Intelligent Identification of Jute Pests Based on Transfer Learning and Deep Convolutional Neural Networks

Md Sakib Ullah Sourav[1] · Huidong Wang[1] ⬤

Abstract

Pest attacks pose a substantial threat to jute production and other significant crop plants. Jute farmers in Bangladesh generally distinguish between different pests that appear to be the same using their eyes and expertise, which isn't always accurate. We developed an intelligent model for jute pests identification based on transfer learning (TL) and deep convolutional neural networks (DCNN) to solve this practical problem. The proposed DCNN model can realize fast and accurate automatic identification of jute pests based on photographs. Specifically, the VGG19 CNN model was trained by TL on the ImageNet database. A well-structured image dataset of four dominant jute pests is also established. Our model shows a final accuracy of 95.86% on the four most vital jute pest classes. The model's performance is further demonstrated by the precision, recall, F1-score, and confusion matrix results. The proposed model is integrated into Android and IOS applications for practical uses.

1 Introduction

Jute is known as Bangladesh's "golden harvest" since it contributes roughly 4% of the country's GDP and generates about 5% of its foreign cash by exporting jute and jute diversified products [39]. More than 33% of the overall worldwide jute is produced in Bangladesh [1]. Likewise, in China, jute is considered one of its valuable plants, which makes a significant account of jute after Bangladesh and occupies the third rank in the world [2]. An extensive study on jute is sorely needed since people are nowadays more concerned about the environment. Jute is the raw material for making numerous environmentally friendly products such as jute-made bags, craftworks, textiles, apparel, etc. It is often regarded as an excellent natural substitute for nylon and polypropylene [1, 40]. It has the advantage of not posing threats to human health or natural environment. Jute is a versatile, long-lasting, reusable, low-cost

---

✉ Huidong Wang
huidong.wang@ia.ac.cn

1 School of Management Science and Engineering, Shandong University of Finance and Economics, Jinan, China

🦵 Springer

fiber that is preferable to synthetic fibers. Jute also has the advantages of being agro-based, annually generated, inexhaustible, biodegradable, and ecologically friendly [41]. However, the jute farmers have to face several challenges throughout the whole production period of Jutes. Among them, one of the vital challenges is pest attack.From seed roping to harvesting, along with other natural enemies, many insects and pests damage jute on a large scale, which causes lower overall productivity. Spilosoma Obliqua (Starctia Obliqua, Jute Hairy Caterpillar, Diacrisia Obliqua—other scientific names) and Yellow Mite have been reported as the most dangerous pests for jute that cause considerable loss in jute productivity [3]. Traditionally, Bangladeshi jute farmers mostly identify various types of pests through eyes using their attained experiences. Even after that, there is always a chance of making mistakes while accurately identifying the pests. In Bangladesh, generally, farmers do not feel comfortable consulting with the affiliated governmental agricultural personnel as long as they do not face any critical and unavoidable circumstances. Instead, they usually take precautions against pest attacks, mainly dependent on pesticide sellers. The traditional farmers believe the pesticides sellers more than anyone else. On the other hand, modern farmers are largely brand-oriented and would like to make decisions on their own. However, in both cases, as traditional crop pest identification methods highly rely on the experience of agricultural technicians and farmers, it implies the problems of low efficiency, high cost, and poor effect. Thus a potential risk of improper identification of pests increases, and hence, the burning concern is whether the proper precautions are implemented or not.

By the same token, in the past, the inspection of plant diseases and insect pests in China often used visual inspection methods [4]. This method requires the inspectors to have considerable professional knowledge and rich work experience, and the manual counting and statistical calculations are incredibly complicated throughout the inspection process. Visual manual inspection cannot meet the current requirements for real-time monitoring of plant diseases and insect pests, nor can it use real-time control methods. In particular, the symptoms of some plant diseases and insect pests are very similar in the early stage of damage, and it is difficult to distinguish them by visual inspection. Since the 1950s, various provinces have successfully managed and controlled agricultural pests and diseases in China, and various plant protection stations have accumulated a large amount of valuable information [5]. However, most of the data were drawn by hand in the early stage, and then the information was saved, summarized, and statistically analyzed. Information sharing and business management have brought a lot of inconveniences, and there are also significant problems in the rapid and effective identification of pests and diseases.

The agriculture-related development reports submitted by some developed countries in Europe and the United States long ago were listed as one of the critical areas of agriculture information research. They also successively carried out studies in the agricultural field and demonstrated those well in the past decade. However, in these countries, the application of intelligent agricultural information is mainly concentrated in agricultural resource monitoring and utilization, agricultural ecological environment monitoring, agricultural production precision management, agricultural product safety traceability, and cloud services. The development of agricultural pest identification is relatively slow [6–8].

