IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 June 2022, accepted 26 June 2022, date of publication 4 July 2022, date of current version 8 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3188282

RESEARCH ARTICLE

A Method to Establish a Synthetic Image Dataset of Stored-Product Insects for Insect Detection

JIANGTAO LI, YUWEI SU, ZHAOJUN CUI, JIDA TIAN, AND HUILING ZHOU
School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding authors: Jiangtao Li (lijiangtao@bupt.edu.cn) and Huiling Zhou (huiling@bupt.edu.cn)

This work was supported by the Beijing University of Posts and Telecomunications (BUPT) Excellent Ph.D. Students Foundation under Grant CX2020313.

ABSTRACT In recent years, deep-learning models have resulted in significant progress in insect recognition. However, training deep neural networks requires a large amount of data, and data collection and labeling are time consuming and labor intensive. This study proposes a method for establishing a synthetic image dataset of stored-product insects to provide well-labelled image data for insect detection tasks. Proxy virtual worlds are leveraged to obtain synthetic data with annotations. A dynamic generation approach was presented to generate synthetic images with diverse insect targets, various backgrounds, and changing lighting conditions by using a camera module in the constructed virtual scene. The coordinates of the bounding boxes and the category labels of insect targets in each synthetic image were obtained by calculating the geometrical relationships between the insect targets and the camera module. A texture translation network was developed to conduct image-to-image translation and launch to enhance the verisimilitude of the synthetic images. A synthetic image dataset was established for three insect species, *Cyptolestes ferrugineus* (Stephens), *Sitophilus oryzae* (Linnaeus), and *Tribolium castaneum* (Herbst).A set of assessments was introduced to evaluate the synthetic image dataset, including the statistical characteristics and experimental verification. The experimental results demonstrated that the use of synthetic data reduces the demand for real data. The proposed method may provide a novel solution for providing training data with correct annotations for insect detection, without tedious image collection and manual labeling.

INDEX TERMS Dynamic generation, insect detection, stored-product insect, synthetic image dataset, texture translation, virtual world.

I. INTRODUCTION

Infestation by stored-product insects is one of the most common causes of grain storage loss. This results in the loss of grain quantity, fungal growth, and quality degradation. Hence, the effective monitoring of stored-product insects is essential. In recent years, with improved hardware computing capability, computer vision techniques based on deep learning have achieved remarkable progress in general object detection [1] and other computer-vision-related studies [2], [3]. Because deep learning methods can avoid rule-based segmentation pipelines and labor-intensive feature engineering, many insect detection tasks have adopted achievements from general object detection tasks [4]–[7].

A growing number of studies have applied deep learning-based methods to process images or videos to monitor stored-product insects [8]–[12]. Recently, an increasing number of grain depots have been equipped with high-definition security cameras and insect-monitoring devices that collect images or videos to detect the occurrence of insects on the surface of grain piles. Therefore, image recognition based on deep learning for insect detection has significant practical applications. However, learning deep hierarchical representations requires a large amount of accurately labeled data, which is the main limitation of using deep convolutional neural networks for insect detection [13].

Our previous study established an image dataset of stored-product insects for specific insect detection devices and application scenarios [14]. However, image collection and labeling are time-consuming, error prone, and laborious. Collecting and labeling insect images is more tedious than collecting

The associate editor coordinating the review of this manuscript and approving it for publication was Lei Wei.

generic images (pedestrians, vehicles, or familiar objects in our daily lives) because of their millimeter-level body size, similar appearance, and scattered distribution in granaries. Furthermore, experts with professional backgrounds are required to guarantee the accuracy of annotations. All these problems remain a crucial bottleneck for insect image recognition.

The use of synthetic images has become an effective method for training and evaluating deep neural networks for certain computer vision tasks where it is difficult to collect ground truth labels. There are two main technical categories: synthesis methods based on computer graphics and generation methods based on generative adversarial networks (GANs) [15].

Synthesis methods leverage computer graphics to acquire fully labeled, dynamic, and photorealistic synthetic images in proxy virtual worlds. Gaidon *et al.* [16] constructed a synthetic dataset (Virtual KITTI) using the Unity3D game engine, and automatically generated annotations for object detection, tracking, depth, and optical flow. Tremblay *et al.* [17] proposed a synthetic dataset containing images with accurate annotations using Unreal Engine 4 (UE4) to combine 3D models with a complex background. Synthesis methods have two significant advantages:1) virtual environments and objects can be selected and designed to control the quantity and variety of synthetic images; and 2) synthetic images are snapshots taken by the virtual stereo camera system, so annotations including object category, bounding box, pixel category, depth, and optical flow can be obtained by computing the geometrical relationship between objects and the camera system in virtual worlds.

Generation methods apply GANs and their variants to acquire synthetic images for data augmentation [18]. Generation methods have been successfully applied in some classification tasks related to agriculture [19]–[21]; they have improved recognition performance by using GANs for data augmentation in their studies. In addition, Abbas *et al.* [22] adopted the conditional GAN [23] to generate synthetic images of tomato plant leaves and improve network generalizability using transfer learning with synthetic images. Cabrera and Villanueva [24] used generation models to synthesize image patches of pests and stick them to actual images to enhance the training dataset, with the aim of facilitating the training of insect detection models.

Inspired by these studies, a method to establish a synthetic image dataset of stored-product insects is proposed by combining synthesis and generation methods and launching to provide a large amount of well-labelled image data for insect detection without tedious collection and labeling. In this study, we used *Cyptolestes ferrugineus* (Stephens), *Sitophilus oryzae* (Linnaeus), and *Tribolium castaneum* (Herbst) adults as examples to illustrate the proposed method and verify the effectiveness of the synthetic image dataset. The main contributions of this study are as follows:

1) A dynamic generation approach is proposed to simultaneously generate synthetic images of multiple detection scenarios and export annotations by calculating the geometric relationship between insect targets and the camera system in the proxy virtual world.

2) A texture translation network was developed based on cycle-consistent adversarial networks (Cycle-GAN) [25] to enhance the verisimilitude of synthetic insect images.

3) A synthetic dataset was established for the three species of stored-product insects. A set of assessments was introduced to evaluate the synthetic image dataset, including the statistical characteristics and experimental verification.

