2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

IP102: A Large-Scale Benchmark Dataset for Insect Pest Recognition

Xiaoping Wu[1], Chi Zhan[1], Yu-Kun Lai[2], Ming-Ming Cheng[1], Jufeng Yang[1]*
[1]College of Computer Science, Nankai University, Tianjin, China
[2]School of Computer Science and Informatics, Cardiff University, Cardiff, UK
{xpwu95, chizhan_nt}@163.com, LaiY4@cardiff.ac.uk, {cmm, yangjufeng}@nankai.edu.cn

Abstract

Insect pests are one of the main factors affecting agricultural product yield. Accurate recognition of insect pests facilitates timely preventive measures to avoid economic losses. However, the existing datasets for the visual classification task mainly focus on common objects, e.g.*, flowers and dogs. This limits the application of powerful deep learning technology on specific domains like the agricultural field. In this paper, we collect a large-scale dataset named IP102 for insect pest recognition. Specifically, it contains more than $75,000$ images belonging to 102 categories, which exhibit a natural long-tailed distribution. In addition, we annotate about $19,000$ images with bounding boxes for object detection. The IP102 has a hierarchical taxonomy and the insect pests which mainly affect one specific agricultural product are grouped into the same upper-level category. Furthermore, we perform several baseline experiments on the IP102 dataset, including handcrafted and deep feature based classification methods. Experimental results show that this dataset has the challenges of inter- and intra- class variance and data imbalance. We believe our IP102 will facilitate future research on practical insect pest control, fine-grained visual classification, and imbalanced learning fields. We make the dataset and pre-trained models publicly available at https://github.com/xpwu95/IP102.*

1. Introduction

Insect pests are known to be a major cause of damage to the commercially important agricultural crops [8]. Categorization of insect pests plays a crucial role in agricultural pest forecasting, which is vital for food security and stable agricultural economy [10]. Due to the vast number of pest species and the subtle differences among species, insect pest recognition heavily relies on the professional knowledge of agricultural experts [1], meaning it is expensive and time-consuming. With the development of machine learning and computer vision techniques, automated insect pest recognition attracts increasing research attention.

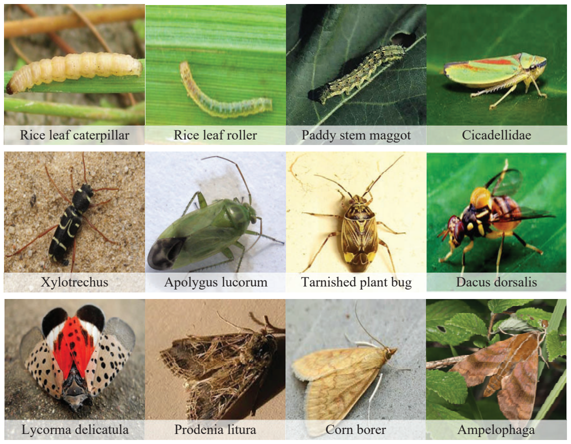

Figure 1. Example images of the IP102 dataset. Each image belongs to a different species of insect pests.

Most of the previous works on insect pest recognition can be described by a traditional machine learning classification framework, which is composed of two modules: (1) feature representation of the insect pest images: a series of handcrafted features including GIST [30], SIFT [25], and SURF [3] *etc*. are adopted to represent the whole image. (2) machine learning classifiers including the support vector machine [4] and the k-nearest neighbor (KNN) classifier. These feature-based methods rely on the careful choice of features. If incomplete or erroneous features are extracted from insect pest images, the subsequent classifier may fail to distinguish similar pest species.

Recently, deep learning enables robust feature learning and achieves state-of-the-art performance on a variety of image classification tasks. It is well known that the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [6] marks the beginning of the rapid development of deep learning, demonstrating that large-scale image datasets play a key role in driving deep learning progress.

*Corresponding author

978-1-7281-3293-8/19/$31.00 ©2019 IEEE
DOI 10.1109/CVPR.2019.00899

However, so far, deep learning methods on insect pest recognition are restricted to small datasets, which only contain very few samples or pest species. Meanwhile, most of the existing insect pest images in public datasets are collected in controlled lab environments, which cannot well satisfy the requirement of insect pest recognition in the real field environment. Moreover, insect pest recognition has its own characteristics different from the existing object or animal classification work [41, 16, 27]. Specifically, different insect pest species may have high appearance similarity and the same species may be in different forms including egg, larva, pupa and adult, *i.e.*, significant intra-class difference and large inter-species similarity.

To advance the insect pest recognition research in computer vision, we introduce the IP102, a new large-scale insect pest dataset in this work. First, we collect more than 300,000 images using common image search engines, which are weakly labeled by the queries. Next, each image is checked by volunteers to make sure it is relevant to insect pests. Agricultural experts then further check and annotate the images with the category label or bounding boxes. The detailed dataset building process is introduced in the following section. Finally, our IP102 dataset covers 102 species of common crop insect pests with over $75,000$ images. Compared with the currently available pest datasets in the literature, the IP102 has a much larger scale, which benefits methods based on deep learning. Our dataset also involves several other features. First, images belonging to the same category may capture different growth forms of the same type of insect pests. Such diversity is unique to the pest datasets but ignored by previous datasets. Besides, the class imbalance is a property of insect pests as some species are much more likely to be observed. Our dataset satisfies the features of imbalanced data distribution, just like in the real world. Fig. 1 shows some examples of the pest dataset.