Therefore, to solve the practical problem for jute farmers, our concern is to provide a rapid and accurate artificial intelligence (AI) driven solution to detect jute pests at the earliest pace. Crop pest identification is the starting point to solving the bottleneck that restricts modern agriculture's efficient and rapid development. AI technology is deeply integrated with modern agricultural technology by applying AI algorithms such as deep learning (DL) and transfer learning (TL) to crop pest identification, which might bring a revolutionary change to the prevention of harmful pests.

The intelligent jute pest identification model classifies jute pests only by receiving images as samples. The users will be able to identify pests of jute by taking a photo with a mobile phone in real-time. This will give the privilege to the farmers to identify the pests quickly and accurately. By doing this, they do not have to rely on any biased, ambiguous source and do not have to wait long to get feedback from specialists. This work uses advanced TL technology to migrate well-trained models to the intelligent identification of jute pests. By applying TL and convolutional neural networks (CNN) to modern agriculture, agricultural modernization has been accelerated, and the development of smart agriculture has been promoted.

The main contributions of this paper are as follows:

- This is the first-ever work focused primly on jute pest identification based on image processing and deep learning.
- We constructed a well-structured image dataset for the four most lethal pests for jute plants.
- We introduce the fast, accurate, and user-friendly way of identifying jute insects via transfer learning and deep convolutional neural networks (DCNN).

The remainder of this article is organized as follows. Section 2 elaborates on the related work that has been done in intelligent pest classification tasks with the help of the transfer learning network we used in this work. The methods that we used are displayed in Sect. 3. Training and validation procedures are discussed in Sect. 4. Section 5 illustrates the results, and discussions of the proposed model are explained in Sect. 6. Deployments of the model have been shown in Sect. 7. Finally, in Sect. 8, conclusions are presented.

2 Related Works

2.1 Insect Image Classification

CNN is trained on huge datasets like COCO, ImageNet, or Google Open Images, which contain a significant number of insect images. Lu et al. [25] suggested a rice disease detection approach based on DCNN methods to identify ten prevalent rice illnesses that have improved the convergence speed as well as the detection rate. In addition, a survey paper [8] reviewed thirty-three papers on automatic insect identification from digital photos. It showed various approaches, mainly use of different modified CNN models to recognize common insects that cause plant diseases. The transfer learning method has become more popular as a feature extractor or to fine-tune the deep networks [13–15]. Fine-tuning is a transfer learning technique that requires some learning but is significantly faster and more accurate than created models [26].

A research study [10] by Thenmozhi and Reddy shows the categorization of insect species using aDCNN model on three publicly accessible insect datasets. The first insect dataset employed was the National Bureau of Agricultural Insect Resources (NBAIR), which comprises 40 classes of field crop insect photos, whereas the second and third datasets (Xie1, Xie2) contain 24 and 40 categories of insects, respectively. For insect classification, the suggested model was assessed and compared to pre-trained deep learning architectures such as AlexNet, ResNet, GoogLeNet, and VGGNet. TL was used to fine-tune the pre-trained models in this work.

Kasinathan et al. [42] created a pest detection technique for Wang, Xie, Deng, and IP102 datasets that use foreground extraction and contour identification to locate insects in a highly complicated backdrop. They used shape features and machine learning techniques such as artificial neural networks (ANN), support vector machines (SVM), k-nearest neighbors

(KNN), Naive Bayes (NB), and CNN models to conduct the experiment, which yielded the highest classification rate of 91.5% and the best detection performance with less computation time. Insects from these datasets come from various places, including tea plants and other plants from Europe and Central Asia.

A couple of months ago, Abeywardhana et al. [43] produced a dataset with an unbalanced and restricted number of photos for six taxa of the Cicindelinae subfamily (tiger beetles) of the order Coleoptera. The authors put much effort into photographing tiger beetles from various sources, perspectives, and scales, although they recognize that tiger beetle categorization can be difficult even for a trained human eye. Francesco et al. [9] demonstrate an insect image classification task with a custom dataset having 11 insect classes using deep CNNs. This automatic insect recognition model was developed using TL on a CNN architecture called MobileNet. They deploy this model into an Android app with a user feedback feature to solve the "open set" problem. Yet this work does not mention the reason behind choosing those particular insect classes.

However, to the best of our knowledge, only a few extant studies specialize in intelligent insects' identification or plant disease categorization, focusing on key insects or crops of regional significance.

2.2 Visual Geometry Group (VGG)

Simonyan and Zisserman [18] presented the VGG network designed in very deep convolutional networks for large-scale image recognition. Since VGG was proposed in 2014, it has drawn much interest from researchers from various domains. Inhae Ha et al. [32] present a new image-based indoor localization method using building information modeling (BIM) on a pre-trained VGG network for image feature extraction for the similarity evaluation of two different types of images (BIM rendered and real images). In their study, Qing Guan et al. [33] exploited a VGG-16 DCNN model to differentiate papillary thyroid carcinoma (PTC) from benign thyroid nodules using cytological images.