II. METHODOLOGY

Establishing the synthetic image dataset consists of three stages (Fig. 1):

1) Manually design 3D models to match the insect's geometry and joint structures.

2) Construct virtual scenes according to the proposed dynamic generation approach and simultaneously generate synthetic images with accurate annotations.

3) Implement image-to-image translation to make synthetic images more photorealistic using the developed texture translation network.

A. 3D MODELING OF INSECTS

Fig.2 shows the steps for building a 3D model of an insect. First, we carefully observed the shapes of insects and crawling postures under high-definition micro-devices. Second, we manually constructed a high-precision mesh with the same proportion of the actual insect and built the body, head, horns, legs, and wings separately to further simulate the postures of insects by designing skeletal animation. Third, the texture was designed using high-definition images of insects, and normal mapping was adopted to map the texture onto the mesh to form a 3D model of insects. Fig.3 shows pictures of 3D models of *C. ferrugineus*, *S. oryzae*, and *T. castaneum* adults. In this study, 3D modeling of insects was implemented using the 3Ds Max software.

B. DYNAMIC GENERATION OF SYNTHETIC IMAGES

1) CONSTRUCTION OF THE VIRTUAL SCENE

The virtual scene includes four parts: background, insect targets, environment variables, and camera module, as shown in Fig.4. The background was a rectangular static mesh textured with background images that exhibited insect detection scenarios, such as sticky boards and bulk grain surfaces. The insect targets are 3D models of stored-product insects. The environmental variables were the brightness and color of the light source. The camera module placed above the insect targets in the virtual scene captures snapshots and controls the snapshot resolution, shooting angle, and field of view. This study constructed a virtual scene using UE4.

2) DYNAMIC GENERATION APPROACH

Owing to the various insect detection devices applied in insect monitoring practices, there are apparent differences in

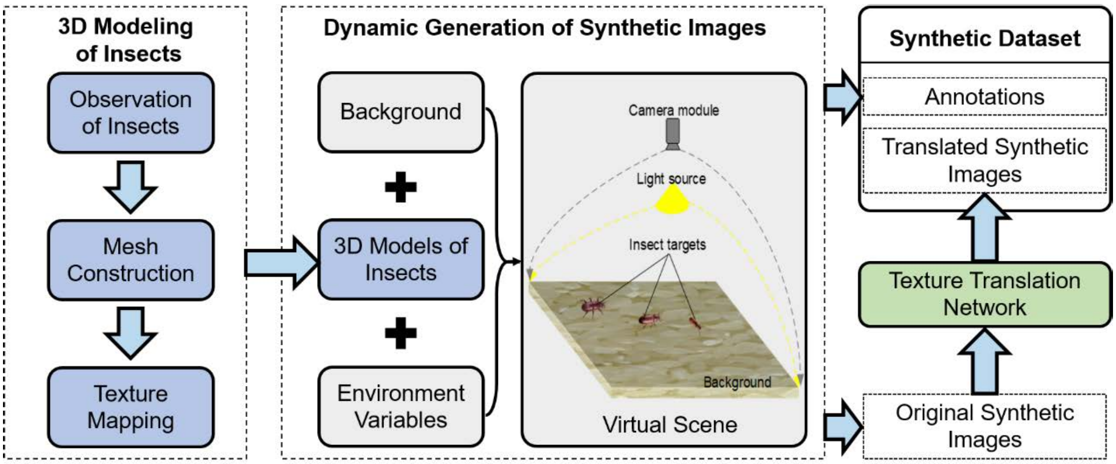

FIGURE 1. Block diagram for establishing the synthetic image dataset.

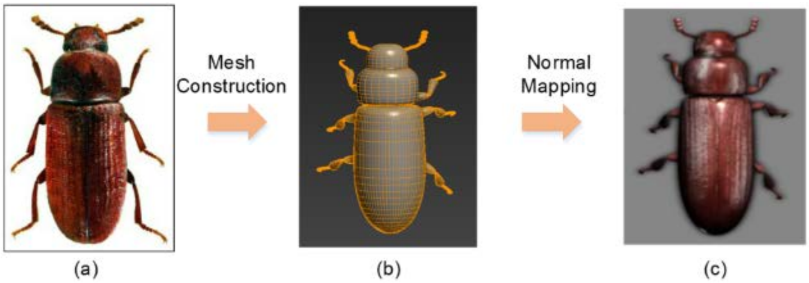

FIGURE 2. Steps for building a 3D model of a T. castaneum adult. (a) A high-definition image of an actual insect. (b) The constructed mesh of a 3D model. (c) The 3D model mapped with the texture.

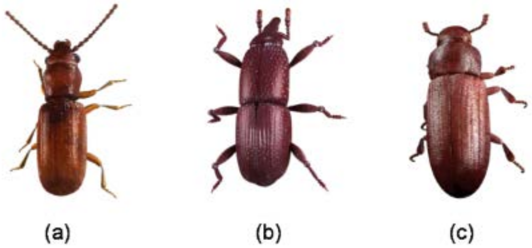

FIGURE 3. 3D models of three species of stored-product insects. (a) A C. ferrugineus adult. (b) A S. oryzae adult. (c) A T. castaneum adult.

the insect images. Specifically, insects always have diverse positions, poses, and scales owing to different detection environments and shooting distances. Moreover, different lighting conditions and cameras lead to different brightness, hues, background, and resolution of the insect images. Therefore, a dynamic generation equation (1) was proposed to generate insect targets with various scales, postures, and positions on diverse backgrounds with different lighting conditions in the virtual scene to simulate complex situations in the field.

$$D(t) = Bac(t) + Ins(t) + Env(t) \quad t = 0, 1, 2, \ldots T \quad (1)$$

where D(t), Bac(t), Ins(t), and Env(t) represent the virtual scene, background, insect targets, and environmental variables at time t, respectively. The virtual scene is updated at each time step t. The pseudocode explaining the flow of this approach is shown in Algorithm1. Details regarding the generation of the background, insect targets, and environmental variables are provided below.