In order to validate the application value of our proposed dataset, we also report extensive performance for state-of-the-art classification and object detection algorithms. The results indicate that the dataset is challenging and creates new opportunities for research.

Our contributions are summarized as follows:

- To our knowledge, we build the largest scale dataset for insect pest recognition, including insect pest classification and detection. The whole dataset is made available to the research community.

- We conduct extensive experiments on the dataset using CNNs and handcrafted features and establish the performance as the baseline for future research. We also test several state-of-the-art detection models on the detection split of the IP102. We hope this can advance the research on insect pest recognition.

2. Related Work

In this section, we introduce the related work of insect pest recognition methods and review the existing datasets.

2.1. Insect Pest Recognition

Early insect pest recognition is helpful for pest control and improving the quality and yield of agricultural products [35]. In recent years, many computer-aided insect pest recognition systems [32, 2] are presented in the vision community. We group them into two types: the handcrafted and deep feature based methods.

Handcrafted features such as SIFT [25], HOG [5] *etc.* perform well on the low-level feature representations (*e.g.*, color, edge, and texture). In the early years, handcrafted feature based methods are the primary solutions for insect pest recognition. Samanta *et al.* [35] utilize correlation-based feature selection and artificial neural networks to diagnose 8 tea insect pests based on a dataset containing 609 samples. In [28, 32], an SVM classifier is applied to identify whiteflies, aphids, and thrips in leaf images. These methods tend to extract several typical handcrafted features to represent the insect pest and then evaluate on the small datasets with few categories. However, there are a large number of insect pest categories in real life. It is inefficient and time-consuming to design feature extractors for recognizing diverse insect pests. In addition, handcrafted features lack the representation ability for high-level semantic information.

Recently, deep learning technology widely attracts the attention of researchers [18, 34, 24]. Deep convolutional neural networks (CNNs) such as GoogleNet [39] and ResNet [13] show excellent performance in the image classification task. There are also several works [23, 2] that successfully apply CNNs to solve the problem of insect pest recognition. Liu *et al.* [23] classify paddy field pests via training a deep CNN, and their dataset comprises approximately $5,000$ training samples for 12 classes. Alfarisy *et al.* [2] also use the CaffeNet [14] for paddy pest classification. In addition, [7] achieves comparable results to the deep CNN (*i.e.*, VGGNet [36]) based on bio-inspired methods. Yet the evaluated dataset is small containing just 563 samples. Overall, these deep feature based works lack enough samples for optimizing the massive hyper-parameters of CNNs. In order to promote further scientific research and practical applications, we should address the issues of limited categories and samples. Hence we collect the large-scale IP102 dataset, which contains 102 categories of insect pests with $75,222$ samples.

2.2. Related Datasets

Some of the small datasets related to insect pest recognition are released, such as [35, 42, 7]. Most of them typically contain fewer than $1,000$ samples. For example, [40] collects a dataset only consists of 200 samples in 20 classes for paddy field insect pest classification. Subsequently, several larger datasets are presented. Xie *et al.* [44] present a dataset which contains $1,440$ samples and 24 common insect pests of field crops. Yet on average it only has 60 samples per class, which is also hard to train a CNN. To tackle this problem, [23, 43, 2] propose some datasets which contain more than $4,500$ samples in total and 100 samples for each class. However, only the dataset of [43] is available so far. Besides, the background, object pose of the same class of pest images in this dataset [43] are highly similar, making it difficult to cope with the complexities of real-life scenes. Table 2 illustrates the details of these related datasets. In contrast, our purposed IP102 covers 102 common insect pest species in practical applications and is built in the wild. Besides, the IP102 dataset has $75,222$ images and an average size of 737 samples per class.

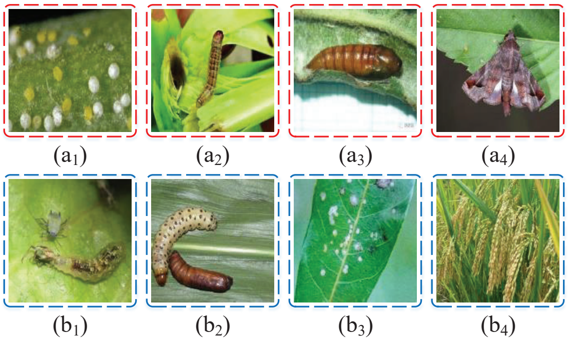

Figure 2. Different forms of insect pest images. The red dashed boxes denote different forms of pests, containing (a_1) egg, (a_2) larva, (a_3) pupa, and (a_4) adult, which belong to the same *sub-class*. The images surrounded by blue dashed box are dropped because there are no or more than one insect pest category.

3. Our Insect Pest Dataset

3.1. Data Collection & Annotation

We collect and annotate the IP102 dataset with following four stages: 1) taxonomic system establishment, 2) image collection, 3) preliminary data filtering, and 4) professional data annotation.