In recent days, following the emergence of COVID-19, researchers from health science backgrounds have come up with different approaches to prompt identification of this deadly virus with image samples using various techniques. Tan et al. [34] introduce a new assisted diagnosis method for COVID-19 based on super-resolution reconstructed pictures and a CNN. To begin, the SRGAN neural network, a generative adversarial network for single image super-resolution, is utilized to create super-resolution pictures from original chest CT scans. The VGG16 neural network is then used to classify COVID-19 and Non-COVID-19 pictures from super-resolution chest CT scans. Likewise, Shui-Hua Wang et al. [35] constructed an attention-based VGG-style network for COVID-19 diagnosis using a previously presented chest CT dataset, which exhibits effective results in detecting COVID-19 disorders. The same group also focused on solving the complex task of detecting Alzheimer's disease by proposing a VGG-inspired deep-learning model in this work [36].

VGGs are also used in other areas, such as smart grids and breast cancer detection. The use of a partial discharge (PD) signal paired with a dual-input VGG-based CNN to forecast the position of the pollution layer on 11 kV polymer insulators subjected to alternating current for smart grid systems was investigated by Vigneshwaran et al. [37]. GSB Jahangeer and Dhiliphan Rajkumar [38] sought to discover the early detection of breast cancer through a combination of a series of network and VGG16-based models.

To sum up, along with the above works, other notable significant tasks have been done based on different DL and TL algorithms by the interested and passionate researchers of

various domains, yet no specific prior work on jute insects. Our motivation for this effort stems from the fact that jute is much more essential in both commercial and environmental aspects for the nations where it has been produced, namely- Bangladesh, India, China, Myanmar, Uzbekistan, Nepal, Thailand, Sudan, Egypt, and many more. Furthermore, a reliable and well-structured dataset for the classification of jute insects is required, considering that even the most experienced taxonomist requires a comprehensive morphological examination to identify species at certain degrees of detail.

3 Method

We followed the transfer learning method so that our model might be well trained. To achieve so, the VGG19 pre-trained CNN model has been picked for our model development.

3.1 Transfer Learning (TL)

Training a neural network (NN) from scratch for image recognition is an issue that necessitates a lot of resources, such as computational power, a significant quantity of data, and time. It frequently results in overfitting owing to too many training cycles [11]. Many neural network models have a large number of hyperparameters that must be tweaked periodically during the training and verification process. To solve these problems, we used the TL technique that borrows CNN architecture with its pre-trained parameters. We can easily reach the target accuracy when we train using our own data on top of the pre-trained parameters.

TL is a brand-new machine learning paradigm that aims to tackle challenging learning problems in the target domain with few or no labeled examples [12]. It enables cross-domain knowledge transmission (Fig. 1). Knowledge is contained in network weights in neural networks, and TL is applied to CNN for image recognition. The concept is that the characteristics learned in the model's initial few layers are generally independent of the training domain. The more similar the domains are, the more characteristics may be shared, and the less training is needed.

The application of TL in our study to distinguish jute insects from pictures focuses on two aspects: First, a CNN is trained using a Softmax classifier in ImgaeNet (a big visualization database for visual recognition), which allows the network output result to be transformed into the similarity between the image and the projected classification result. The CNN training
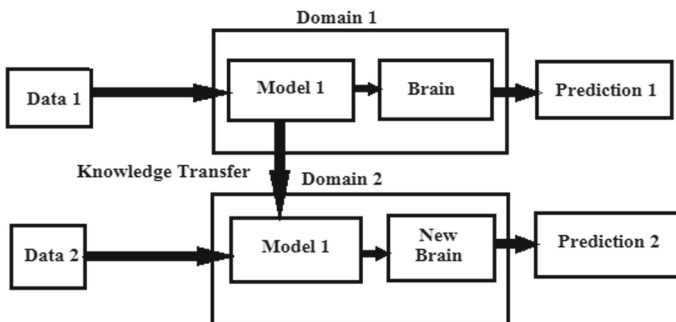

Fig. 1 Transfer learning principle

Fig. 2 A visualization of the VGG architecture

model was then fine-tuned, using parameters tailored to the features of the agricultural pests in this study.

3.2 VGG19 Architecture

We chose to train our network with the VGG pre-trained CNN model. VGG has a thorough knowledge of how shape, color, and structure define a picture. Though there are some lightweight networks such as MobileNet [21], YOLO3 [31], and others, we preferred VGG since the VGG architecture is straightforward and has a low loss rate, despite being large size. VGGs have already established themselves as a prominent deep neural network trained on millions of photos with the capability of solving many challenging classification problems [16, 28–30]. It is evident in the work done by Nilanjan Dey et al. [17] that the modified VGG19 exhibits superior capability compared to other popular TL networks. By taking all these aspects mentioned above into careful consideration, we chose the VGG19 among the variants of VGG to train our model.

Figure 2 shows the layout of the VGG19 CNN model architecture. This is a really huge network, with over 201 million parameters in total. We can use the ImageNet database to load a pre-trained VGG19 network that has been trained on over a million photos. With only 33 convolutional layers stacked on top of each other in increasing depth, this network is characterized by its simplicity.

