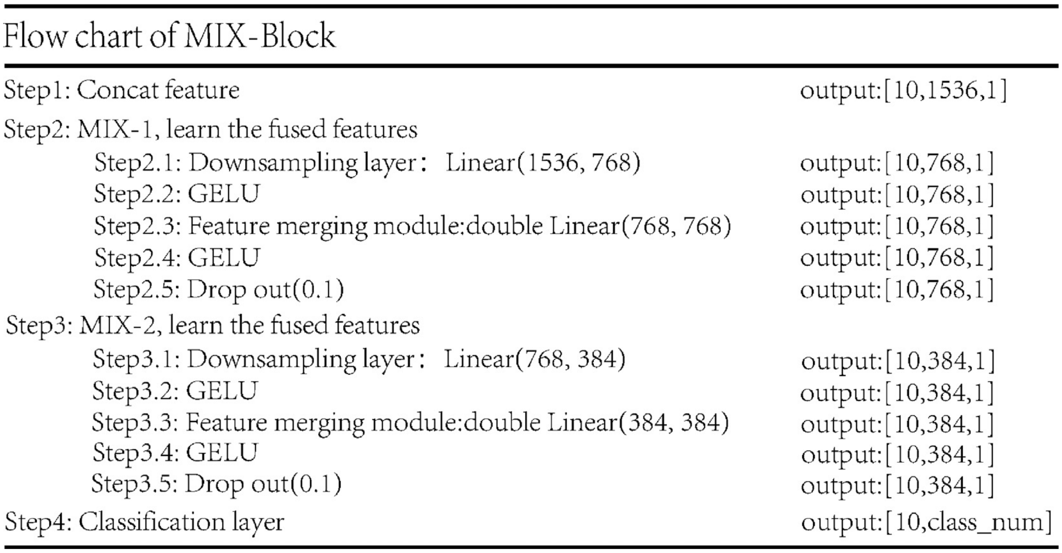
Step2: Create Swin transformer and ConvNeXt, then load ImageNet pre-training parameters.

Step3: Train Swin transformer and ConvNeXt on the training set created in Step 1.

Step4: Extract and spliced pest image features using the trained Swin transformer and ConvNeXt as backbones of FNSC.

Step5: Use the jointed image features to train MIX-Block and give the final classification.

**Fig. 3.** The workflow of FNSTC.



| Flow chart of MIX-Block | |
|---|---|
| Step1: Concat feature | output:[10,1536,1] |
| Step2: MIX-1, learn the fused features | |
| Step2.1: Downsampling layer: Linear(1536, 768) | output:[10,768,1] |
| Step2.2: GELU | output:[10,768,1] |
| Step2.3: Feature merging module:double Linear(768, 768) | output:[10,768,1] |
| Step2.4: GELU | output:[10,768,1] |
| Step2.5: Drop out(0.1) | output:[10,768,1] |
| Step3: MIX-2, learn the fused features | |
| Step3.1: Downsampling layer: Linear(768, 384) | output:[10,384,1] |
| Step3.2: GELU | output:[10,384,1] |
| Step3.3: Feature merging module:double Linear(384, 384) | output:[10,384,1] |
| Step3.4: GELU | output:[10,384,1] |
| Step3.5: Drop out(0.1) | output:[10,384,1] |
| Step4: Classification layer | output:[10,class_num] |

**Fig. 4.** Flow chart of MIX-Block.

After entering the MIX-Block, the fused features go through the calculation process, as shown in Fig. 4. Assuming that after the downsampling layer of MIX-1, the fused feature vector is downsampled and represented as $v$. In the feature merging module of MIX-1, after the conversion of two linear layers, GELU($v$) is represented as $w$. The formal description of output $O_1$ of MIX-1 is as follows:

$$O_1 = GELU(w + v) * M \tag{3}$$

$$M \sim Bernoulli\,(p) \tag{4}$$

where $M$ is a vector of Bernoulli distribution with probability $p$.

MIX-2 and MIX-1 have the same structure and calculation processes. We use Adam (Kingma and Ba, 2014) as the optimizer and SoftTargetCrossEntropy (Hinton et al., 2015) as the loss function to train the proposed model.

## 4. Experiments

### 4.1. Data sets and experimental settings

#### 4.1.1. Data sets

Three public and commonly used pest datasets, namely, IP102, insect, and d0, are used for evaluating FNSTC feasibility. The imbalance degree of the long-tail distribution can be measured by the imbalance ratio (IR), that is, the number of samples in the class with the most samples divided by the number of samples in the class with the fewest samples (Zhu et al., 2020). The larger the IR value, the higher the imbalance degree of a dataset.

IP102 contained images of forest and field pests, the quality of which is uneven, and some are noisy. IP102 is also a large-scale pest dataset with a severe LTD, and its IR value reaches 82. IP102 contains over 75000 pest images belonging to 102 classes, in which the largest class contains 3444 images, and the smallest one includes 42 images. Another challenge of the IP102 dataset is that it contains images of different growth stages of the same pest, including eggs, larvae, pupae, and adults. However, there are differences in pests' morphology and living environment at each stage, especially some pests with metamorphosis. In this study, the IP102 dataset is subdivided into training, test, and validation sets with a 6:3:1 ratio. The image pixels of IP102 are different, and the length and width of pixels are between 150 and 400.

The other two datasets, insect and d0, are used as supplementary validation datasets to prove the effectiveness and applicability of FNSTC. The insect dataset contains 2251 pest images belonging to nine classes, with 1981 images in the training set and 270 images in the test set. To strengthen the LTD, we process the insect dataset by deleting random numbers of images in random classes. The image pixels of insects are different, and the length and width of pixels are between 200 and 350. The IR value of the processed insect dataset is 6. The d0 dataset contains 4500 pest images belonging to 40 classes, subdivided into the training, test, and validation sets at a 7:1:2 ratio. The IR value of d0 is 4.76, with the largest and smallest classes containing 238 and 50 images, respectively. The image pixels of d0 are all 200X200. All the above datasets are in the ImageNet1k format (Deng et al., 2009). All images are cropped to 224 × 224 pixels and normalized. Batch mixup (Zhang et al., 2017) is used as the data augmentation method. The images in the same batch are randomly added in pairs at a ratio of 0.2. Their details are listed in Table 1.

**Table 1**
Datasets used in our evaluation.

| Dataset | Classes | Training | Validation | Test | Total |
|---|---|---|---|---|---|
| IP102 | 102 | 45 095 | 7508 | 22 619 | 75 222 |
| Insect | 9 | 1981 | – | 270 | 2251 |
| d0 | 40 | 3140 | 451 | 909 | 4500 |

#### 4.1.2. Experimental settings

All images used for the experiments had a resolution of 224 × 224. The learning rate is 5 × 10-7 for the first twenty rounds and 5 × 10-4 afterward. The same regularization rules and data augmentation methods are applied to all models. Four NVIDIA RTX 3090 cards are used for computing. The batch size is set to 128. The same original hyperparameter configurations as Swin Transformer and ConvNeXt are used for comparing models.

Table 2 presents the computational complexity (in terms of GFLOPs) and network size (in terms of the number of parameters) of the compared methods, which are ordered based on GFLOPs. The GFLOPs of our proposed FNSTC are at 8.3, which is the median GFLOPs among all methods. Models with fewer parameters tend to have lower computational complexity. Therefore, we compare FNSTC with the models with a similar number of parameters. As shown in Table 2, it can be found that the GFLOPs of models with parameter size of between 50–100M are mostly greater than 10 except for Efficientnet b7, while FNSTC is only 8.3. The number of parameters of our method is 64 which is greater than the median network size. Despite that our network has more than 30 million parameters compared to the median network size among all methods, the computational complexity is moderate.