3.1.1 Taxonomic System Establishment

We establish a hierarchical taxonomic system for the IP102 dataset. We invite several agricultural experts and discuss the common categories of insect pests which exist in daily life. There are 102 classes finally obtained and they present a hierarchical structure as shown in Fig 4. Each insect pest is assigned an upper-level class (denoted as *super-class* in the following) based on the crop that suffers from the pest. In other words, each insect pest is a subordinate class (denoted as *sub-class* in the following) of a certain *super-class*. For example, the pest of paddy stem maggot spoils the crop of rice, and the rice belongs to the field crop. Hence, in the taxonomic system of the IP102, the *sub-class* of paddy stem maggot has the *super-class* of rice and field crop. The detailed structure of the IP102 dataset is introduced in the following dataset structure subsection 3.3.

3.1.2 Image Collection

We utilize the Internet as the primary source to collect images, which is widely used to build datasets such as the ImageNet [6] and the Microsoft COCO [21]. The first collection step relies on common image search engines, including Google, Flickr, and Bing *etc*. We use the English name and corresponding synonyms of each *sub-class* as the query keywords. Only top-$2,000$ results are kept for each keyword. Then we search from several professional agriculture and insect science websites. In addition to the image form, we also collect video clips which contain the content of insect pests. From the video clips, we capture images at 5 frames per second. As a consequence, we collect more than $300,000$ candidate images for the IP102 dataset.

3.1.3 Preliminary Data Filtering

We organize 6 volunteers to manually filter the candidate images. Before data filtering, they receive three parts of training content, *i.e.*, 1) the common sense of insect pests from agricultural experts, 2) the taxonomic system of the IP102, and 3) different forms of insect pests. For example, Fig. 2 shows four forms of insect pests, containing egg, larva, pupa, and adult. Even they are at different stages of the life cycle, yet all of them can cause varying degrees of damage to agricultural products. At the process of preliminary data filtering, volunteers delete the images which contain none or more than one insect pest category as illustrated in Fig 2. Then, we convert the format of filtered images to JPEG and delete the images which are repeated or damaged. Finally, we have about $120,000$ images with weak labels of query keywords. The label of *super-class* is assigned according to the taxonomic system of the IP102 dataset.

3.1.4 Professional Data Annotation

Data annotation by agricultural experts is the most important procedure. In the taxonomic system of the IP102, there are 8 kinds of crops damaged by insect pests. For each crop, we invite a corresponding agricultural expert who studies it primarily. Therefore, in total, we invite 8 agricultural experts to annotate the images filtered in the previous procedure. We build a Question/Answer (Q/A) system for convenient annotation. For the image shown on the interface of the Q/A system, experts need to answer which category the image belongs. The professional data annotation comprises

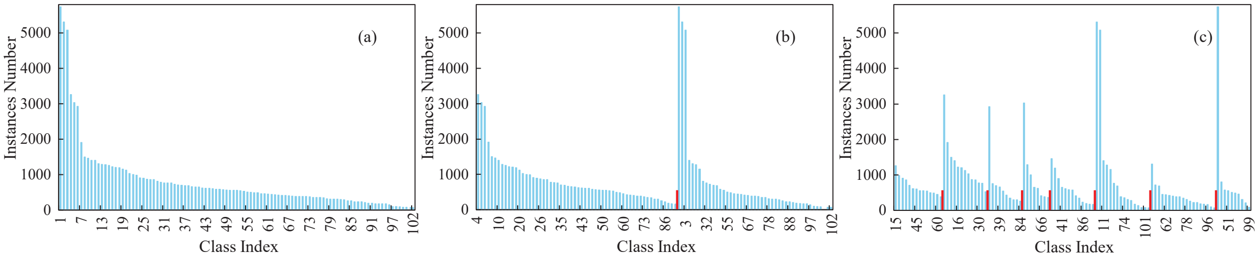

Figure 3. Sample number distribution of the IP102 dataset in different levels. The red calibration tails split 2 *super-classes* in the sub-figure (b) and 8 *super-classes* in the sub-figure (c), respectively.

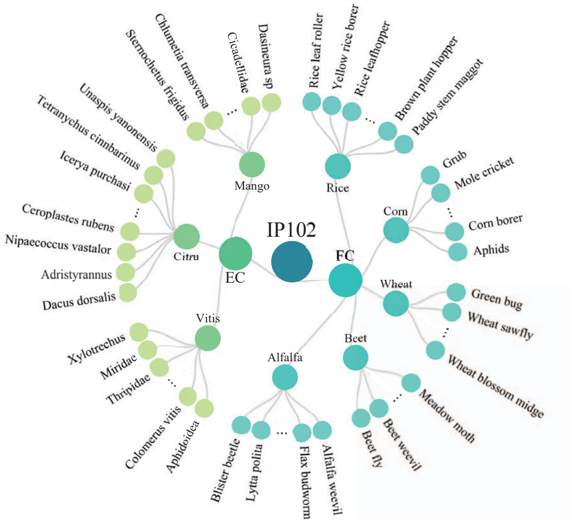

Figure 4. Taxonomy of the IP102 dataset. The ‘FC’ and ‘EC’ denote the field and economic crops, respectively. On the *sub-class* level, only 35 classes are shown. The full list of each *sub-class* can be found in the released IP102 dataset.

Table 1. Training/validation/testing (denoted as Train/Val/Test) set split and imbalance ratio (IR) of the IP102 dataset on different class levels. The ‘Class’ indicates the *sub-class* number of the corresponding *super-class*. The ‘FC’ and ‘EC’ denote the field and economic crops, respectively.