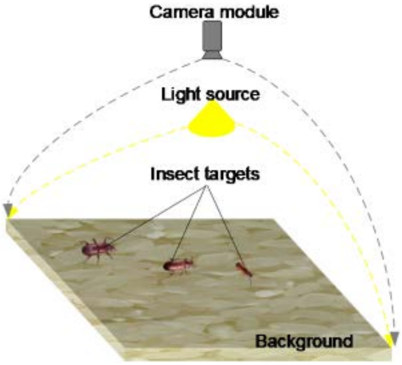

FIGURE 4. Structure of a virtual scene.

a: BACKGROUND

Background images were collected from practical insect detection scenarios, such as sticky boards, collectors of radially fluted plate traps, and surfaces of bulk grains. Because background images are collected from real insect-monitoring

Algorithm 1 The Flow of the Dynamic Generation Approach

```
Input:
    The overall running time of the dynamic generation process, $T$;
    The number of insect species, $M$;
    The number of insects of the kth species, $N_k$;
    The 3D model of insects, Insect, whose data structure is class, initialized with the species index $k$;
    The set of background images, BD;
Output:
    The snapshot of the virtual scene at each time step $t$;
    Annotations of the snapshot.
for $t = 1; t < T; t++$ do
    $\mathrm{Bac}(t) = \mathrm{BD}\ [t\ \%\ \mathrm{len(BD)}]$;
    $\mathrm{Ins}(t) = []$;
    for $i = 1; i < M \times N_k; t++$ do
        $insect(i) = \mathrm{Insect.init}(k)$;
        for each $Joint_{ij}^{k}$ in $insect(i)$ do
            Trans $(Joint_{ij}^{k}, t)$;
        end for
        $\mathrm{Ins}(t) \cup \mathrm{LSR}(insect(i), t)$;
    end for
    $\mathrm{Env}(t) = \mathrm{Light}(t) + \mathrm{Color}(t)$;
    $\mathrm{D}(t) = \mathrm{Bac}(t) + \mathrm{Ins}(t) + \mathrm{Env}(t)$;
end for
```

environments, there are interference factors affecting insect detection, such as powder, foreign matter, damaged grains, and other objects that are not insects. The store addresses of the background images are listed using a long list (background database, BD). Background Bac(t) was textured with the selected background image according to the selection rule defined in (2).

$$Bac(t) = BD[t\%len(BD)] \tag{2}$$

where $len(BD)$ represents the length of the list BD, and % is the remainder operator. According to the result (r) of the reminder operation, the rth background image is selected and textured onto the background mesh in the virtual scene. In this study, approximately 1,000 background images were collected.

b: INSECT TARGETS

Because there are differences among individual insects and complex crawling behaviors of insects, insects usually exhibit diverse appearances and postures in images captured in practice. To ensure the diversity and validity of insect targets, each body part was taken as the basic unit for the dynamic generation of each insect target to diversify the posture. The generation pipeline for insect targets was designed as shown in (3).

$$Ins(t) = \sum_{k=1}^{M}\sum_{i=1}^{N_k} LSR(Trans\sum_{j=1}^{P}(Joint_{ij}^{k}(t)) \tag{3}$$

TABLE 1. Constraints of Trans and LSR operations.

Operation		Range
Trans	Stretch ratio	[0.5,1.5]
	Rotation angle	[0°,90°]
LSR	Location	Within the background area
	Scale ratio	[0.5,1.5]
	Rotation angle	[0°,360°]

where M is the number of insect species, N_k is the number of insects of kth species, P is the number of body parts of an insect target, and the $Joint_{ij}^{k}(t)$ is the jth body part of the ith insect target belonging to the kth species. Trans is the skeletal animation performed on each body part of the insect targets to simulate the moving behavior of real insects. The operation to controls insect target location, scale, and rotation angle (head direction).

First, the Trans operation was performed on each body part of each insect target by stretching and rotating the separable body parts (Fig.5a). After the Trans-operation, insect targets with various postures were generated (Fig. 5b and 5c). Second, the LSR operation was implemented on each insect target in the virtual scene at time t. Through the LSR operation, insect targets with different positions, scales, and rotation angles were generated (Fig.5d). The generation process is executed at time t to update the state of all insect targets in the virtual scene. Thus, insect targets are not static, but move and change within the background area in the virtual scene. To guarantee the rationality and naturalness of insect targets, we set constraints for the Trans and LSR operations, as described in Table 1.

c: ENVIRONMENT VARIABLES

The environment variables are the light intensity and color in the virtual scene (Fig.6). We simulated different detection environments by varying these variables. The definition of Env(t) is shown in (4), where Light (t) represents light intensity and Color (t) represents light color at time t.

$$Env(t) = Light(t) + Color(t) \tag{4}$$

3) SYNTHETIC IMAGES AND ANNOTATIONS

In a virtual scene, the insect targets, background, and environment variables change dynamically according to the dynamic generation equation. The camera module in UE4 is programmed to capture snapshots of insect targets, as shown in Fig.4, and to control the resolution, shooting angle, and field of view of snapshots at the same time. After calculating the geometrical relationships between insect targets and the camera module, the coordinates of the bounding boxes and the category label of insect targets in each snapshot were obtained. Snapshots are exported as original synthetic images in the JPG format, and the corresponding annotations are stored in a JSON file. After testing, an average of 53 original

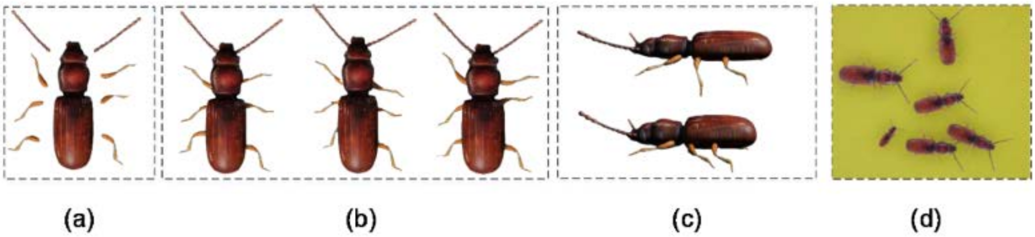

FIGURE 5. Generation process of insect targets. (a) Different body parts of a 3D model of a C. ferrugineus adult. (b) Top views of insect targets with different postures. (c) Side views of insect targets with different postures. (d) A synthetic image contains insect targets with different positions, scales, and rotation angles.