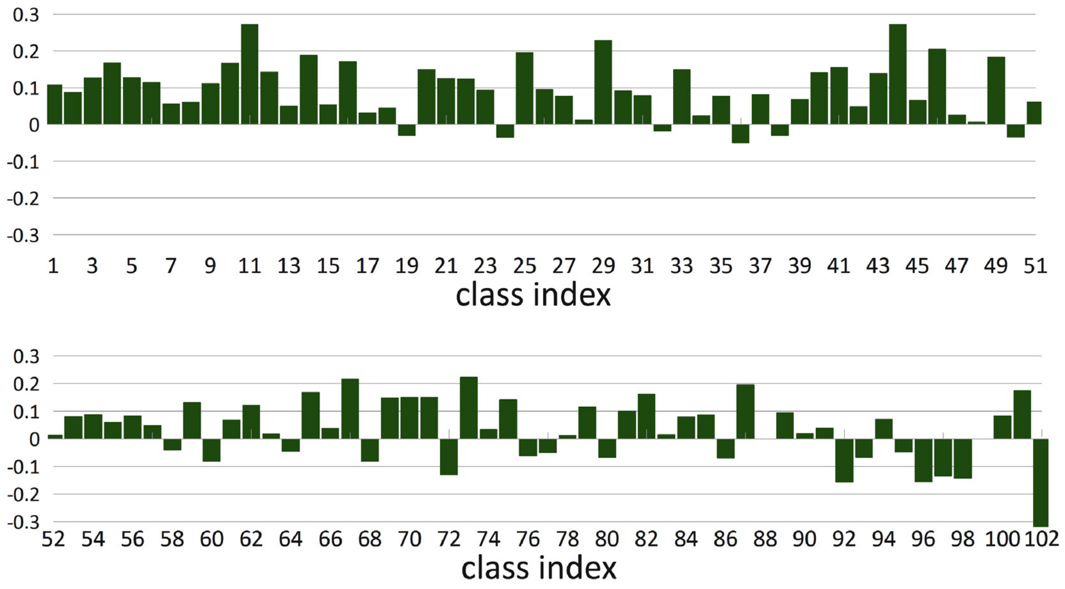


**Fig. 5.** Histogram of differences in recall of all classes.

**Table 2**
Parameters and computational complexity of models under comparison.

| Model &method | of Parameters (Million) | GFLOPs |
|---|---|---|
| mobilenet v3 (Howard et al., 2017) | 4.3 | 0.22 |
| Efficientnet b0 (Tan and Le, 2019) | 4.3 | 0.7 |
| resnet 50 (He et al., 2015) | 25 | 3.9 |
| ConvNeXt tiny (Liu et al., 2022a) | 28 | 4.16 |
| ConvNeXt V2 (Woo et al., 2023) | 28 | 4.5 |
| Swin Transformer tiny (Liu et al., 2021) | 28 | 4.5 |
| Efficientnet b7 (Tan and Le, 2019) | 64 | 4.9 |
| CvT (Wu et al., 2021) | 32 | 7.1 |
| ConvNeXt base (Liu et al., 2022a) | 88 | 8.1 |
| **FNSTC** | **64** | **8.3** |
| conformer (Peng et al., 2021) | 32 | 9 |
| PIM (Chou et al., 2022) | 46 | 9 |
| VIT small (Dosovitskiy et al., 2020) | 49 | 9.22 |
| resnet 152 (He et al., 2015) | 58 | 10.82 |
| densenet 201 (Huang et al., 2017) | 69 | 12.2 |
| VIG (Han et al., 2022) | 88 | 15.4 |
| Swin Transformer base (Liu et al., 2021) | 88 | 15.7 |
| Ensemble (Xia et al., 2022) | > 100 | > 10 |
| VIT base (Dosovitskiy et al., 2020) | 88 | 16.37 |

In Table 2, we list the models and methods included in the comparative experiments, mainly including some vision baselines, recent pest classification methods, and methods specifically oriented to the long-tailed distribution. Compared to the other models, FNSTC requires less computational power under the same parameter quantity.

### 4.2. Verification of complementarity between models

Four groups of experiments are conducted to assess the Swin Transformer and ConvNeXt suitability for extracting different types of features. The complementarity of the two models and the synergetic effect of combining them is demonstrated in terms of the classification results and feature extraction mechanism.

#### 4.2.1. Recall of different classes

As shown in Fig. 5, Swin Transformer and ConvNeXt are applied to recall experiments for each class in IP102. The histogram of the difference in the recall is constructed by subtracting the recall of ConvNeXt from that of the Swin Transformer. In Fig. 5, the x-axis represents the class (in descending order of the number of images belonging to each class), and the y-axis is the difference in accuracy between the Swin Transformer and ConvNeXt for each class.

It can be seen in Fig. 5 that the Swin Transformer had a higher recall than the ConvNeXt transformer overall. Swin Transformer is superior to ConvNeXt in the first 50% of classes, exhibiting no apparent advantages in the last 50% of classes, which samples only accounted for 18.6% of the total data samples. For the last 10% of classes (with samples accounting for 1.4% of the total data samples), the recall of ConvNeXt exceeded that of Swin Transformer. This implies a complementarity between the two models for the learning of small sample classes in the LTD. We further constructed feature heatmaps based on the above classification results for further analysis.

#### 4.2.2. Feature extraction

We performed a detailed analysis of the feature extraction process of the Swin Transformer and ConvNeXt and compared the output features in each layer. As shown by the feature heatmaps in Fig. 6, the Swin Transformer and ConvNeXt differed in the feature extraction process in the following aspects:

First, as shown by the images in the first row, the Swin Transformer focused more on larger objects, while ConvNeXt extracted features in a less focused manner. The same conclusion is confirmed by analyzing the fourth row of images. Second, as shown by the images in the third row, the features extracted by the Swin Transformer are more aggregated along the margins of the pest's body at the beginning, while those extracted by ConvNeXt are scattered all over the pest's body. According to the feature heatmaps, the colors are deeper in aggregated extracted features, indicating that the Swin Transformer had a more robust feature aggregation ability than ConvNeXt. Finally, as shown by the images in the second row, the Swin Transformer aggregated feature maps. However, the feature focus decreased by a large margin after going to the second layer.

After going through the third layer, the Swin Transformer focuses on two parts of the image. The features focused on ConvNeXt in the first layer are equally scattered as those of the Swin Transformer. However, the feature focus did not decrease with ConvNeXt, in contrast to the Swin Transformer as we went deeper. We separately trained the two backbones on IP102 and performed the Fourier transform for the output feature maps of each layer. Fig. 7(a) and (b) show the relative logarithmic amplitude ($\Delta$log amplitude) of the high-frequency components ($1.0\pi$) in the Fourier transformed feature map for the Swin Transformer and ConvNeXt on IP102, respectively. The gray