Super-Class		Class	Train	Val	Test	IR
FC	Rice	14	5,043	843	2,531	6.4
	Corn	13	8,404	1,399	4,212	27.9
	Wheat	9	2,048	340	1,030	5.2
	Beet	8	2,649	441	1,330	15.4
	Alfalfa	13	6,230	1,037	3,123	10.7
EC	Vitis	16	10,525	1,752	5,274	74.8
	Citrus	19	4,356	725	2,192	17.6
	Mango	10	5,840	971	2,927	61.7
IP102	FC	57	24,602	4,098	12,341	39.4
	EC	45	20,721	3,448	10,393	80.8
IP102		102	45,095	7,508	22,619	80.8

of independent and synergistic annotations. At the phase of independent annotation. Each agricultural expert is responsible for annotating only one kind of crop *super-class*. For example, for the expert who studies the rice primarily, he needs to annotate these images with the *super-class* of rice. In this case, the expert has 15 options for category selection in the Q/A system. These options consist of 14 insect pest classes which mainly damage the rice crop and an “other class” option. The “other class” means that the image does not belong to the 14 insect pest classes of concern or contains none or more than one insect pest class. The next phase is the synergistic annotation. There are fixed 103 (*i.e.*, 102 insect pest classes plus 1 “other class”) category options in the Q/A system for each expert. Besides, these 8 experts synergistically annotate the “other class” images from the last independent annotation phase. For an image, each expert needs to annotate it, *i.e.*, choose one of the 103 options. The final annotation results follow a strict criterion: one image belongs to a category only when it is agreed by more than 5 experts, otherwise it will be deleted.

The detection of pest locations in images is also very important. It can help agricultural experts or users better find the specific location of pests (especially those that are not obvious in the image). In addition, the real-world scenario makes it complex to recognize the insect pests. A cluttered background can misguide the classifier when the target pest is not salient, and the existence of multiple samples of pests in the image demands respective recognition. The pest control measures in the scene need accurate pest location and category of each pest. Therefore, effective pest insect detection can alleviate the complexity of realistic scenario by sample-aware recognition with spatial information. It can also boost classification performance by removing irrelevant background features. Considering the difficulty and cost of labeling the bounding box, we randomly select part of images from each class to form a subset for the object de-

Table 2. Comparison with existing datasets related to insect pests. The 'Class' denotes the class number. The 'Avail' indicates if the dataset is available. The 'Y' and 'N' denote 'yes' and 'no', respectively. The 'Avg' denotes average numbers of samples per class.

Dataset	Year	Class	Avail	Sample	Avg
Samanta *et al*. [35]	2012	8	N	609	76
Wang *et al*. [42]	2012	9	Y	225	25
Venugoban *et al*. [40]	2014	20	N	200	10
Xie *et al*. [44]	2015	24	Y	1,440	60
Liu *et al*. [23]	2016	12	N	5,136	428
Xie *et al*. [43]	2018	40	Y	4,500	113
Deng *et al*. [7]	2018	10	Y	563	56
Alfarisy *et al*. [2]	2018	13	N	4,511	347
IP102	2019	102	Y	75,222	737

tection task. The experts label the bounding boxes of insect pests following the format of Pascal VOC [9].

3.2. Dataset Split

The IP102 dataset contains $75,222$ images and 102 classes of insect pests, yet the smallest category only has 71 samples. For more reliable test results on the IP102, there should be enough samples of each category on the testing set. Hence we follow a roughly $6:1:3$ split. The training, validation, and testing sets are split at *sub-class* level. Specifically, the IP102 is split into $45,095$ training, $7,508$ validation, and $22,619$ testing images for classification task. Detailed splits at different levels are shown in Table 1. Corresponding image lists for each set are released in the IP102 dataset. For the task of object detection, there are totally $18,983$ annotated images. We split those images containing bounding box annotations into $15,178$ and $3,798$ images as training and testing sets, respectively.

3.3. Dataset Structure

The IP102 dataset has a hierarchical structure and Fig. 4 shows its detailed taxonomy. Each *sub-class* is assigned with a *super-class* according to the crop that the insect pest class mainly damages. For example, the *sub-class* of tetranychus cinnbarinus (TC) has the *super-class* of citrus. The 8 crops (*e.g*., rice, corn, and wheat) are further grouped into two *super-classes* (*i.e*., field crop and economic crop). For example, the citrus belongs to the *super-class* of economic crop. In addition, Table 1 shows the number distributions of *sub-classes* in different *super-class* levels.

3.4. Comparison with Other Datasets

In Table 2, we compare the IP102 with several existing datasets related to the task of insect pest recognition. Compared to the largest datasets [23, 43, 2], our dataset contains over 14 times more samples. With respect to the class diversity, the largest and least datasets only have 40 and 8 classes, respectively. However, there are a large number of insect pests in real life and our IP102 comprises of 102 classes. Considering the average number of samples per class, the IP102 has at least 309 more images than those compared datasets. In addition to the statistic distinction, only half of the datasets is available and only [43] has a relatively large scale. Due to these limitations, most existing datasets (*e.g*., [40, 44, 7]) related to insect pests are hard to be applied to practical applications.