4 Training and Validation

4.1 Data Preprocessing

Obstacles that every image classification algorithm must overcome include picture occlusion, lighting conditions, image distortion, perspective changes, size variations, intra-class alterations, background clutter, and so on [18]. We attempted to overcome these difficulties using data-driven approaches in supervised learning.

At the first stage of the training part, we created two folders naming—training and validation. Under each of these two folders, we created four subfolders for the four classes we worked on. The data set is split into the training and validation sets in a random but repeatable manner. Each category has gotten 90% of its images for the training portion and the rest 10% of the images went with the validation part.

Then we introduced the VGG19 algorithm in the coding to read our dataset and the data class. The VGG19 algorithm adapted our four data classes by replacing them in its dense layer, where it found 100,356 trainable parameters in a total of 20,124,740 parameters. We used ImageDataGenerator class that lets us augment the images in real-time while the model

is in training. We set the rescaling to 1/255, enabling less computation time and a zoom range of 0.2. This will not only make the model robust but will also save up on the overhead memory.

4.2 Dataset

The algorithmic model's data collection should be as diversified as feasible and capable of covering all conceivable versions. In this work, we used a script to collect all photographs of insects of our interest from Google Images and combined them with photos processed in the lab. Our dataset is composed of 4 classes. Each class counts a mean of 384 for a total amount of 1535 images. The dataset was constructed in such a way that the number of examples within the classes is balanced (Fig. 3).

The classes included in the dataset are the following: Spilosoma Obliqua, Jute Stem Weevil, Yellow Mite, and Field Cricket. The number of training Field Cricket images is 347, Jute Stem Weevil images is 324, Spilosoma Obliqua images is 390, and Yellow Mite images is 329. The total number of training samples is 1390. The number of validation Field Cricket images is 33, Jute Stem Weevil images is 36, Spilosoma Obliqua images is 39 and Yellow Mite images is 37. The total number of validation samples is 145. The classes were chosen because they are the most dangerous and deadly in terms of the damage to Jute (Fig. 4).
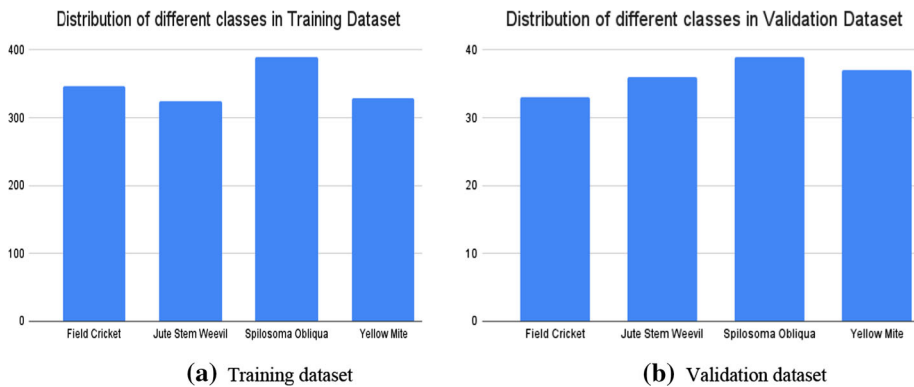

(a) Training dataset **(b)** Validation dataset

Fig. 3 Training and validation datasets for four classes

Spilosoma Obliqua   Jute Stem Weevil   Yellow Mite   Field Cricket

Fig. 4 Four classes of jute insects of our interest

4.3 Training Settings

To train our neural network, we used Keras5 and TensorFlow6 as the backend. Keras is a deep learning open-source package built-in Python. It includes several deep learning models, most of which are employed in this model. In addition, Adam optimizers with a default learning rate of 0.001 and the categorical cross-entropy function as the loss function were utilized to get the best Softmax classifier results. Furthermore, given the nature of the challenge, we evaluated the findings using accuracy as a metric. We ran 15 epochs with a batch size of 64. Every training run comes to an early end, and the validation set was constructed using 10% of the training set images for each class. Google Research's Colaboratory Notebook was used to train and evaluate the model. The code resources are made publicly available here.[1]

4.4 Features Extractor via VGG19

Feature extraction involves the methods for selecting and/or combining variables to form features, hence minimizing the quantity of data that has to be processed [19]. To improve the performance of classification, feature extraction and dimension reduction are necessary. To improve the effectiveness of the classifier, feature extraction seeks out the most compressed and informative set of features or unique patterns. As the first stage in model training, we employed the transfer learning approach to train the network, and as a feature extractor, we used the VGG19 CNN model that has been pre-trained on ImageNet. At the initial stage, we import the VGG19 model from Tensorflow Keras. The preprocess input module is imported to scale pixel values properly for the VGG19 model, and the image module is loaded to preprocess the picture object. The NumPy module is used to process arrays. The pre-trained weights for the ImageNet dataset are then fed into the VGG19 model.