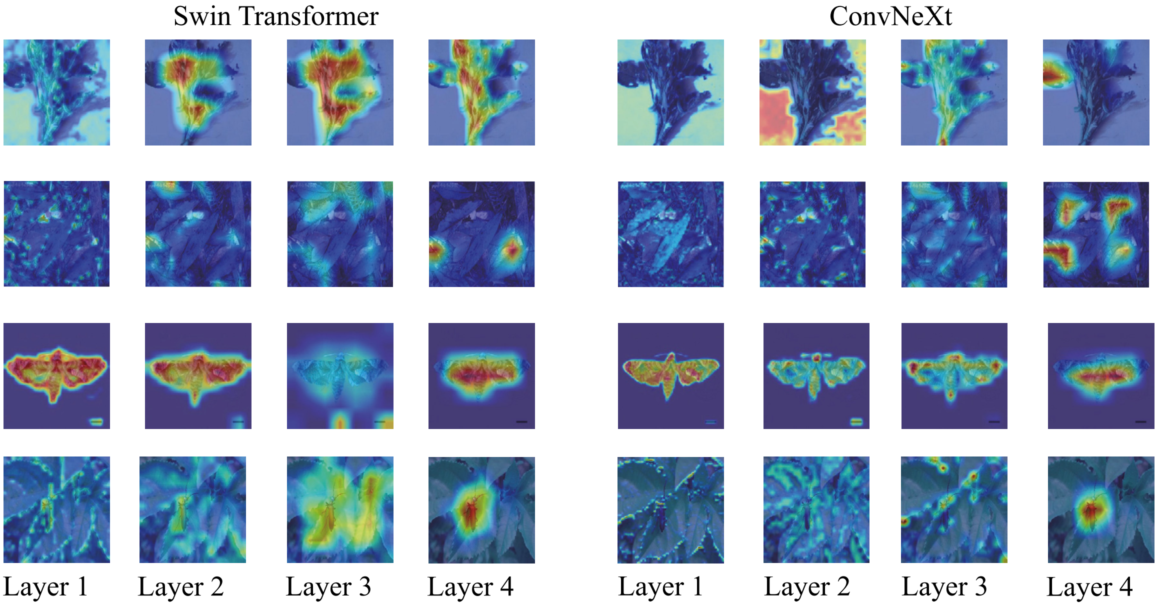


**Fig. 6.** Feature heatmaps of the Swin Transformer and ConvNeXt.

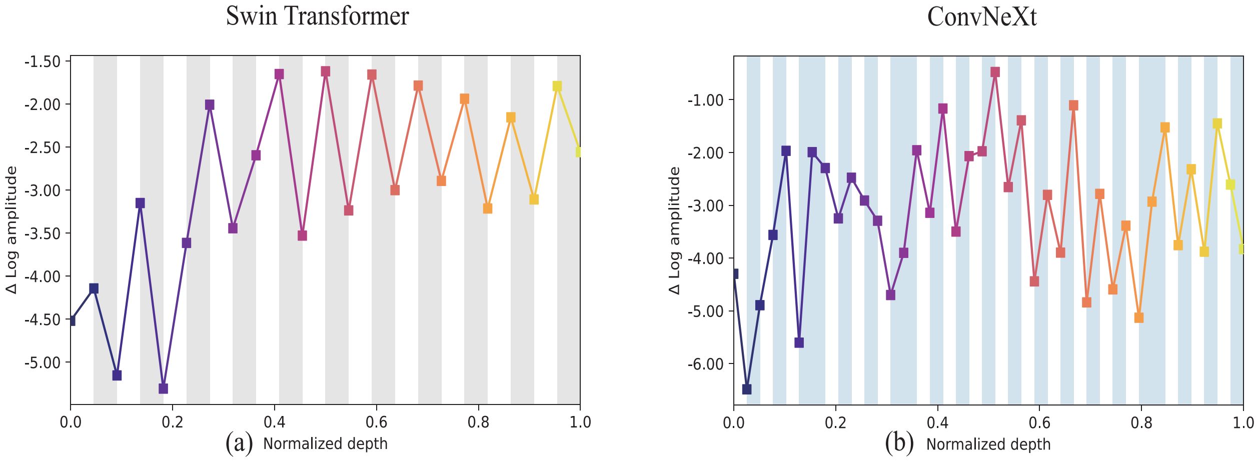


**Fig. 7.** $\Delta$log amplitude for the Fourier transform.

and white areas in Fig. 7(a) are the $\Delta$log amplitudes of Multi-headed Self-attentions (MSAs) and MLPs, respectively, at the high frequency ($1.0\pi$).

In Fig. 7(b), the white areas are the $\Delta$log amplitude of the convolution module at the high frequency ($1.0\pi$), and the blue areas are the $\Delta$log amplitude of the downsampling module at the high frequency ($1.0\pi$). In the left inset, the MSAs of the Swin Transformer (gray areas) usually reduced the high-frequency components of the feature maps, while MLPs (white areas) amplified them. In the right inset, the convolution module (white areas) of ConvNeXt always amplified the high-frequency components. The only exception is at the early stage of the model. At this stage, the Swin Transformer's MSAs shared a working mechanism similar to the convolution module. That is, the MSAs in the Swin Transformer also increased the amplitude.

The convolution module in ConvNeXt always increases the high-frequency amplitude. As shown in Fig. 8, the feature maps from each layer of the network are subjected to a Fourier transform for different frequencies. $\Delta$log amplitude on the ordinate is the difference in $\Delta$log amplitude between the normalized frequency of $0.0\pi$ (center) and $1.0\pi$ (boundaries). The analysis shows that the MSAs reduced the high-frequency components while the convolution module amplified them. In other words, MSAs act as low-pass filters, while the convolution model is a high-pass filter.

We investigate the variance of feature maps between the two models. We measure the variance of the feature mapping for each layer of the two models. Fig. 9(a) depicts the feature variance of ConvNeXt. The white shade highlights the convolution, and the gray shade represents the normalization and the activation of the down-sampling layer. Fig. 9(b) shows the feature variance of the Swin Transformer, where the white shade highlights MLPs and the gray shade shows the MSAs. The variance in the feature map of ConvNeXt is much greater than that of the Swin Transformer. The variance of ConvNeXt increases sharply in the shallow layers but decreases as the network depth increases. The variance of the Swin Transformer, however, is fairly stable. The MSAs of the Swin Transformer tend to reduce the variance, whereas the convolution module of ConvNeXt and MLPs of the Swin Transformer increases the variance. This indicates that the features extracted by ConvNeXt are more scattered and the features extracted by Swin Transformer are more concentrated. Reducing the feature map uncertainty helps optimization by stabilizing the feature maps (Park and Kim, 2021). That is, the features extracted by MSAs and the convolutions are different and largely complementary, which enables diversified features and improved performance for pest classification.