3.5. Diversity and Difficulty

Insect pests at different stages of life cycle can damage agricultural products in different degrees. So we retain images containing all of these during data collection and annotation. Figs. 2(a_1-a_4) show different forms of pests in the IP102, containing egg, larva, pupa, and adult. For the classification model, classifying them to the same category is difficult because it is hard to extract discriminant features. In addition to the biological diversity, the imbalanced data distribution also cannot be ignored. As illustrated in Fig. 3, the three sub-figures demonstrate the imbalanced distribution of the proposed dataset in different levels, where (a), (b), and (c) show the instance number distributions of the 102 sub-classes, 2 super-classes, and 8 super-classes, respectively. Specifically, based on the hierarchical label system of the IP102 dataset, the 102 sub-classes are divided into 8 super-classes according to the crop that the insect pest class mainly damages, *e.g*., rice and corn, and 2 super-classes according to the type of damaged crops, *i.e*., field crop and economic crop. The imbalanced distribution across different levels brings challenges to the imbalanced learning field and the use of hierarchical labels. Table 1 also shows that the dataset has high imbalance ratio (IR) (*i.e*., higher than 9 IR [12]) at most *super-class* level of the IP102. Imbalanced data can lead to the classification model learning a biased result to those classes with relative more training samples.

4. Experimental Evaluation

The choice of features usually plays a significant role in image recognition. To comprehensively evaluate the IP102 dataset, we first evaluate the classification performance utilizing handcrafted and deep features, respectively. Subsequently, we evaluate several object detection frameworks on the subset of IP102.

4.1. Experiment Settings

The SVM classifier is trained with the one-vs-rest scheme by employing LIBLINEAR [11]. The near neighbor number of the KNN classifier is set to 5. When training the deep networks, we fine-tune all layers via a Mini-

Table 3. Classification performance of the SVM and KNN classifiers under different evaluation metrics on the IP102 dataset. The representations are divided into handcrafted and deep features, respectively.

#	Methods	SVM						KNN					
		Pre	Rec	F1	GM	M_{AUC}	Acc	Pre	Rec	F1	GM	M_{AUC}	Acc
Handcrafted Feature	CH	9.7	3.2	2.5	0.3	12.0	12.9	18.2	14.2	15.0	8.3	16.8	15.8
	Gabor [29]	8.5	3.9	3.6	0.5	12.1	14.2	22.0	14.9	16.5	9.1	20.0	19.2
	GIST [30]	12.2	3.8	3.8	0.6	12.1	13.1	19.1	15.1	15.4	9.2	19.2	18.2
	SIFT [25]	25.1	6.3	6.8	1.0	19.9	18.1	19.4	10.3	12.1	5.6	15.9	13.1
	SURF [3]	28.2	7.3	8.3	1.5	21.2	19.5	21.3	11.5	13.4	7.1	17.5	14.7
	LCH [38]	7.2	5.0	4.7	0.9	11.1	13.1	21.6	14.7	16.1	8.3	19.0	16.8
Deep Feature	Alexnet [17]	41.5	16.4	21.0	9.3	32.5	28.3	36.7	32.4	33.5	23.9	41.0	40.7
	GoogleNet [39]	45.8	25.8	30.4	16.0	41.9	40.5	36.8	31.7	33.0	23.3	41.6	40.7
	VGGNet [36]	43.4	37.6	39.1	28.3	48.1	48.7	41.9	37.8	39.0	29.8	47.6	47.1
	ResNet [13]	43.6	39.1	40.6	31.0	48.7	49.5	43.7	39.1	40.5	30.7	48.2	49.4

batch Stochastic Gradient Descent optimizer with the mini-batch size of 64. The learning rate is initialized as 0.01 and drops by a factor of 0.1 every 40 epochs. The weight decay and momentum parameters are set to 0.0005 and 0.9, respectively. To avoid overfitting, we also employ the dropout [37], set to 0.3. We keep the basic architectures of these deep models unchanged, and only change the last fully connected layer from $1,000$ to the class number we aim to classify. The size of input images is fixed to 224×224. The deep feature based experiments are implemented using PyTorch [31] and performed on an NVIDIA Titan X GPU with 12 GB onboard memory.

4.2. Evaluation Metrics

The IP102 has an imbalanced class distribution. We employ several comprehensively metrics for the classification task, including precision, recall, F-measure, G-mean, and M_{AUC}. The precision (denoted as Pre) describes the ability of the classifier not to label a negative sample as positive. The recall (denoted as Rec) indicates the ability to find all the positive samples for one specific class. The F1 combines the precision and recall as a trade-off. The G-mean (denoted as GM) evaluates class-wise sensitivity and indicates the balanced classification performances on the majority and minority classes. The micro average scheme M_{AUC} [15] is defined as the area under the curve metric. As for the task of object detection, we utilize the Average Precision (AP) (IoU=[.50:.05:.95]), $AP^{.50}$ (IoU=.50), and $AP^{.75}$ (IoU=.75) as performance evaluation metrics. The IoU is defined as the intersection over the union between detected box and ground-truth. The larger the threshold of IoU, the greater the difficulty of detection.

Table 4. Classification performance of different deep models. The 'st' denotes training from scratch.