A convolutional layer is followed by one or more dense (or completely linked) layers in the VGG19 model. 'Include_top' allows you to choose whether or not to include the last thick layers. When the model is loaded, 'False' indicates that the last thick layers are not included. The feature extraction component of the model runs from the input layer to the final max-pooling layer (labeled by $7 \times 7 \times 512$), whereas the classification part of the model runs from the input layer to the last max-pooling layer (labeled by $7 \times 7 \times 512$). After defining the model, we must load the input picture with the model's predicted size, which is 224×224 in this example. The image PIL object must then be transformed to a NumPy pixel data array and enlarged from a 3D to a 4D array with the dimensions (samples, rows, cols, channels), where we only have shown one sample (Fig. 5). After that, the pixel values must be scaled suitably for the VGG model.

4.5 Fine-Tuning of VGG19

In general, fine-tuning refers to making little tweaks to a process in order to get the intended output or performance [20]. Fine-tuning is the process of fine-tuning or changing a model previously trained for one job to make it execute a related task. A deep learning network that can distinguish automobiles, for example, may be fine-tuned to detect trains. In DL, fine-tuning entails programming another comparable DL process with weights from a prior DL method. Each neuron in one layer is connected to every neuron in the next layer in the NN by

---

[1] https://github.com/Sakibsourav019/Jute-Insect-Classification-using-Transfer-Learning-and-Deployment-in-Android-App.
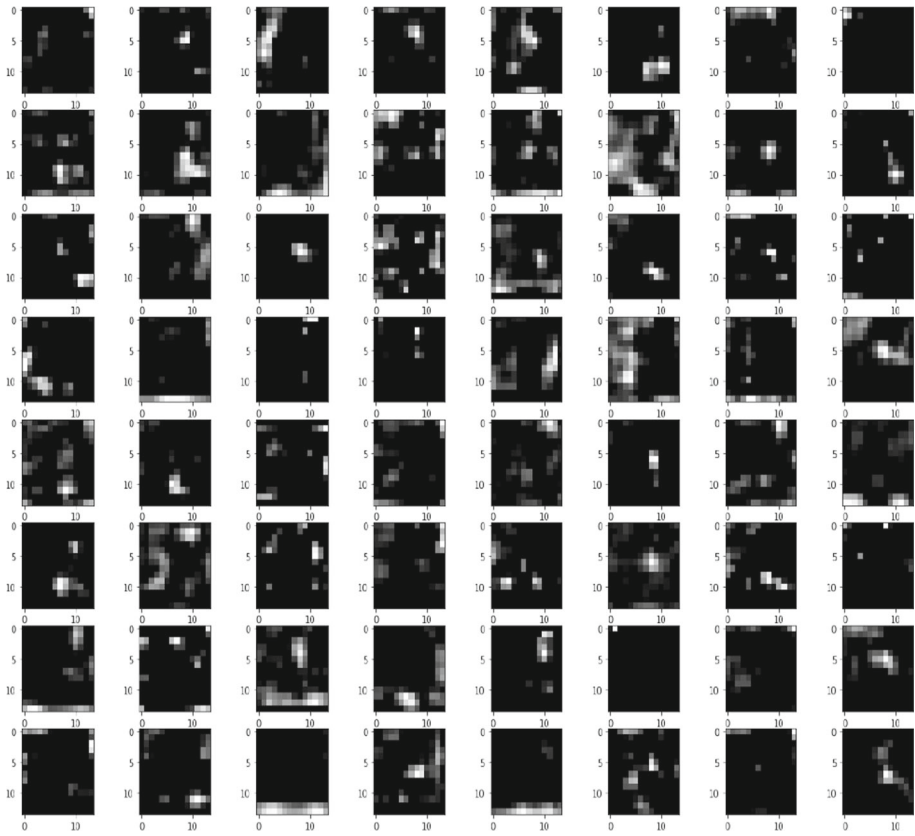
Fig. 5 Visualization of feature maps by VGG19 on our data

weights. Because it already has crucial knowledge from a previous deep learning algorithm, the fine-tuning time dramatically decreases compared to developing and processing a new deep learning algorithm. While fine-tuning is useful for training new deep learning algorithms, it can only be employed when the datasets of an existing model and the new deep learning model are similar.