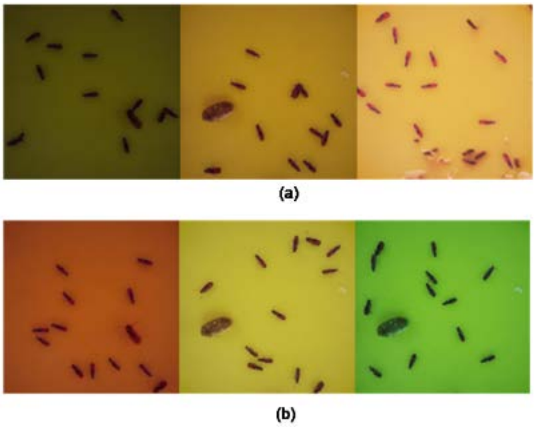

FIGURE 6. Synthetic images under different environmental settings. (a) Examples of different light intensities; (b) Examples of different light colors.

synthetic images were output per second using NVIDIA 1080ti.

C. TEXTURE TRANSLATION NETWORK

The texture translation network, named TextureNet, is proposed based on Cycle-GAN, which minimizes the difference between synthetic and real images. TextureNet can learn the cross-domain mapping from synthetic images (X from domain A) to real images (Y from domain B) and conduct image-to-image translation to make synthetic images contain more details similar to the real images.

TextureNet comprises two identical generators (G_{AB} and G_{BA}) and two identical discriminators (D_A and D_B) (Fig.7). G_{AB} translates synthetic images (X) to make them closer to real images (Y), whereas G_{BA} reconstructs synthetic images based on translated images. The D_A discriminates whether the input image is synthetic or reconstructed, and D_B discriminates whether the input image is real or translated. After training TextureNet, the desired translated synthetic images were generated.

1) GENERATOR

The generator model comprises four parts: Down-sampling block, Transforming block, Up-sampling block, and Output block (Table 2). There are three convolution layers (Conv2d) and two transposed convolution layers (T_Conv2d) in the Down-sampling and Up-sampling blocks, respectively. The Transforming block is composed of nine stacked residual blocks, and the structure of the residual block is shown in Fig.8.

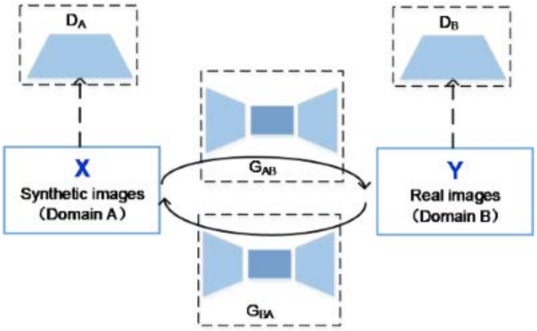

FIGURE 7. Overall structure of texture translation network.

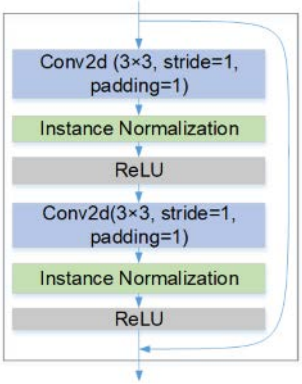
Conv2d (3×3, stride=1, padding=1)
Instance Normalization
ReLU
Conv2d(3×3, stride=1, padding=1)
Instance Normalization
ReLU

FIGURE 8. Structure of the Residual block.

2) DISCRIMINATOR

The discriminator model adopts the structure of a Patch GAN [26], which is composed of a series of convolution layers. In this study, Leaky ReLU was applied as the activation function of the discriminator. The details of the discriminator model are listed in Table 3.

TABLE 2. Summary of the generator model.

Block Name	Layer Name	Parameters
Down-sampling	Conv2d	7 × 7,stride=1,padding=3
	IN	Instance Normalization
	ReLU	Max(0,x)
	Conv2d	3 × 3,stride=2,padding=1
	IN	Instance Normalization
	ReLU	Max(0,x)
	Cov2d	3 × 3,stride=2,padding=1
	IN	Instance Normalization
	ReLU	Max(0,x)
Transforming	9 × Residual blocks	
Up-sampling	T_Conv2d	3 × 3,stride=2,padding=1
	IN	Instance Normalization
	ReLU	Max(0,x)
	T_Conv2d	3 × 3,stride=2,padding=1
	IN	Instance Normalization
	ReLU	Max(0,x)
Output	Reflection Padding	padding=3
	Conv2d	7 × 7,stride=1,padding=0
	Activation	Tanh(x)

TABLE 3. Summary of the discriminator model.

Layer Name	Parameters
Conv2d	4 × 4, stride=2, padding=1
Leaky ReLU	Negative slope=0.2
Conv2d	4 × 4, stride=2, padding=1
IN	Instance Normalization
Leaky ReLU	Negative slope=0.2
Conv2d	4 × 4, stride=2, padding=1
IN	Instance Normalization
Leaky ReLU	Negative slope=0.2
Conv2d	4 × 4, stride=1, padding=1
IN	Instance Normalization
Leaky ReLU	Negative slope=0.2
Conv2d	4 × 4, stride=1, padding=1
Output	Global average pool and Flatten

3) NETWORK TRAINING

The training process of TextureNet involves antagonism between the generators and discriminators. Through adversarial training using the losses defined in (5) and (6), generators output vivid fake images, whereas discriminators focus on distinguishing between fake and real images. Because synthetic images (X) and real images (Y) are un-paired, the cycle consistency loss composed of the forward and backward processes defined in (7) and (8), respectively, is added to supervise the training process of the two generators. Adding the cycle consistency loss can maintain the consistency of the shape, size, posture, location, and rotation angle of insect targets in the training process of generators. In addition, G_{AB} is designed to generate real images (Fake Y in Fig.9) based on synthetic images (X), so it should be able to generate real images (Fake $Y2$ in Fig.9) even when real images (Y) are sent as the input. Moreover, G_{BA} should be capable of generating synthetic images (Fake $X2$ in Fig.9) when synthetic images (X) are also sent as the input. Therefore, the identity losses defined in (9) and (10), are added to constrain the outputs of the two generators. The overall loss function (11) of TextureNet is the weighted sum of adversarial loss, cycle consistency loss and identity loss, as shown in Fig.9.