Method	F1	GM	Acc	$F1^{st}$	GM^{st}	Acc^{st}
AlexNet [17]	34.1	27.0	41.8	29.1	22.2	35.3
GoogleNet [39]	32.7	21.3	43.5	27.0	11.3	40.2
VGGNet [36]	38.7	30.9	48.2	33.3	25.5	41.4
ResNet [13]	40.1	31.5	49.4	29.6	22.2	35.7

4.3. Classification Results on Handcrafted Features

We extract several handcrafted texture and color features from the IP102 dataset, including Color Histogram (CH), LCH [38], Gabor [29], GIST [30], SIFT [25], and SURF [3]. Then, we utilize the SVM and KNN classifiers to build the baseline methods on handcrafted features.

Table 3 shows the classification performance of handcrafted features. We can see that color (CH) features perform poorly on most evaluation metrics compared to texture (Gabor [29]) features. This indicates that texture features play a more important role when insect pests appear in the wild. As shown in Fig. 1, large area of monotonous background color makes it difficult to discriminate insect pests by color features. The best handcrafted feature barely achieves about 19.5% accuracy with SURF [3] features and SVM classifier. The main reason is that these handcrafted features can neither catch the comprehensive information relating to insect pests nor eliminate the noise in pests images in the real environment. Furthermore, plenty of different insect pests share similar appearance but traditional handcrafted features are not enough to capture subtle differences. The large accuracy gap between the IP102 and previous small-scale dataset [19, 44] also demonstrates that the IP102 exhibits high recognition difficulty.

4.4. Classification Results on Deep Features

Deep features are proved to be effective in image recognition. In this section, we evaluate the performance of state-of-the-art deep convolutional networks on the IP102 dataset, including AlexNet [17], GoogleNet [39], VGGNet-16 (VGGNet) [36], and ResNet-50 (ResNet) [13].

All the networks are pre-trained on the ImageNet [6] and then fine-tuned on the IP102 dataset. We extract deep features from the CNNs by removing the last layer in model architectures. Subsequently, we utilize these deep features to train the SVM and KNN classifiers. Table 3 shows the classification performance of deep features. The ResNet performs best compared to the other three models on most of the metrics. So it can make a better feature representation of the IP102, even if its feature dimension $(2,048)$ is less than VGGNet $(4,096)$. In addition, deep feature outperforms handcrafted feature based methods in general. This demonstrates the feature learning ability of deep models. Then, we can further see that the KNN performs better overall versus the SVM classifier. Especially with the AlexNet features, the KNN results outperform the SVM on most of the metrics. It has 40.7% accuracy with the KNN while only has 28.3% accuracy with the SVM classifier. Moreover, the SVM achieves the poor performance of 16.4 recall and 9.3% G-mean. This illustrates that the deep features from AlexNet have low sensitivity.

Table 4 shows the softmax classification performance of deep models on different evaluation metrics. Note that ResNet achieves the best results on all metrics. Yet the big gap between 49.4% accuracy and 31.5% G-mean indicates the high imbalance of our IP102 dataset. The classification models bias to those classes with a large number of samples. In addition, the highest accuracy of 49.4% demonstrates the challenges of IP102. We also train the deep models from scratch, *i.e*., without pre-training on the ImageNet. The results are much worse compared to fine-tuning pre-trained models, due to the fact that these deep models have a huge number of hyper-parameters and can easily overfit on the classes with fewer training samples.

4.5. Detection Results

We evaluate several state-of-the-art object detection methods on the IP102 dataset. Two stage based methods including Faster R-CNN (FRCN) [34] and FPN [20] (utilizing FRCN as the backbone detection framework). They detect objects through first sliding the window on a feature map to scan potential objects and then classifying them and regressing corresponding box coordinates. One stage based methods including SSD300 [22], RefineDet [45], and YOLOv3 [33] directly regress the category and position for each object. Detection performance in Table 6 shows the superiority of region proposal based two-stage detectors (FPN) over the unified ones (SSD300, RefineDet, and YOLOv3). We observe that combining the feature maps from multiple layers (FPN and YOLOv3) in deep networks is efficient for the multi-scale adaption of object sizes.

Table 5. Classification performance with different hierarchical labels. Each row shows the results of the *sub-classes* of corresponding crop.

Super-Class		Pre	Rec	F1	GM	M$_{\text{AUC}}$	Acc
FC	Rice	31.5	30.0	30.4	28.3	32.3	32.1
	Corn	55.1	54.4	54.6	50.3	61.9	62.2
	Wheat	37.5	34.5	35.5	29.3	52.1	53.0
	Beet	51.6	49.5	50.4	45.3	62.0	62.2
	Alfalfa	42.1	41.2	41.4	38.1	46.2	46.4
EC	Vitis	78.2	76.3	77.1	74.9	86.8	86.7
	Citrus	69.6	68.5	68.8	65.2	76.6	76.6
	Mango	75.8	74.7	75.1	72.3	89.0	89.0

Table 6. Average precision performance of object detection methods under different IoU thresholds.