The previous phase's findings advised that we should employ a small batch size and a slow learning rate. We used the Softmax classifier to replace the final layer of VGG19 (Fig. 6) on top of ImageNet given by Keras with our four output classes and removed two dense layers before the last prediction layer. As previously declared, ImageNet has a large number of insect photos, so we can confidently assume that the previously learned characteristics will provide decent outcomes at this stage.
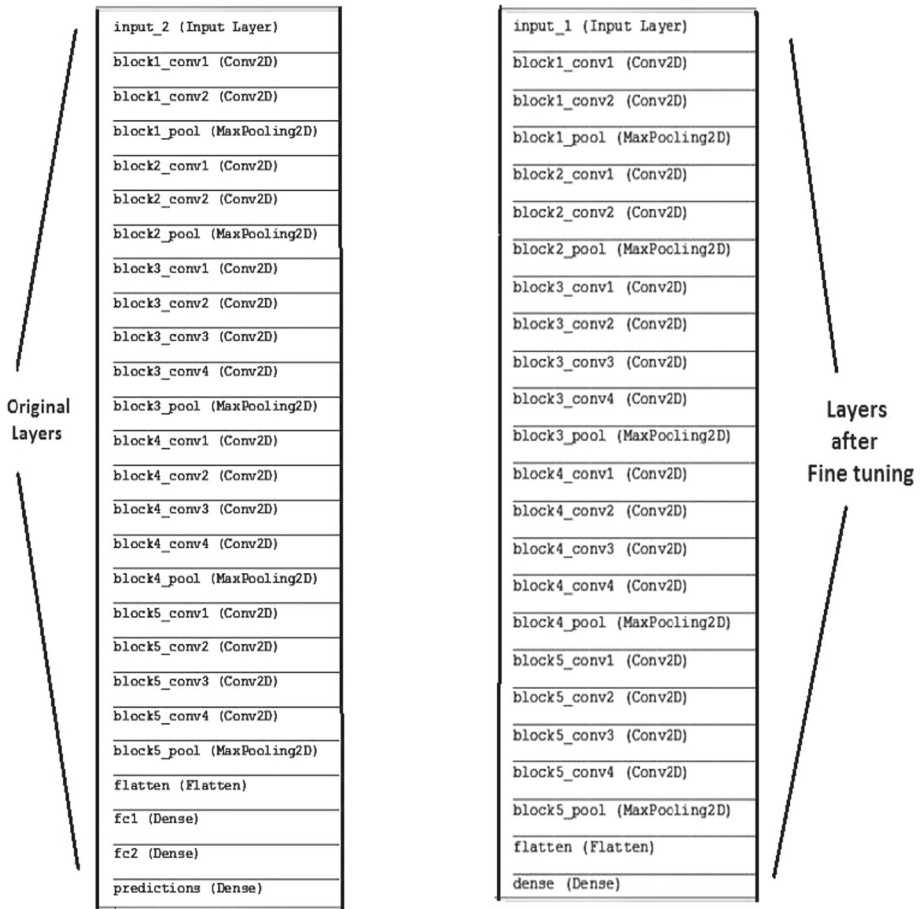
Fig. 6 Fine-tuning of VGG19 by replacing and training the last layer

5 Results

In terms of execution and performance, we have noticed that our intelligent pest identification model is consistently producing excellent results. Given the vast number of parameters such networks incorporate, this should be no surprise. Because of the unique nature of the challenge and the fact that we created a custom dataset, we are unable to compare our results to previous work.

5.1 Accuracy and Loss

Classification accuracy is a salient evaluation metric used in our model, given by Eq. (1).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP} \tag{1}$$

The letters TP, FP, FN, and TN represent true positive, false negative, and true negative, respectively. If the insect in the photograph is correctly classified, it is categorized as TP; otherwise, it is labeled as FN. If the categorization is done improperly, the insect that is not there in the photograph is classified as TN; otherwise, it is classified as FP.

Training and validation accuracy

(a)

Training and validation loss
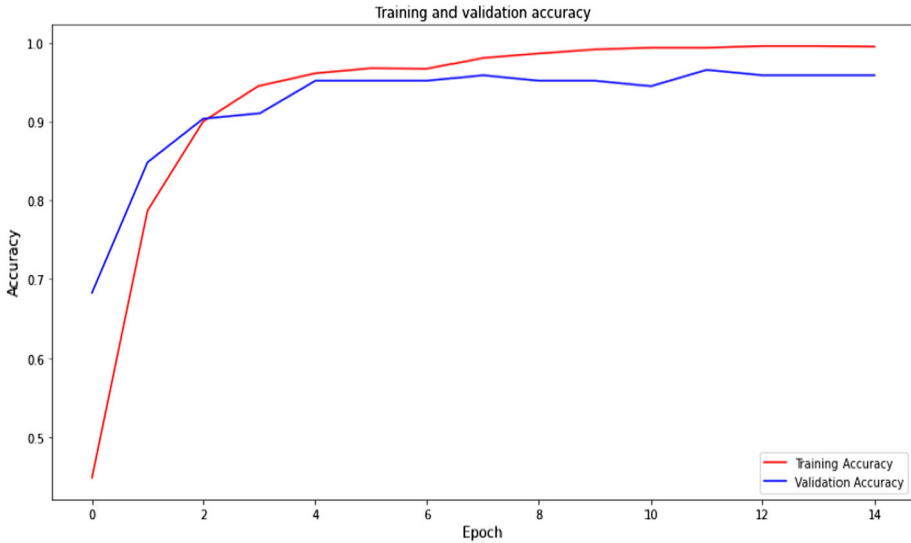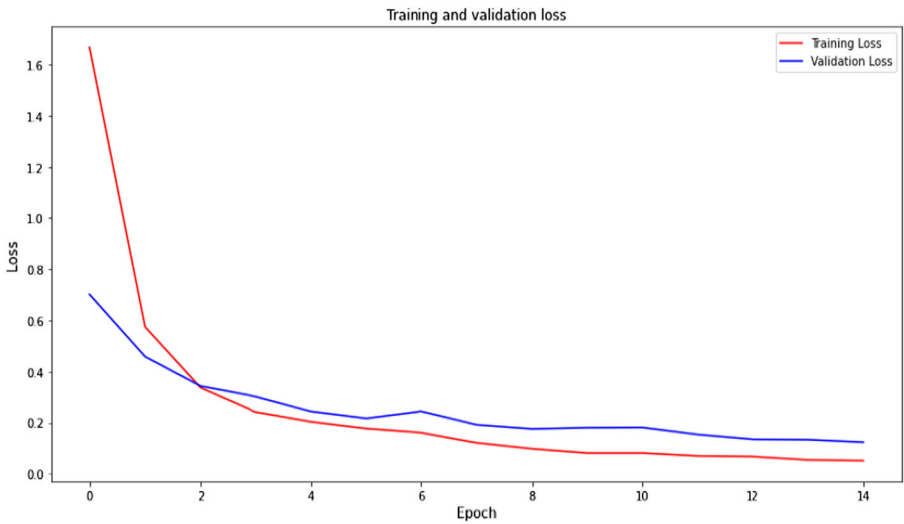
(b)