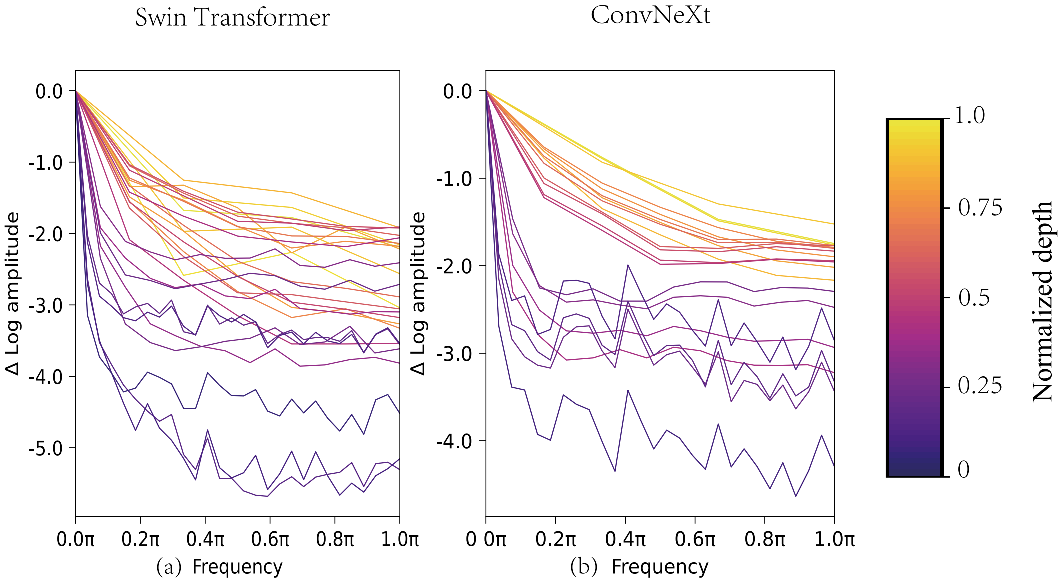


**Fig. 8.** Relative log amplitudes of the Fourier-transformed feature maps.

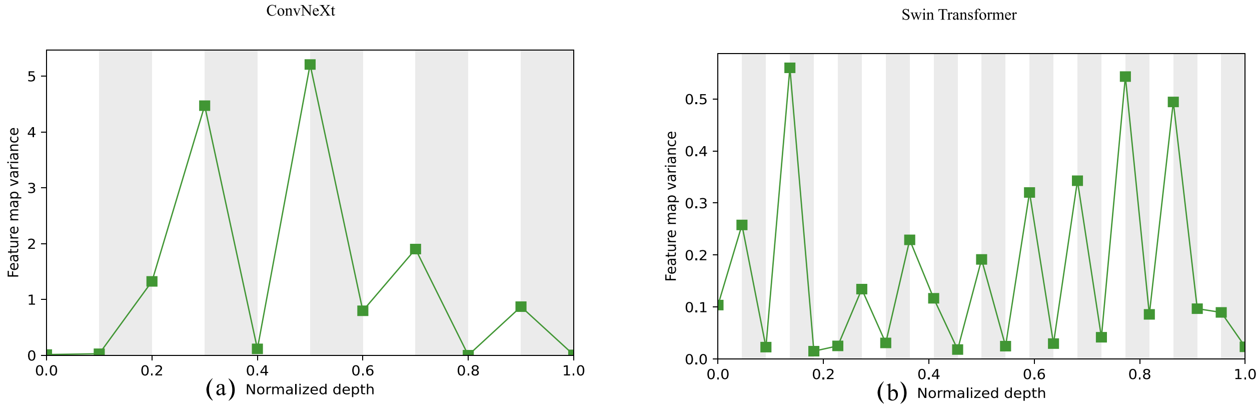


**Fig. 9.** Feature map variance with respect to the normalized depth.

### *4.3. Performance analysis*

We design two sets of comparative experiments to compare the proposed FNSTC with several methods in terms of accuracy, average recall, and F1 score. The evaluation values of these performance metrics are derived from our self-test using interval estimates. The first set of experiments is conducted on IP102. The experiments are divided into two parts. In the first part, all reference models are trained and tested directly on IP102. In the second part, the classification performance of all models is compared based on the pretraining on Imagenet1K (Deng et al., 2009). The second set of experiments is conducted on the d0 and insect datasets further to verify our method's effectiveness.

#### *4.3.1. Evaluation on IP102*

We compare FNSTC with five recent methods for pest image classification (ResNet50 MMM (Zhang et al., 2022), Attention-based MIL-Guided (Bollis et al., 2022), CRN (Yang et al., 2021), MIL-Guided (Bollis et al., 2020) and GAEnsemble (Ayan et al., 2020)) and some baseline models. The backbones of FNSTC are pre-trained on IP102 and then frozen before MIX-Block training. FNSTC is trained for 50 epochs, and the other models are trained for 300 epochs. The experimental results in Table 3 show that FNSTC outperforms other methods in accuracy, recall, and F1-Score, achieving 71.6%, 71.2%, and 0.714, respectively.

Table 4 shows the model classification performance after pre-training on large-scale dataset ImageNet1K (Deng et al., 2009). After the pre-training on ImageNet1K, the classification performance of FNSTC has been further improved to 76.1% on IP102. Compared with the latest pest classification methods MSCD (Liu et al., 2022b), VFL (Setiawan et al., 2022), CTF (Peng and Wang, 2022), the accuracy of FNSTC is 1.3% higher than that of the best method (CTF). Compared with recent integration models Ensemble (Xia et al., 2022), CvT (Wu et al., 2021), PIM (Chou et al., 2022), and conformer (Peng et al., 2021), the accuracy of FNSTC is 2.6% higher than that of the best method (Ensemble).

#### *4.3.2. Evaluation on d0 and insect*

To evaluate the generalization ability of FNSTC, we conduct the same experiments on d0 and insect datasets. FNSTC is trained for 50 epochs, and the other models are trained for 300 epochs. As shown in Tables 5 and 6, (2) the performance measures of FNSTC for insect and d0 datasets are higher than those of the other models. FNSTC respectively archives 98.5%, 98.2%, and 0.984 in the accuracy, recall, and F1 score on d0; the accuracy, recall, and F1 score of insects are 93.1%, 92.9%, and 0.93 respectively.

In the d0 data set, most of the insects belong to Hemiptera, which are hexagonal or elliptical. The morphological differences between these pests are small; however, in the insect data set, the species of insects include worms, moths, locusts, etc., and their morphology is very different. Therefore, for Vision Transformer, which is more inclined to extract shape features, its performance on the insect data

**Table 3**
Classification performance on the ip102 data set.

| Model | Accuracy | Recall | F1-Score |
|---|---|---|---|
| mobilenetv3 (Howard et al., 2017) | 63.2% | 58.7% | 0.609 |
| Efficientnet b0 (Tan and Le, 2019) | 67.5% | 63.1% | 0.652 |
| Efficientnet b7 (Tan and Le, 2019) | 65.7% | 60.9% | 0.644 |
| densenet201 (Huang et al., 2017) | 61.2% | 56.3% | 0.586 |
| resnet50 (He et al., 2015) | 67.0% | 63.0% | 0.649 |
| resnet152 (He et al., 2015) | 67.3% | 63.3% | 0.652 |
| VIT small (Dosovitskiy et al., 2020) | 65.5% | 57.7% | 0.614 |
| Swin Transformer tiny (Liu et al., 2021) | 70.0% | 69.8 | 0.699 |
| Swin Transformer base (Liu et al., 2021) | 70.2% | 69.7% | 0.699 |
| ConvNeXt tiny (Liu et al., 2022a) | 68.6% | 67.5% | 0.680 |
| ConvNeXt base (Liu et al., 2022a) | 68.4% | 67.2% | 0.678 |
| MIL-Guided (Bollis et al., 2020) | 69.5% | – | 0.690 |
| GAEnsemble (Ayan et al., 2020) | 67.1% | 67.1% | 0.658 |
| Attention-based MIL-Guided (Bollis et al., 2022) | 68.3% | – | 0.680 |
| ResNet50-MMM (Zhang et al., 2022) | 56.1% | – | – |
| CRN (Yang et al., 2021) | 70.4% | – | – |
| Swin R | 68.4% | – | – |
| IBloss (Park et al., 2021) | 58.4% | – | – |
| BBN (Zhou et al., 2020) | 62.0% | – | – |
| **FNSTC (proposed method)** | **71.6%** | **71.2%** | **0.714** |

* The accuracy of MIL-Guided and Attention-based MIL-Guided for IP102 is from Bollis et al. (2020) and Bollis et al. (2022), which do not report recall. ResNet50-MMM and CRN for IP102 are from Zhang et al. (2022) and Yang et al. (2021), which do not report recall and F1 Score. The symbol "–" indicates that the respective indicator is not reported in the original literature.

set is better than that on CNNs, while its performance on the d0 data set is the opposite. Because FNSTC can combine the advantages of CNNs and Vision Transformer, its performance on two data sets is better than that of every single backbone.

Experimental results demonstrate that the fusion of features extracted by the Swin Transformer and ConvNeXt as the backbones improve the performance of pest classification. FNSTC fully fused the features extracted by the backbones, thus significantly improving the classification performance of the pest dataset with LTD, especially for the tail classes with a small number of samples.

### *4.3.3. Discussion and analysis*

Compared with the small parameter version, the large parameter version in comparison models has little improvement, and some are even less effective than the small parameter version. For example, the accuracy of EfficientNet b0 is higher than that of EfficientNet b7 by 1.8%. Combining two highly complementary small parameter backbones, the method to improve the performance of the model is more cost-effective than deepening the model and increasing the number of parameters.