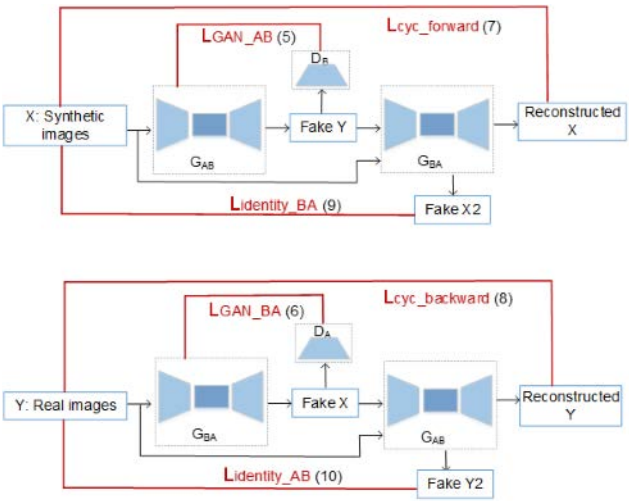

FIGURE 9. Training losses of texture translation network.

$$
\begin{aligned}
&L_{GAN\_AB}(G_{AB}, D_B, X, Y)\\
&\quad = E_{y\sim P_{data}(y)}[log(D_B(y))]\\
&\qquad + E_{x\sim P_{data}(x)}[log(1 - D_B(G_{AB}(x)))] \qquad (5)\\
&L_{GAN\_BA}(G_{BA}, D_A, X, Y)\\
&\quad = E_{x\sim P_{data}(x)}[log(D_A(x))]\\
&\qquad + E_{y\sim P_{data}(y)}[log(1 - D_A(G_{BA}(y)))] \qquad (6)\\
&L_{cyc\_forward}(G_{AB}, G_{BA}, X)\\
&\quad = E_{x\sim P_{data}(x)}[\|G_{BA}(G_{AB} - x)\|_1] \qquad (7)\\
&L_{cyc\_backward}(G_{AB}, G_{BA}, Y)\\
&\quad = E_{y\sim P_{data}(y)}[\|G_{AB}(G_{BA} - y)\|_1] \qquad (8)\\
&L_{identity\_AB}(G_{AB}, Y)\\
&\quad = E_{y\sim P_{data}(y)}[\|G_{AB}(y) - t\|_1] \qquad (9)\\
&L_{identity\_BA}(G_{BA}, X)\\
&\quad = E_{x\sim P_{data}(x)}[\|G_{BA}(x) - t\|_1] \qquad (10)\\
L_{total} &= \alpha(L_{GAN\_AB} + L_{GAN\_BA})\\
&\quad + \gamma(L_{cyc\_forward} + L_{cyc\_backward})\\
&\quad + \beta(L_{identity\_Ab} + L_{identity\_BA}) \qquad (11)\\
&L_G(G_{AB}, G_{BA}, X, Y)\\
&\quad = E_{x,y\sim P_{data}(x,y)}[\|y - G_{AB}(x)\|_1 - \|x - G_{BA}(y))\|_1] \qquad (12)
\end{aligned}
$$

The generators and discriminators were trained jointly. First, the parameters of the two discriminators were frozen, and the two generators were trained by calculating the sum of the identity losses, cycle consistency losses, and generator loss defined in (12). The second step was to sequentially train the two discriminators with the generator parameters frozen. The previous two steps were then repeated until convergence.

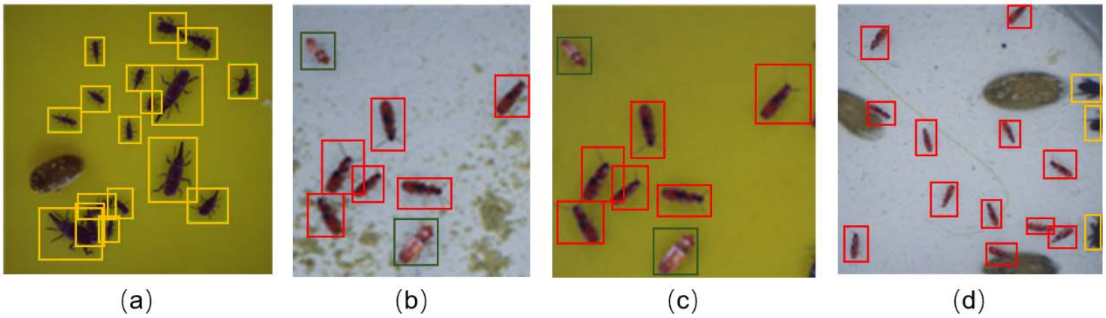

FIGURE 10. Example images in the synthetic dataset. Ground truth boxes are drawn in different colors for the three species of insects.

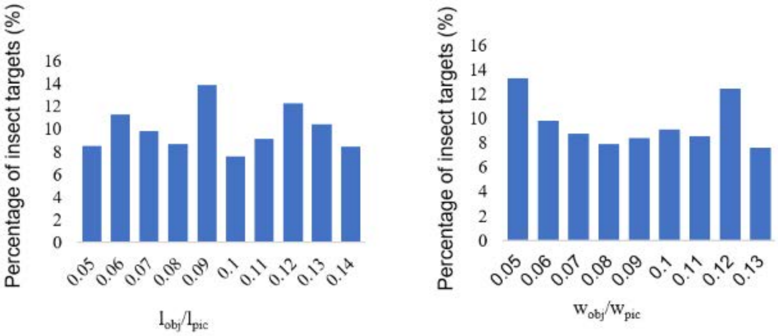

FIGURE 11. Distribution of the scale of insect targets.

TABLE 4. Experimental settings with various synthetic and real data ratios.