Method	Backbone	AP	AP$^{.50}$	AP$^{.75}$
FRCNN [34]	VGG-16	21.05	47.87	15.23
FPN [20]	ResNet-50	28.10	54.93	23.30
SSD300 [22]	VGG-16	21.49	47.21	16.57
RefineDet [45]	VGG-16	22.84	49.01	16.82
YOLOv3 [33]	DarkNet-53	25.67	50.64	21.79

4.6. Further Analysis

In Table 5, we further evaluate the performance of deep models on each *super-class*. In the hierarchical structure of our proposed IP102 dataset, each *sub-class* is assigned a *super-class*. Each *super-class* is a subset of IP102, which covers a portion of the 102 insect pests. For example, for the *super-class* "Rice", our target is to classify one of the subset of IP102 into 14 categories. The detailed class distribution on *super-class* is shown in Table 1. We choose ResNet [13] as the basic CNN model, which performs best on the IP102 in the last subsection. We also report the classification results on the metrics for imbalanced learning evaluation, since the sample number distribution of the IP102 at the *super-class* level is still imbalanced as shown in Fig. 3. Observed from Table 5, the model performance varies among the 8 *super-classes*. Moreover, the gap between the best performance "Mango" and the worst performance "Rice" is 56.9% accuracy. The classification results

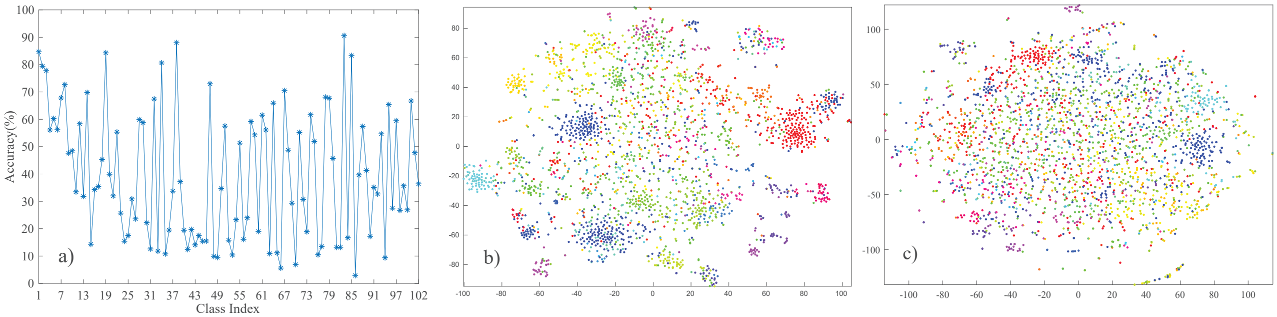

Figure 5. (a) The top-1 accuracy of ResNet on each *sub-class* of the IP102. (b) and (c) Visualizations of 2D t-SNE [26] feature embeddings on the IP102. (b) ResNet fine-tuned from the IP102 with ImageNet pre-training. (c) ResNet trained from scratch on the IP102.

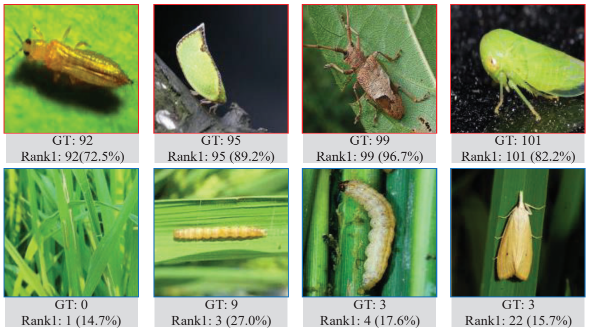

Figure 6. Samples of ResNet classification results on the "Mango" (top) and "Rice" (bottom) *super-classes*. The images in the top row are correctly classified and those in the bottom row are wrongly classified.

on these two *super-classes* are illustrated in Fig. 6. We can see that the insect pests from "Mango" have discriminative characteristics in respect of shape, color, background *etc*. As for "Rice", the images are easily misclassified due to three aspects. First, the colors between object and background are similar. The pests are hard to be distinguished with massive background information. Second, the intra-class variation is large, as illustrated in Fig. 2. These pests typically affect crops to varying degrees throughout their life cycle, and they are hard to be correctly classified especially in the larval period. Third, the pests between classes are often similar, *e.g*., Asiatic rice borer and yellow rice borer. Consequently, as illustrated in Fig. 7, the difficulty of insect pest recognition also brings challenges to the detection task. Even the target is accurately detected, yet it may be misclassified.

Moreover, in Fig. 5(a), we show the classification accuracy results of ResNet [13] on each *sub-class* of IP102. In addition, Fig. 5(b) and Fig. 5(c) visualize the feature embedding of IP102 by t-SNE [26]. We can see that, with the ImageNet [17] pre-trained model, ResNet represents better to discriminate different insect pests in the feature space.

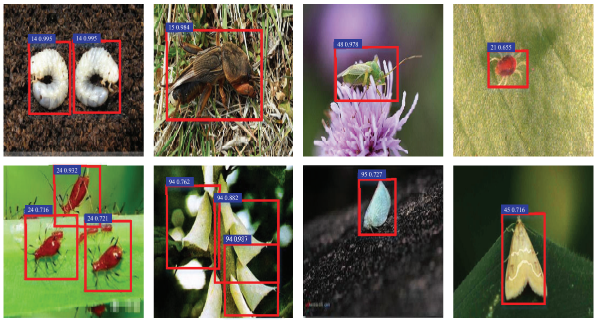

Figure 7. Sample detection results on the IP102 dataset. The top row shows the images which are correctly detected. The bottom row shows some failure cases, such as the right two images which are correctly detected but wrongly classified.