**Table 4**
Classification performance with pre-training on imagenet1k for the ip102 data set.

| Model | Accuracy |
|---|---|
| Swin Transformer tiny (Liu et al., 2021) | 75.6% |
| ConvNeXt tiny (Liu et al., 2022a) | 73.8% |
| resnet50 (He et al., 2015) | 72.9% |
| resnet152 (He et al., 2015) | 73.4% |
| Efficientnet b0 (Tan and Le, 2019) | 73.5% |
| ConvNeXt v2 (Woo et al., 2023) | 75.3% |
| CTF (Peng and Wang, 2022) | 74.9% |
| MSCD(MobileNetV3l + Sparse + Cut-Mix + DLR) (Liu et al., 2022b) | 71.3% |
| VFL(VIT base + FRCF + LSMAE + pre train) (Setiawan et al., 2022) | 74.6% |
| Ensemble (Xia et al., 2022) | 74.2% |
| conformer (Peng et al., 2021) | 40.3% |
| VIG (Han et al., 2022) | 66.3% |
| PIM + MIXUP (Chou et al., 2022) | 72.8% |
| PIM (Chou et al., 2022) | 69.8% |
| CvT (Wu et al., 2021) | 66.8% |
| **FNSTC** | **76.1%** |

* The accuracy of resnet50 and resnet152 , Efficientnet b0 for IP102 is from CTF (Peng and Wang, 2022), CTF , Ensemble , MSCD and VFL for IP102 are from Peng and Wang (2022), Xia et al. (2022), Liu et al. (2022b) and Setiawan et al. (2022). The symbol "–" indicates that the respective indicator is not reported in the original literature.

**Table 5**
Performance for the insect data set.

| Model | Accuracy | Recall | F1-Score |
|---|---|---|---|
| mobilenetv3 (Dosovitskiy et al., 2020) | 75.9% | 69.6% | 0.726 |
| Efficientnet b0 (Kusrini et al., 2020) | 83.3% | 78.6% | 0.809 |
| Efficientnet b7 (Kusrini et al., 2020) | 85.5% | 80.6% | 0.830 |
| densenet201 (Nanni et al., 2020) | 85.3% | 80.2% | 0.827 |
| resnet50 (Sambasivam and Opiyo, 2021) | 84.9% | 81.2% | 0.830 |
| resnet152 (Sambasivam and Opiyo, 2021) | 85.2% | 81.1% | 0.831 |
| VIT small (Chawla et al., 2002) | 84.2% | 76.0% | 0.799 |
| Swin Transformer tiny (Yang et al., 2022) | 92.2% | 91.8% | 0.920 |
| Swin Transformer base (Yang et al., 2022) | 92.7% | 92.2% | 0.924 |
| ConvNeXt tiny (Tan and Le, 2019) | 79.8% | 72.2% | 0.758 |
| ConvNeXt base (Tan and Le, 2019) | 82.2% | 78.4% | 0.803 |
| **FNSTC** | **93.1%** | **92.9%** | **0.930** |

The overall performance of the Swin Transformer is better than that of CNNs and VIT. The Swin Transformer achieves a 1.6% improvement compared with the best-performing CNNs. The accuracy of FNSTC exceeds that of the Swin Transformer base by 1.4%, being higher by 3% than that of the best-performing CNN (ConvNeXt).

When compared to the state-of-the-art methods for pest classification, ResNet50-MMM (Zhang et al., 2022), Attention-based MIL-Guided (Bollis et al., 2022), MIL-Guided (Bollis et al., 2020), and GAEnsemble (Ayan et al., 2020), FNSTC outperforms them, its accuracy exceeding the best of them (MIL-Guided) by 2.1%.

The average recall and F1 score are calculated for each model. The results are shown in Table 3. FNSTC had the best average recall and

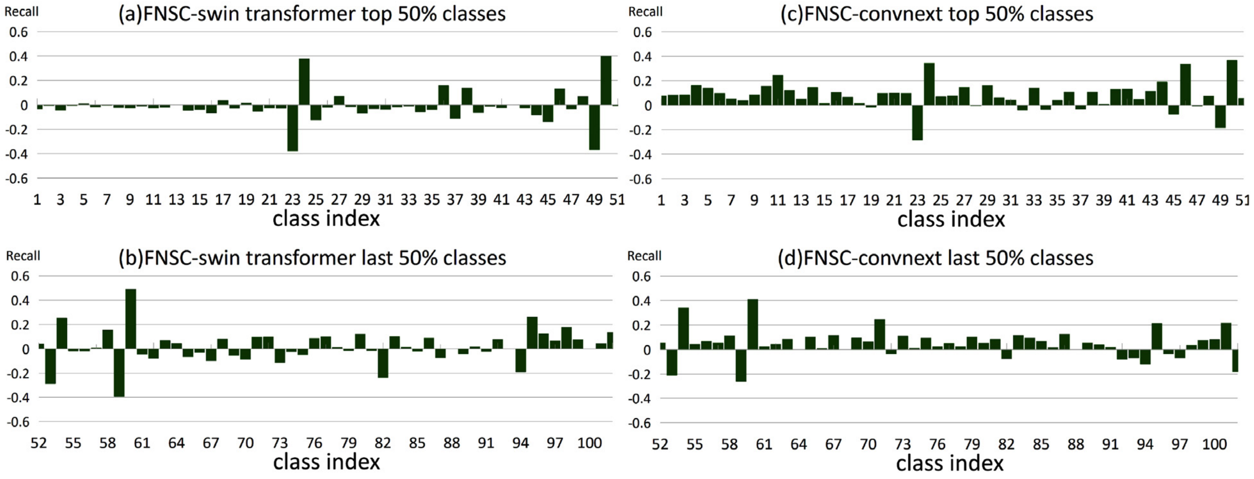


**Fig. 10.** Differences in recall for each class between FNSTC and ConvNeXt, Swin Transfor.

**Table 6**
Performance for the d0 data set.

| Model | Accuracy | Recall | F1-Score |
|---|---|---|---|
| Efficientnet b0 (Kusrini et al., 2020) | 95.1% | 92.7% | 0.939 |
| Efficientnet b7 (Kusrini et al., 2020) | 96.2% | 95.2% | 0.957 |
| densenet201 (Nanni et al., 2020) | 98.0% | 96.4% | 0.972 |
| resnet50 (Sambasivam and Opiyo, 2021) | 97.2% | 94.8% | 0.960 |
| resnet152 (Sambasivam and Opiyo, 2021) | 96.9% | 95.0% | 0.959 |
| VIT small (Chawla et al., 2002) | 95.2% | 93.6% | 0.944 |
| Swin Transformer tiny (Yang et al., 2022) | 97.6% | 96.8% | 0.972 |
| Swin Transformer base (Yang et al., 2022) | 97.7% | 96.8% | 0.973 |
| ConvNeXt tiny (Tan and Le, 2019) | 98.1% | 97.6% | 0.979 |
| ConvNeXt base (Tan and Le, 2019) | 97.1% | 94.0% | 0.955 |
| **FNSTC** | **98.5%** | **98.2%** | **0.984** |

F1 score among all models due to the separate performance optimization of its backbones. Besides, its MIX-Block fuses and re-learns the output features of all backbones, integrating their inductive bias and improving recognition and classification results.

When the pretraining model is used, the respective results are obtained and summarized in Table 4. As the performance of each backbone is improved, that of FNSTC is also enhanced. Compared with FNSTC without pretraining, the accuracy of FNSTC with pretraining increases by 4.5%. The accuracy of the two backbones with pretraining improves by 5.4 and 5.2%, respectively, compared with that of backbones without pretraining. Thus, the overall extent of improvement of FNSTC is lower than that of its integrated backbones. The reason might be that FNSTC fuses the features from its backbones and expands the feature space. However, some features might conflict with each other and negatively impact the classification accuracy of some pest classes.

To further analyze the reasons for the performance improvement of FNSTC, we conduct the following analysis. Fig. 10 shows the histogram of the recall differences between Swin and FNSTC and that between ConvNeXt and FNSTC in each class of IP102. The horizontal coordinate is the class sorted in descending order by the number of samples belonging to each class.

Fig. 10(a) and (b) show the differences in recall for each class between FNSTC and the Swin Transformer, all other parameters being equal. The Swin Transformer, ConvNeXt, and FNSTC differ little in the overall recall distribution for various classes. The highest and lowest recalls of the Swin Transformer for all classes are 97 and 12%, respectively. The highest recall of ConvNeXt is 91%, with the lowest one of 10%. The highest recall of FNSTC is 96%, and the lowest one is 13%.