Experiments	Training data	Models
Experiment 1	100% real images	Baseline
Experiment 2	90% original synthetic images and 10% real images	SSD_10_O
	90% translated synthetic images and 10% real images	SSD_10_T
Experiment 3	80% original synthetic images and 20% real images	SSD_20_O
	80% translated synthetic images and 20% real images	SSD_20_T
Experiment 4	50% original synthetic images and 50% real images	SSD_50_O
	50% translated synthetic images and 50% real images	SSD_50_T

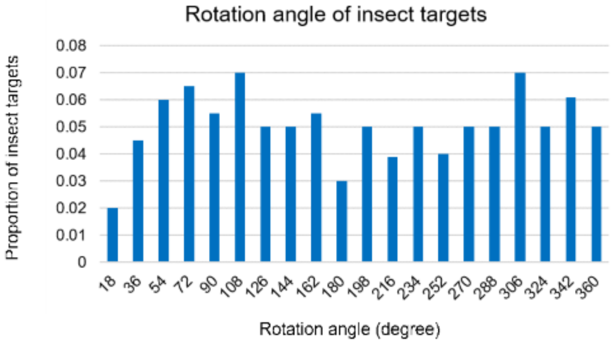

FIGURE 12. Distribution of rotation angles of insect targets.

III. SYNTHETIC DATASET AND EXPERIMENTS

A. SYNTHETIC DATASET

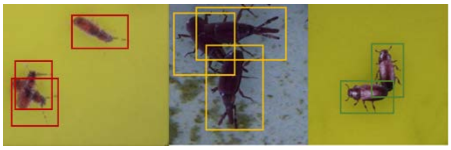

FIGURE 13. Examples of overlapped insect targets.

Based on the proposed method, a synthetic dataset named Virtual Insect was established for the three species of insects. This dataset contained approximately 10,000 synthetic images and 100,000 insect targets. Examples of these images are shown in Fig.10. The two ratios defined in (13) and (14) were calculated to exhibit the scale distribution of insect targets. The distributions of these two ratios are shown in Fig.11. The distribution of the rotation angles of the insect targets is shown in Fig.12.

$$r_l = l_{obj}/l_{pic} \tag{13}$$

$$r_w = w_{obj}/w_{pic} \tag{14}$$

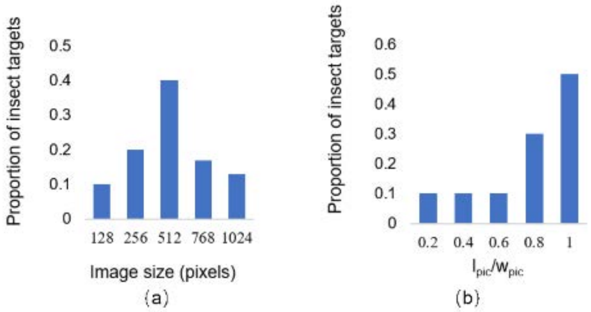

FIGURE 14. Statistics for image sizes and aspect ratios. (a) The distribution of image sizes. (b) The distribution of aspect ratios of images.

where l_{obj} is the length of the bounding box of the insect target, l_{pic} is the length of the image, w_{obj} is the width of the bounding box of the insect target, and w_{pic} is the width of the image.

Considering the occlusion of some insects in the actual images, overlapping insect targets in the synthetic images were generated, as shown in Fig. 13. Because synthetic images were captured in virtual worlds, the accuracy of annotations was guaranteed, although the insects highly overlapped. In practice, there is not only one type of device that can be applied for insect monitoring. The resolution of images taken by different photographing devices varies significantly, and most images do not have an aspect ratio of 1.0. Therefore, we made a prior design of the image size (length of the image) and aspect ratio (lpic / wpic). The statistical results for the image sizes and aspect ratios are presented in Fig.14.

B. INSECT DETECTION EXPERIMENTS

In this section, we explore and analyze the practical significance of the synthetic images and the effectiveness of TextureNet. Insect detection experiments using different combinations of synthetic and real data were conducted to verify the feasibility of the synthetic data for model training.

1) EVALUATION METRICS

We used each category's average precision (AP) and the mean value of each category's AP (mAP) as model performance evaluation metrics. When calculating the AP, the intersection-of-union threshold between the detections and nearby ground truth boxes was 0.5.

2) EXPERIMENT SETUP

Inspired by [27], we used different ratios of synthetic data instead of real data to train the same detection model with a fixed number of training images to verify the effectiveness of synthetic data as training data. Suppose that the model trained using a certain amount of image data, including synthetic and real data, achieves a similar performance as the model trained using the same amount of real data. In this case, it can be concluded that the synthetic data generated by the proposed method are adequate for insect detection tasks, and the synthetic dataset might alleviate the need for a large number of real images for model training. This study introduced a validation experiment with four settings, as defined in Table 4, to verify this assumption.

We adopted images from the RGBInsect dataset [14] as real data. Because the image resolution of RGB-Insect images is high, we cut these images using a window size of 512 pixels to acquire image patches of an appropriate size. These image patches are referred to as real images in the following sections. Single Shot Multi-Box Detector (SSD) [28] with VGG16 [29] backbone was applied as the insect detection model.

8,000 real images containing the three insect species were selected, of which 6,000 images were used as the training set and 2,000 images were used as the testing set. 6,000 original synthetic images and their corresponding translated synthetic images were prepared. We then randomly selected the corresponding number of real and synthetic images according to the experimental settings shown in Table 4 and conducted validation experiments. For each experimental setting, the process of image selection and model validation was repeated three times. The final results shown in Table 5 are the average values of the three experiments.

3) IMPLEMENTATION DETAILS

a: TRAINING OF TextureNet

A total of 1500 real images containing the three species of insects were randomly selected from RGBInsect, and the same quantity of original synthetic images was selected for training TextureNet. It is worth noting that the images used for training TextureNet are independent of the experimental images. Three TextureNets were trained for the three species of insects using the Adam optimizer with hyper-parameters β_1 of 0.5 and β_2 of 0.999. The images were resized to 512×512 pixels before being sent to the network. The training batch size was set as 1. The maximum number of iterations is 12. The learning rate remained at 0.0002 for the first six epochs and decreased by 0.00003 per epoch from the seventh epoch to the end of training. α, γ, and β in (11) are 1.0, 10.0, and 5.0, respectively.

b: TRAINING OF DETECTION MODELS

The SSD was first pretrained using synthetic images and fine-tuned using real images. The same training strategy was adopted for both the pretraining and fine-tuning processes. Images were resized to 300×300 pixels before being sent to the detection model. The training batch size was set as 32. The stochastic gradient descent (SGD) optimizer had an initial learning rate of 0.0001, momentum of 0.9, and weight decay of 0.9. the learning rate is multiplied by 0.1 after the 20th and 40th epochs. The maximum number of training epochs was set as 50. The first 40 epochs were trained using synthetic data and the last ten epochs were fine-tuned using real data. The experimental results are presented in Table 5.