5. Conclusion

In this work, we collect a large-scale dataset, named IP102, for insect pest recognition, including over $75,000$ images of 102 species. Compared with previous datasets, the IP102 conforms to several characteristics of insect pest distribution in the real environments (*e.g*., diversity and class imbalance). Meanwhile, we also evaluate some state-of-the-art recognition methods on our dataset. The results demonstrate that current handcrafted feature methods and deep feature methods cannot yet handle the pest recognition well. We hope this work will help advance future research on several fundamental problems as well as common object classification and detection tasks, such as the fine-grained visual classification and imbalanced learning *etc*.

Acknowledgment

This work was supported by the NSFC (No. 61876094, 61620106008, 61572264), Natural Science Foundation of Tianjin, China (No. 18JCYBJC15400, 18ZXZNGX00110, 17JCJQJC43700), the National Youth Talent Support Program, and the Open Project Program of the National Laboratory of Pattern Recognition (NLPR).

References

[1] H Al Hiary, S Bani Ahmad, M Reyalat, M Braik, and Z Alrahamneh. Fast and accurate detection and dlassification of plant diseases. *International Journal of Computer Applications*, 17(1):31–38, 2011.

[2] Ahmad Arib Alfarisy, Quan Chen, and Minyi Guo. Deep learning based classification for paddy pests & diseases recognition. In *ICMAI*, 2018.

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *ECCV*, 2006.

[4] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[5] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[7] Limiao Deng, Yanjiang Wang, Zhongzhi Han, and Renshi Yu. Research on insect pest image detection and recognition based on bio-inspired methods. *Biosystems Engineering*, 169:139–148, 2018.

[8] Juan J Estruch, Nadine B Carozzi, Nalini Desai, Nicholas B Duck, Gregory W Warren, and Michael G Koziel. Transgenic plants: An emerging approach to pest control. *Nature Biotechnology*, 15(2):137, 1997.

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[10] Fina Faithpraise, Philip Birch, Rupert Young, J Obu, Bassey Faithpraise, and Chris Chatwin. Automatic plant pest detection and recognition using k-means clustering algorithm and correspondence filters. *International Journal of Advanced Biotechnology and Research*, 4(2):189–199, 2013.

[11] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9(Aug):1871–1874, 2008.

[12] Alberto Fernández, Salvador García, María José del Jesus, and Francisco Herrera. A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. *Fuzzy Sets and Systems*, 159(18):2378–2398, 2008.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[14] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, 2014.

[15] Qi Kang, Lei Shi, MengChu Zhou, XueSong Wang, QiDi Wu, and Zhi Wei. A distance-based weighted undersampling scheme for support vector machines and its application to imbalanced classification. *IEEE Transactions on Neural Networks and Learning Systems*, 29(9):4152–4165, 2018.

[16] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *ICCV Workshop*, 2013.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[18] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[20] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, 2016.

[23] Ziyi Liu, Junfeng Gao, Guoguo Yang, Huan Zhang, and Yong He. Localization and classification of paddy field pests using a saliency map and deep convolutional neural network. *Scientific Reports*, 6:20410, 2016.

[24] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[25] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[26] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[27] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[28] M Manoja and J Rajalakshmi. Early detection of pest on leaves using support vector machine. *International Journal of Electrical and Electronics Research*, 2(4):187–194, 2014.

[29] Rajiv Mehrotra, Kameswara Rao Namuduri, and Nagarajan Ranganathan. Gabor filter-based edge detection. *Pattern Recognition*, 25(12):1479–1494, 1992.

[30] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[31] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Workshop*, 2017.

[32] R Uma Rani and P Amsini. Pest identification in leaf images using SVM classifier. *International Journal of Computational Intelligence and Informatics*, 6(1):30–41, 2016.

[33] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[35] RK Samanta and Indrajit Ghosh. Tea insect pests classification based on artificial neural networks. *International Journal of Computer Engineering Science*, 2(6):336, 2012.

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[37] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[38] Michael J Swain and Dana H Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[39] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[40] Kanesh Venugoban and Amirthalingam Ramanan. Image classification of paddy field insect pests using gradient-based features. *International Journal of Machine Learning and Computing*, 4(1):1–5, 2014.

[41] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[42] Jiangning Wang, Congtian Lin, Liqiang Ji, and Aiping Liang. A new automatic identification system of insect images at the order level. *Knowledge-Based Systems*, 33:102–110, 2012.

[43] Chengjun Xie, Rujing Wang, Jie Zhang, Peng Chen, Wei Dong, Rui Li, Tianjiao Chen, and Hongbo Chen. Multi-level learning features for automatic classification of field crop pests. *Computers and Electronics in Agriculture*, 152:233–241, 2018.

[44] Chengjun Xie, Jie Zhang, Rui Li, Jinyan Li, Peilin Hong, Junfeng Xia, and Peng Chen. Automatic classification for field crop insects via multiple-task sparse representation and multiple-kernel learning. *Computers and Electronics in Agriculture*, 119:123–132, 2015.

[45] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Single-shot refinement neural network for object detection. In *CVPR*, 2018.