The above histograms strongly indicate that the proposed model is closer to a better backbone in the overall distribution of recall. For the 50% of head classes, the recall of FNSTC is the same as that of the better-performing backbone. Among them, the recall of some classes has increased or decreased. In general, the increased range exceeded the decline range. For the 50% of tail classes, the recall of FNSTC significantly increased in half of these classes, especially in the last 10%. Therefore, the improvement in the recall of tail classes is the main reason for enhancing overall classification performance. This is because FNSTC learns more features through MIX-Block and fully utilizes the complementarity between backbones.

### 4.4. Ablation study

We conduct two ablation studies: 1. Evaluation of the impact of the number of MIX-Block and inclusion of skip connection; 2. An analysis of the impact of using different backbone networks for feature extraction. In the first set of experiments, MIX-Block is fine-tuned to assess how FNSTC achieves the optimal classification performance. In the second set of experiments, we compare the performance gap between every single backbone and the integration of two backbones using FNSTC and prove that FNSTC combined with CNN and Vision Transformer can effectively improve the classification performance. All experiments are conducted on IP102. The performance measures are accuracy and the epoch for obtaining the best model.

*Experiment 1:* We studied the impact of network depth and residual MLP on performance. Fig. 11 depicts the accuracy of different MIX-Blocks. An accuracy of 71.1% is attained for MIX-Block3 with a three-layer MLP. By increasing the depth of the network, MIX-Block7 contains seven MLP layers and its accuracy achieves 71.38%. However, adding another MLP layer decreases the accuracy to 71.2%. By adding skip connections, the loss surface becomes smoother (Li et al., 2017), which makes the model easier to optimize and reduces the risk of over-fitting. We observe that the accuracy of MIX-Block 7 with skip

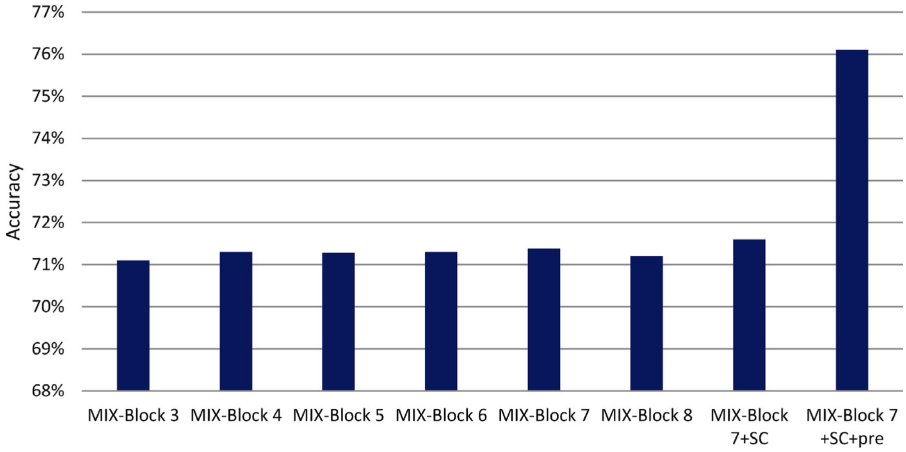


**Fig. 11.** Accuracy of using different MIX-Blocks.

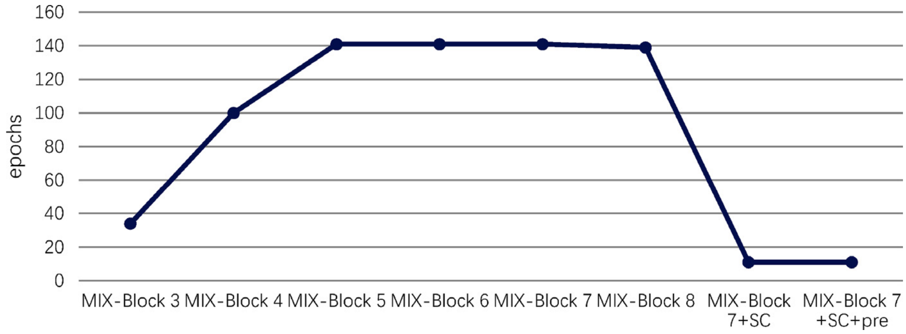


**Fig. 12.** The number of epochs for getting the best results.

**Table 7**
Accuracy of different mix-block designs.

| Model | Accuracy | Epoch |
|---|---|---|
| MIX-Block_3 | 71.10% | 34 |
| MIX-Block_4 | 71.30% | 100 |
| MIX-Block_5 | 71.28% | 141 |
| MIX-Block_6 | 71.30% | 141 |
| MIX-Block_7 | 71.38% | 141 |
| MIX-Block_8 | 71.20% | 139 |
| MIX-Block_7 + Skip connection | 71.60% | 11 |
| MIX-Block_7 + Skip connection + pretrain | 76.10% | 11 |

* Note: MIX-Block_n is a fusion module with n-layer MLP.

**Table 8**
Average accuracy of different backbone combinations.

| Method | Accuracy |
|---|---|
| VIT (Dosovitskiy et al., 2020) | 65.50% |
| Resnet50 (He et al., 2015) | 67.00% |
| Efficientnet b0 (Tan and Le, 2019) | 67.50% |
| ConeNeXt (Liu et al., 2022a) | 68.60% |
| Swin Liu et al. (2021) | 70.00% |
| Resnet50 + VIT | 68.70% |
| Efficientnet b0 + VIT | 69.30% |
| ConvNeXt + VIT | 69.70% |
| Efficientnet b0 + Swin | 70.90% |
| Resnet50 + Swin | 71.30% |
| Swin + ConvNeXt | 71.60% |

connection (i.e., MIX-Block 7+SC) improved to 71.6%. We replaced the backbone in FNSTC with those pre-trained on ImageNet (i.e., MIX-Block 7+SC+pre) and achieved improved classification accuracy at 76.1%. Table 7 reports the average accuracy for all cases.

Table 7 also reports the number of epochs that yield the best results. As shown in Fig. 12 As the network depth is increased, the number of epochs needed for obtaining the best model increases. When the depth is at 6 and above, the number of epochs needed for the best model changes little. This implies a balance is achieved with the complexity of the network and the number of training examples. We added skip connections to the MIX-Block. MIX-Block 7+SC with the residual MLP reduced the epoch for obtaining the best model from 150 to 11.

*Experiment 2:* FNSTC is an ensemble of models. We experimented with different network architectures, including Efficientnet b0, Resnet, ConvNeXt tiny, Swin Transformer, and VIT small, to evaluate the performance and efficiency of the combinations. MIX-Block 7 + skip connect is used. The accuracy of the combinations is reported in Table 8. Fig. 13 depicts a bubble plot of accuracy and GFLOPs. The size of the bubbles represents the number of parameters. As shown in Fig. 13, the number of parameters in the combination method is relatively large, so the GFLOPs of the combination method are higher than that of a single backbone. However, the combination method can achieve higher classification accuracy than a single backbone. MIX-Block effectively fused the features extracted by different backbones. The classification performance after MIX-Block fusion is better than that of a single backbone, demonstrating the effectiveness of the proposed method.