TABLE 5. Insect detection results on the Testing set.

Model	Insect species			mAP
	Tc[a]	Cf[b]	So[c]	
Baseline	91.83	76.13	86.03	84.66
SSD_10_O	87.30	66.43	79.83	77.85
SSD_10_T	89.10	68.00	79.72	78.94
SSD_20_O	89.02	70.84	82.27	80.71
SSD_20_T	89.88	71.30	83.30	81.49
SSD_50_O	93.30	76.35	87.94	85.86
SSD_50_T	93.26	76.61	87.80	85.89

[a] Tc = *T. castaneum*
[b] Cf = *C. ferrugineus*
[c] So = *S. oryzae*

4) RESULTS AND DISCUSSION

From the results listed in Table 5, some promising conclusions can be summarized as follows:

Using synthetic images as the training data can significantly reduce the demand for a large number of real images. When 90% of the training images were synthetic, the detection mAP was 78.94%, decreasing by 5.71% compared with the Baseline. In particular, for C. ferrugineus adults, AP decreased by 8.13%. When the synthetic data ratio decreased to 80%, insect detection performance improved to 81.49%. There was still a drop of 3.17% in mAP compared to Baseline. Although insects' postures, textures, and background environments are considered when generating synthetic images, the decline in mAP implies that there is still a gap between synthetic data and real data. However, when the real data ratio was 50%, the detection mAP reached 85.89% and exceeded the Baseline by 1.23%. This result confirms the feasibility of using synthetic data for model training and reveals that synthetic data contain critical information that remains in the real data. In addition, using synthetic data for model pre-training provides a better initialization of the model parameters, and fewer real images are required for training with the desired performance.

Image-to-image translation is critical for improving insect detection performance when a considerable proportion of synthetic data is used for model training. When the proportion of synthetic images was 90% and 80%, image-to-image translation improved the detection performance by 1.09% and 0.87%, respectively. TextureNet learned the detailed characteristics of insect targets and the background of real images and made translated synthetic images more realistic. Thus, more realistic details were exhibited in the translated synthetic images, and more desired features were learned during training, which led to better detection performance. Examples of the translation results of the G_{AB} generator are shown in Fig. 15 to illustrate the effectiveness of TextureNet. When the proportion of real data increased to 50%, the desired features from real data might become a bottleneck for insect detection in this setting. Hence, the improvement in detection performance caused by image-to-image translation was only 0.03%.

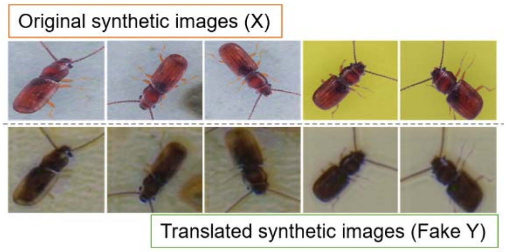
Original synthetic images (X)
Translated synthetic images (Fake Y)

FIGURE 15. Examples of translating results of the G_{AB} generator.

Nevertheless, it should be pointed out that the fidelity and fineness of the constructed virtual scene are limitations of the proposed method. Because the natural movements of insects are complex, some dusty environments in grain granaries are difficult to simulate in the virtual world. In future work, we will further expand the insect species of the synthetic dataset, design more reasonable skeletal amination for 3D models of insects and enrich background images by collecting more images from actual granaries in practice.

IV. CONCLUSION

This paper proposed a novel method for establishing a synthetic image dataset of stored-product insects, aiming to provide a large amount of training data for insect detection based on deep learning. The proposed dynamic generation approach can automatically generate well-labelled synthetic images containing diverse insect targets, backgrounds, and lighting conditions by simulating various detection scenarios in a proxy virtual world. The automatic production of well-labelled synthetic insect images can shorten the image collection period and reduce the labor-intensive manual labeling work to a great extent. Moreover, the developed texture translation network learns mapping from synthetic images to real images, making synthetic insect images more photorealistic. A synthetic image dataset named Virtual Insect was established for the three species of stored-product insects. Statistical analysis and validation experiments using different combinations of synthetic and real data are introduced. The validation experimental results demonstrate that using synthetic images as training data can significantly reduce the demand for a large number of real images. The proposed method to establish a synthetic image dataset might help relieve the scarcity of available image datasets for insect detection tasks, and has great potential for providing a large amount of training data for detecting insects in forestry and agriculture.

REFERENCES

[1] X. Wu, D. Sahoo, and S. C. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, Jul. 2020.

[2] L. Ruotsalainen, A. Morrison, M. Makela, J. Rantanen, and N. Sokolova, "Improving computer vision-based perception for collaborative indoor navigation," *IEEE Sensors J.*, vol. 22, no. 6, pp. 4816–4826, Mar. 2022.

[3] A. K.-F. Lui, Y.-H. Chan, and M.-F. Leung, "Modelling of destinations for data-driven pedestrian trajectory prediction in public buildings," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 1709–1717.

[4] L. Liu, R. Wang, C. Xie, P. Yang, F. Wang, C. Xie, P. Yang, F. Wang, S. Sudirman, and W. Liu, "PestNet: An end-to-end deep learning approach for large-scale multi-class pest detection and classification," *IEEE Access*, vol. 7, pp. 45301–45312, 2019.

[5] R. Li, X. Jia, M. Hu, M. Zhou, D. Li, W. Liu, R. Wang, J. Zhang, C. Xie, L. Liu, F. Wang, H. Chen, T. Chen, and H. Hu, "An effective data augmentation strategy for CNN-based pest localization and recognition in the field," *IEEE Access*, vol. 7, pp. 160274–160283, 2019.