Fig. 7 Accuracy (**a**) and loss (**b**) of the model

Categorical cross-entropy is used as the loss function of our model as shown in Eq. (2), which employs Softmax activations in the output layer.

$$\text{Loss} = \sum_{i=1}^{N} \sum_{j=1}^{K} t_{ij} \ln y_{ij} \tag{2}$$

N indicates the number of insect samples, K is the number of insect classes, t_{ij} indicates that i_{th} insect sample belongs to j_{th} insect class and y_{ij} represents the output for insect sample i for insect class j.

During the training period of the CNN network, we randomly set fifteen epochs in total. From epoch one to epoch fifteen, training and validation accuracy have increased in a consistent manner, hence giving us the best model with a final validation accuracy of 95.86%, as illustrated in Fig. 7a. In addition, from the first epoch to epoch fifteen, the loss of training and validation has decreased, as shown in Fig. 7b.

5.2 Precision, Recall and F1-score

Precision is a metric used to quantify how many correct positive forecasts have been made. It is determined as the number of true positives divided by the total number of true positives and false positives (Eq. (3)). When the model predicts an individual as positive, precision informs us how much we can trust it.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

The Recall is calculated by dividing the number of true positive elements by the total number of positively categorized units (row sum of the actual positives). False Negative components, in particular, are those the model labeled as negative but actually positive. Precision and recall make it possible to assess the performance of a classifier in the minority class [44].

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

The F1 score is a weighted average of the accuracy and recall scores. When the F1 score is 1, the model is deemed perfect; when it is 0, the model is considered a complete failure. It's a metric for checking the quality of a classifier's prediction. Because it includes both precision and recall, it is frequently the metric of choice for most individuals.

$$F1 = 2 \times \frac{\text{Precision} \times \text{recall}}{\text{Precision} + \text{recall}} \tag{5}$$

From Table 1, we can observe the classification report of our model that includes precision, recall and F1-score of each class. The model exhibits satisfactory indicators in all these three parameters.

Table 1 Classification report of our model

Labels	Precision	Recall	F1-Score
Field Cricket	1.0	0.9	0.95
Jute Stem Weevil	1.0	1.0	1.0
Spilosoma Obliqua	0.91	1.0	0.95
Yellow Mite	1.0	1.0	1.0
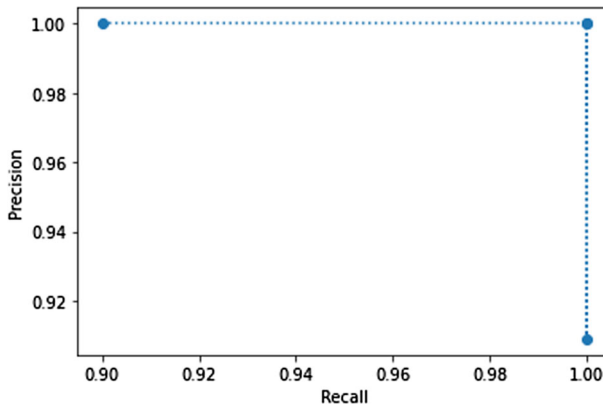
Fig. 8 P-R curve of intelligent jute pest identification model

5.3 P-R Curve

A precision-recall curve (also known as a PR Curve) is a graph that shows the precision (y-axis) and recall (x-axis) for various probability thresholds. A competent model is represented as a curve that bends in the direction of a coordinate (1, 1). A horizontal line on the plot representing a no-skill classifier will have a precision proportionate to the number of positive instances in the dataset. Figure 8 shows the p-r curve of our model with an ideal (1, 1) coordinate.