## 5. Conclusion

Our proposed method FNSTC improves the performance of classifying pest images by balancing the recall of tail and head classes of the imbalanced dataset. Using Fourier transform, we illustrate that the features extracted by ConvNeXt focus on the high-frequency components, whereas the features extracted by Swin Transform are mostly

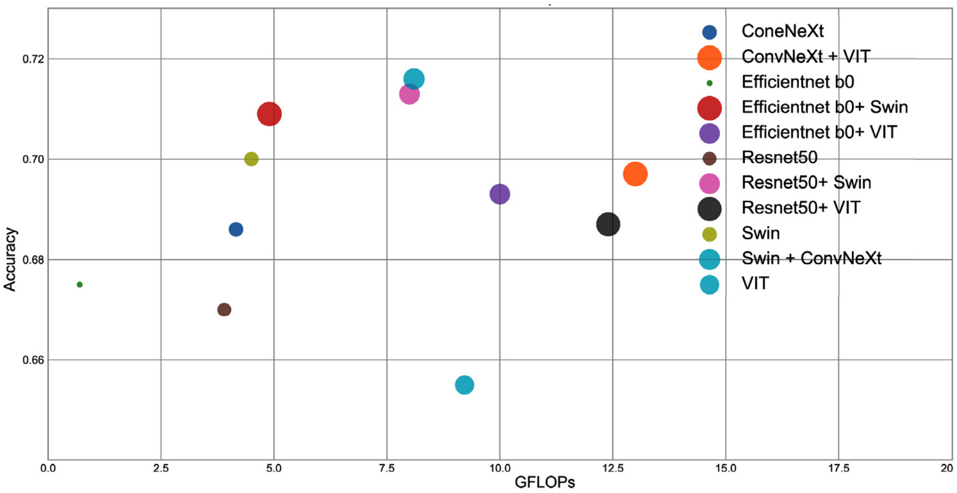


**Fig. 13.** Bubble diagram of GFLOPs vs Accuracy.

in the low-frequency range. That is, the features are complementary, which scaffold the improved performance of insect classification. By integrating different deep networks, we demonstrate that FNSTC achieves improved performance by 1.6% in accuracy. Compared with the state-of-the-art methods, FNSTC achieved the best results in accuracy, recall, and F1 scores at 71.6%, 71.2%, and 0.714, respectively. This is the best result known for the IP102 data set. The accuracy, recall, and F1 score of FNSTC from the other two data sets are also superior. The number of parameters and GFLOPs of FNSTC are 64M and 8.3 respectively, which has advantages over other models with a similar number of parameters. Our method addresses the problems of imbalanced real-world pest image sets. This enables image-based agricultural pest control for increased economic, ecological, and social benefits.

Although the proposed FNSTC network achieved superior performance compared to the state-of-the-art methods, the choice of backbone networks for feature extraction has an unneglectable impact. The key idea is the extraction of complementary features that capture global and local properties for better discriminant for all classes including the ones with fewer examples. In theory, a two-stream network based on distinct backbones helps; yet a careful selection of proper networks is important in practice. In our future work, we plan to explore the analysis of deep features and seek means of quantifying feature significance for the target applications.

## CRediT authorship contribution statement

**Chao Wang:** Conceptualization, Methodology, Writing – original draft, Writing. **Jinrui Zhang:** Software, Experiments, Writing. **Jin He:** Results analysis, Resources. **Wei Luo:** Experimental design, Results analysis. **Xiaohui Yuan:** Conceptualization, Formal analysis, Writing & editing. **Lichuan Gu:** Conceptualization, Validation, Resources, editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (31771679, 31671589), Major Scientific and Technological Projects in Anhui Province, China (201903a06020009), Anhui Natural Science Foundation (2108085MF209), The foundation of Key Laboratory of Agricultural Electronic Commerce, Ministry of Agriculture of China (AEC2021001), Natural Science Research Project of Anhui Provincial Department of Education (KJ2020A0107, KJ2021A1550), The University Synergy Innovation Program of Anhui Province (GXXT-2022-046, GXXT-2022-055, GXXT-2022-040).

## References

Aiadi, Oussama, Khaldi, Belal, Saadeddine, Cheraa, 2022. MDFNet: An unsupervised lightweight network for ear print recognition. J. Ambient Intell. Humaniz. Comput. 1–14.

Ayan, Enes, Erbay, Hasan, Varçın, Fatih, 2020. Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks. Comput. Electron. Agric. 179, 105809.

Bollis, Edson, Maia, Helena, Pedrini, Helio, Avila, Sandra, 2022. Weakly supervised attention-based models using activation maps for citrus mite and insect pest classification. Comput. Electron. Agric. 195, 106839.

Bollis, Edson, Pedrini, Helio, Avila, Sandra, 2020. Weakly supervised learning guided by activation mapping applied to a novel citrus pest benchmark. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 70–71.

Chan, Tsung-Han, Jia, Kui, Gao, Shenghua, Lu, Jiwen, Zeng, Zinan, Ma, Yi, 2015. PCANet: A simple deep learning baseline for image classification? IEEE Trans. Image Process. 24 (12), 5017–5032.

Chawla, Nitesh V., Bowyer, Kevin W., Hall, Lawrence O., Kegelmeyer, W. Philip, 2002. SMOTE: Synthetic minority over-sampling technique. J. Artificial Intelligence Res. 16, 321–357.

Chou, Po-Yung, Lin, Cheng-Hung, Kao, Wen-Chung, 2022. A novel plug-in module for fine-grained visual classification. arXiv e-prints, arXiv:2202.03822.

Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, Fei-Fei, Li, 2009. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Ieee, pp. 248–255.

Dosovitskiy, Alexey, Beyer, Lucas, Kolesnikov, Alexander, Weissenborn, Dirk, Zhai, Xiaohua, Unterthiner, Thomas, Dehghani, Mostafa, Minderer, Matthias, Heigold, Georg, Gelly, Sylvain, Uszkoreit, Jakob, Houlsby, Neil, 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv e-prints, arXiv:2010.11929.

Geirhos, Robert, Rubisch, Patricia, Michaelis, Claudio, Bethge, Matthias, Wichmann, Felix A., Brendel, Wieland, 2018. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv e-prints, arXiv:1811.12231.

Han, Kai, Wang, Yunhe, Guo, Jianyuan, Tang, Yehui, Wu, Enhua, 2022. Vision GNN: An image is worth graph of nodes. arXiv e-prints, arXiv:2206.00272.

He, Kaiming, Chen, Xinlei, Xie, Saining, Li, Yanghao, Dollár, Piotr, Girshick, Ross, 2021. Masked autoencoders are scalable vision learners. arXiv e-prints, arXiv:2111.06377.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, Sun, Jian, 2015. Deep residual learning. Image Recognit. 7.

Hendrycks, Dan, Gimpel, Kevin, 2016. Gaussian error linear units (GELUs). arXiv e-prints, arXiv:1606.08415.

Hinton, Geoffrey, Vinyals, Oriol, Dean, Jeff, 2015. Distilling the knowledge in a neural network. arXiv e-prints, arXiv:1503.02531.

Howard, Andrew G., Zhu, Menglong, Chen, Bo, Kalenichenko, Dmitry, Wang, Weijun, Weyand, Tobias, Andreetto, Marco, Adam, Hartwig, 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

Huang, Gao, Liu, Zhuang, Van Der Maaten, Laurens, Weinberger, Kilian Q., 2017. Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4700–4708.

IPPC Secretariat, 2021. Summary for Policymakers of the Scientific Review of the Impact of Climate Change on Plant Pests: A Global Challenge to Prevent and Mitigate Plant Pest Risks in Agriculture, Forestry and Ecosystems. FAO on behalf of the IPPC.

Khanramaki, Morteza, Asli-Ardeh, Ezzatollah Askari, Kozegar, Ehsan, 2021. Citrus pests classification using an ensemble of deep learning models. Comput. Electron. Agric. 186, 106192.

Kingma, Diederik P., Ba, Jimmy, 2014. Adam: A method for stochastic optimization. arXiv e-prints, arXiv:1412.6980.

Kusrini, Kusrini, Suputa, Suputa, Setyanto, Arief, Agastya, I. Made Artha, Priantoro, Herlambang, Chandramouli, Krishna, Izquierdo, Ebroul, 2020. Data augmentation for automated pest classification in Mango farms. Comput. Electron. Agric. 179, 105842.

Li, Yanghao, Mao, Hanzi, Girshick, Ross, He, Kaiming, 2022. Exploring plain vision transformer backbones for object detection. arXiv e-prints, arXiv:2203.16527.

Li, Yanan, Sun, Ming, Qi, Yang, 2021. Common pests classification based on asymmetric convolution enhance depthwise separable neural network. J. Ambient Intell. Humaniz. Comput. 1–9.