[6] D. J. A. Rustia, C.-Y. Lu, J.-J. Chao, Y.-F. Wu, J.-Y. Chung, J.-C. Hsu, and T.-T. Lin, "Online semi-supervised learning applied to an automated insect pest monitoring system," *Biosyst. Eng.*, vol. 208, pp. 28–44, Aug. 2021.

[7] Y. Sun, X. Liu, M. Yuan, L. Ren, J. Wang, and Z. Chen, "Automatic in-trap pest detection using deep learning for pheromone-based dendroctonus valens monitoring," *Biosyst. Eng.*, vol. 176, pp. 140–150, Dec. 2018.

[8] L. Wu, Z. Liu, T. Bera, H. Ding, D. A. Langley, A. Jenkins-Barnes, C. Furlanello, V. Maggio, W. Tong, and J. Xu, "A deep learning model to recognize food contaminating beetle species based on elytra fragments," *Comput. Electron. Agricult.*, vol. 166, Nov. 2019, Art. no. 105002.

[9] Y. Shen, H. Zhou, J. Li, F. Jian, and D. S. Jayas, "Detection of stored-grain insects using deep learning," *Comput. Electron. Agricult.*, vol. 145, pp. 319–325, Feb. 2018.

[10] H. Zhou, H. Miao, J. Li, F. Jian, and D. S. Jayas, "A low-resolution image restoration classifier network to identify stored-grain insects from images of sticky boards," *Comput. Electron. Agricult.*, vol. 162, pp. 593–601, Jul. 2019.

[11] J. Li, H. Zhou, Z. Wang, and Q. Jia, "Multi-scale detection of stored-grain insects for intelligent monitoring," *Comput. Electron. Agricult.*, vol. 168, Jan. 2020, Art. no. 105114.

[12] S. Zhang, K. Xia, X. Du, H. Feng, and L. Chen, "SA faster R-CNN method for insect detection in stored grain based on clustering feature," *J. Chin. Cereal Oil Ass.*, vol. 35, no. 4, pp. 165–172, 2020.

[13] M. Martineau, D. Conte, R. Raveaux, I. Arnault, D. Munier, and G. Venturini, "A survey on image-based insect classification," *Pattern Recognit.*, vol. 65, pp. 273–284, May 2017.

[14] J. Li, H. Zhou, D. S. Jayas, and Q. Jia, "Construction of a dataset of stored-grain insects images for intelligent monitoring," *Appl. Eng. Agricult.*, vol. 35, no. 4, pp. 647–655, 2019.

[15] I. J. Goodfellow, "Generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 3, 2014, pp. 2672–2680.

[16] A. Gaidon, W. Qiao, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4340–4349.

[17] J. Tremblay, T. To, and S. Birchfield, "Falling things: A synthetic dataset for 3D object detection and pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 2119–21193.

[18] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," 2017, *arXiv:1711.04340*.

[19] M. V. Giuffrida, H. Scharr, and S. A. Tsaftaris, "ARIGAN: Synthetic arabidopsis plants using generative adversarial network," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 2064–2071.

[20] G. Hu, H. Wu, Y. Zhang, and M. Wan, "A low shot learning method for tea leaf's disease identification," *Comput. Electron. Agricult.*, vol. 163, Aug. 2019, Art. no. 104852.

[21] B. Espejo-Garcia, N. Mylonas, L. Athanasakos, E. Vali, and S. Fountas, "Combining generative adversarial networks and agricultural transfer learning for weeds identification," *Biosyst. Eng.*, vol. 204, pp. 79–89, Apr. 2021.

[22] A. Abbas, S. Jain, M. Gour, and S. Vankudothu, "Tomato plant disease detection using transfer learning with C-GAN synthetic images," *Comput. Electron. Agricult.*, vol. 187, Aug. 2021, Art. no. 106279.

[23] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.

[24] J. Cabrera and E. Villanueva, "Investigating generative neural-network models for building pest insect detectors in sticky trap images for the Peruvian horticulture," in *Proc. Annu. Int. Conf. Inf. Manage. Big Data*, vol. 2022, pp. 356–369.

[25] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2242–2251.

[26] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976.

[27] F. Erlik Nowruzi, P. Kapoor, D. Kolhatkar, F. Al Hassanat, R. Laganiere, and J. Rebut, "How much real data do we actually need: Analyzing object detection performance using synthetic and real data," 2019, *arXiv:1907.07061*.

[28] W. Liu, D. Anguelov, D. Erhan, and C. Szegedy, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

JIANGTAO LI received the B.E. degree in automation from the Beijing University of Posts and Telecommunications, Beijing, China, in 2015, where she is currently pursuing the Ph.D. degree with the School of Artificial Intelligence. Her several related works have been published in academic journals, including *Computers and Electronics in Agriculture* and *Applied Engineering in Agriculture*. Her research interests include deep learning and data mining.

YUWEI SU received the B.E. degree from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2020. He is currently pursuing the M.S. degree with the Beijing University of Posts and Telecommunications, Beijing, China. His current research interests include deep learning and computer vision.

ZHAOJUN CUI received the B.E. degree in automation from the Beijing University of Posts and Telecommunications, Beijing, China, in 2020, where she is currently pursuing the M.S. degree in control science and engineering. Her current research interests include deep learning and computer vision.

JIDA TIAN received the B.E. degree in engineering from Luoyang Normal University, Luoyang, China, in 2013, and the M.S. degree in engineering from Xi'an Polytechnic University, Xi'an, China, in 2016. He is currently pursuing the Ph.D. degree with the Beijing University of Posts and Telecommunications. His research interests include machine learning and computer vision.

HUILING ZHOU was a Visiting Scholar with the University of Darmstadt, Germany, and the University of Manitoba, Canada. She is currently a Professor with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include technology and applications of the Internet of Things, machine learning, and data mining. She is currently focuses on the intelligent monitoring of stored-grain pests. She is a Reviewer of multiple academic journals, such as *Computers and Electronics in Agriculture*, *Crop Protection*, *Journal of Stored Products Research*, and *Biosystems Engineering*.