Li, Hao, Xu, Zheng, Taylor, Gavin, Studer, Christoph, Goldstein, Tom, 2017. Visualizing the Loss Landscape of Neural Nets. arXiv e-prints, arXiv:1712.09913.

Liu, Ze, Lin, Yutong, Cao, Yue, Hu, Han, Wei, Yixuan, Zhang, Zheng, Lin, Stephen, Guo, Baining, 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022.

Liu, Zhuang, Mao, Hanzi, Wu, Chao-Yuan, Feichtenhofer, Christoph, Darrell, Trevor, Xie, Saining, 2022a. A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11976–11986.

Liu, Honglin, Zhan, Yongzhao, Xia, Huifen, Mao, Qirong, Tan, Yixin, 2022b. Self-supervised transformer-based pre-training method using latent semantic masking auto-encoder for pest and disease classification. Comput. Electron. Agric. 203, 107448.

Mallick, M.D. Tausif, Biswas, Shrijeet, Das, Amit Kumar, Saha, Himadri Nath, Chakrabarti, Amlan, Deb, Nilanjan, 2022. Deep learning based automated disease detection and pest classification in Indian mung bean. Multimedia Tools Appl. 1–25.

Nanni, Loris, Maguolo, Gianluca, Pancino, Fabio, 2020. Insect pest image detection and recognition based on bio-inspired methods. Ecol. Inform. 57, 101089.

Nanni, Loris, Manfè, Alessandro, Maguolo, Gianluca, Lumini, Alessandra, Brahnam, Sheryl, 2022. High performing ensemble of convolutional neural networks for insect pest image detection. Ecol. Inform. 67.

Naseer, Muhammad Muzammal, Ranasinghe, Kanchana, Khan, Salman H., Hayat, Munawar, Shahbaz Khan, Fahad, Yang, Ming-Hsuan, 2021. Intriguing properties of vision transformers. Adv. Neural Inf. Process. Syst. 34, 23296–23308.

Park, Namuk, Kim, Songkuk, 2021. Blurs behave like ensembles: Spatial smoothings to improve accuracy, uncertainty, and robustness. arXiv e-prints, arXiv:2105.12639.

Park, Namuk, Kim, Songkuk, 2022. How do vision transformers work? arXiv e-prints, arXiv:2202.06709.

Park, Seulki, Lim, Jongin, Jeon, Younghan, Choi, Jin Young, 2021. Influence-balanced loss for imbalanced visual classification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 735–744.

Peng, Zhiliang, Huang, Wei, Gu, Shanzhi, Xie, Lingxi, Wang, Yaowei, Jiao, Jianbin, Ye, Qixiang, 2021. Conformer: Local features coupling global representations for visual recognition. arXiv e-prints, arXiv:2105.03889.

Peng, Yingshu, Wang, Yi, 2022. CNN and transformer framework for insect pest classification. Ecol. Inform. 72, 101846.

Sambasivam, G.A.O.G.D., Opiyo, Geoffrey Duncan, 2021. A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks. Egypt. Inform. J. 22 (1), 27–34.

Setiawan, Adhi, Yudistira, Novanto, Wihandika, Randy Cahya, 2022. Large scale pest classification using efficient convolutional neural network with augmentation and regularizers. Comput. Electron. Agric. 200, 107204.

Shang, Wenling, Sohn, Kihyuk, Almeida, Diogo, Lee, Honglak, 2016. Understanding and improving convolutional neural networks via concatenated rectified linear units. arXiv e-prints, arXiv:1603.05201.

Tan, Mingxing, Le, Quoc, 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. PMLR, pp. 6105–6114.

Touvron, Hugo, Cord, Matthieu, Douze, Matthijs, Massa, Francisco, Sablayrolles, Alexandre, Jégou, Hervé, 2021. Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning. PMLR, pp. 10347–10357.

Wang, Xianfeng, Wang, Xuqi, Huang, Wenzhun, Zhang, Shanwen, 2021. Fine-grained recognition of crop pests based on capsule network with attention mechanism. In: Intelligent Computing Theories and Application: 17th International Conference, ICIC 2021, Shenzhen, China, August 12–15, 2021, Proceedings, Part I 17. Springer, pp. 465–474.

Wei, Depeng, Chen, Jiqing, Luo, Tian, Long, Teng, Wang, Huabin, 2022. Classification of crop pests based on multi-scale feature fusion. Comput. Electron. Agric. 194, 106736.

Wei, Zimian, Pan, Hengyue, Niu, Xin, Li, Dongsheng, 2022. ConvFormer: Closing the gap between CNN and vision transformers. arXiv e-prints, arXiv:2209.07738.

Woo, Sanghyun, Debnath, Shoubhik, Hu, Ronghang, Chen, Xinlei, Liu, Zhuang, Kweon, In So, Xie, Saining, 2023. ConvNeXt V2: Co-designing and scaling ConvNets with masked autoencoders. arXiv e-prints, arXiv:2301.00808.

Wu, Haiping, Xiao, Bin, Codella, Noel, Liu, Mengchen, Dai, Xiyang, Yuan, Lu, Zhang, Lei, 2021. CvT: Introducing convolutions to vision transformers. arXiv e-prints, arXiv:2103.15808.

Xia, Wanshang, Han, Dezhi, Li, Dun, Wu, Zhongdai, Han, Bing, Wang, Junxiang, 2022. An ensemble learning integration of multiple CNN with improved vision transformer models for pest classification. Ann. Appl. Biol. 1–15.

Yang, Guofeng, Chen, Guipeng, Li, Cong, Fu, Jiangfan, Guo, Yang, Liang, Hua, 2021. Convolutional rebalancing network for the classification of large imbalanced rice pest and disease datasets in the field. Front. Plant Sci. 12, 671134.

Yang, Lu, Jiang, He, Song, Qing, Guo, Jun, 2022. A survey on long-tailed visual recognition. Int. J. Comput. Vis. 1–36.

Yu, Helong, Liu, Jiawen, Chen, Chengcheng, Heidari, Ali Asghar, Zhang, Qian, Chen, Huiling, 2022. Optimized deep residual network system for diagnosing tomato pests. Comput. Electron. Agric. 195, 106805.

Yuan, Xiaohui, Xie, Lijun, Abouelenien, Mohamed, 2018. A regularized ensemble framework of deep learning for cancer detection from multi-class, imbalanced training data. Pattern Recognit. 77, 160–172.

Yun, Sangdoo, Han, Dongyoon, Oh, Seong Joon, Chun, Sanghyuk, Choe, Junsuk, Yoo, Youngjoon, 2019. CutMix: Regularization strategy to train strong classifiers with localizable features. arXiv e-prints, arXiv:1905.04899.

Zhang, Hongyi, Cisse, Moustapha, Dauphin, Yann N., Lopez-Paz, David, 2017. mixup: Beyond empirical risk minimization. arXiv e-prints, arXiv:1710.09412.

Zhang, Li, Du, Jianming, Dong, Shifeng, Wang, Fenmei, Xie, Chengjun, Wang, Rujing, 2022. AM-ResNet: Low-energy-consumption addition-multiplication hybrid ResNet for pest recognition. Comput. Electron. Agric. 202, 107357.

Zhang, Yifan, Kang, Bingyi, Hooi, Bryan, Yan, Shuicheng, Feng, Jiashi, 2021. Deep long-tailed learning: A survey. arXiv e-prints, arXiv:2110.04596.

Zhou, Boyan, Cui, Quan, Wei, Xiu-Shen, Chen, Zhao-Min, 2020. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9719–9728.

Zhou, Shi-Yao, Su, Chung-Yen, 2020. Efficient convolutional neural network for pest recognition-ExquisiteNet. In: 2020 IEEE Eurasia Conference on IOT, Communication and Engineering. ECICE, IEEE, pp. 216–219.

Zhu, Rui, Guo, Yiwen, Xue, Jing-Hao, 2020. Adjusting the imbalance ratio by the dimensionality of imbalanced data. Pattern Recognit. Lett. 133, 217–223.