5.4 Confusion Matrix

To further evaluate our model performance, we passed 40 random test images of the data classes we worked on to our model that are from outside of the train and validation dataset.

In Fig. 9, if we look at the performance matrix of our model, we can see that in almost every case of the four classes of jute pests, the predicted and true labels are the same (except for one wrong result in Jute Stem Weevil). That indicates that our model has been doing very well in classifying and distinguishing the four major pests of jute.
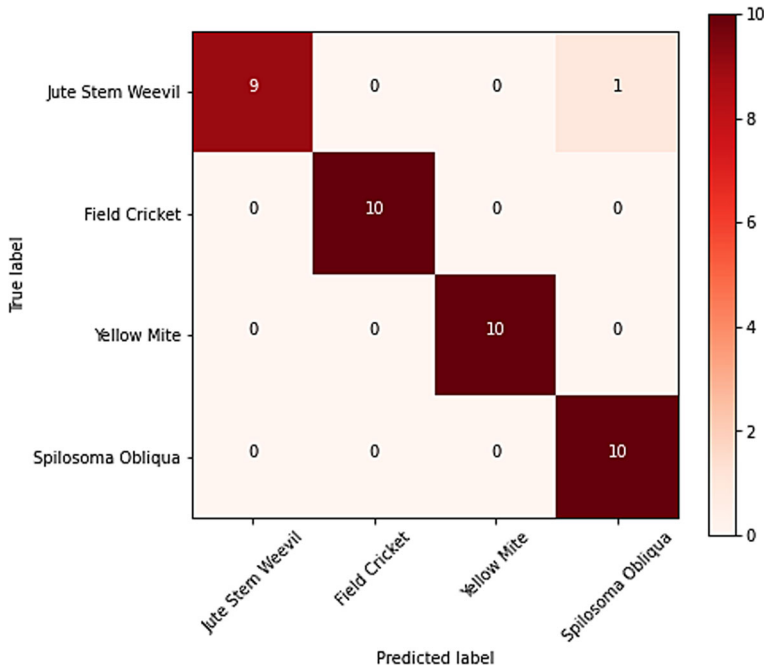
Fig. 9 Performance Matrix of the model

6 Discussions

The results presented in Sect. 5 demonstrate that the image-based model can be used to detect jute pests successfully. The accuracy, recall, and f1-score all stay consistent, indicating that our suggested strategy is suitable. There are no particular criteria on how many epochs are ideal to pick. During the CNN training, we didn't choose to go for too many steps and restricted the epoch size to fifteen only as from the fourth epoch, the accuracy had shown a consistency of ninety-five percent or more till the end. The model training took a substantial amount of time, which is why we keep the batch size at sixty-four.

Nevertheless, generally speaking, the network's accuracy may decline as the number of classes grows. However, there are other options available to train the model. The design of our network can be altered by utilizing similar sorts of other advanced networks such as MobileNet [21], AlexNet [22], ShuffleNet [23], or GoogLeNet [27]. The claim would be that the dataset's affluence should be improved in all situations to improve the recognition rate. This is a slogan shared by all deep-learning algorithms, and it is similar to the mantra of classical AI, which states that any system developed by artificial intelligence may reach human intelligence if a complicated enough local function is built inside it.

Human has a complexly amalgamated ability that is formed in combination of some strange and absurd components while they look at objects to identify things. This is still a concern in terms of the development of sophisticated computer vision algorithms and models to some extent. Probably, our system has certain drawbacks in that area that prevent it from achieving a hundred percent accuracy in recognizing pest images.

Furthermore, agricultural pest photos are frequently murky and indistinct due to the poor density of cameras. This makes it harder to identify and monitor pests. A generative adversarial network (GAN) containing quadric-attention, residual, and dense fusion methods was used to modify low-resolution pest photos in a prior study [24]. This method might be integrated with our model in future studies to improve our model's performance. Along with the implemented image classification feature in this model, the further necessary instructions can also be included for each particular pest as future up-gradation of this model.

7 Deployments

We have deployed this intelligent jute pest identification model as an Android and IOS app in the flutter software platform by Google, and the web-based information system can also be developed by the same software platform.

Figure 10 shows the Pest identification sample library and the main interface of the Android app. This app will allow users to capture a new photo of pests in the app or import images of insects from the phone gallery. The default input size of images for this model is 224 × 224. With the help of the built-in Keras input preprocessing function, the input image of whatever size is resized to 224 × 224 before passing it to the model. After getting the input, the app will then return to its user with the corresponding name of the jute pest along with the percentage of confidence. With reduced computing time, the detection task is delivered by this app. In Fig. 8, the model is saying with 93% confidence that this is an image of a Spilosoma Obliqua.
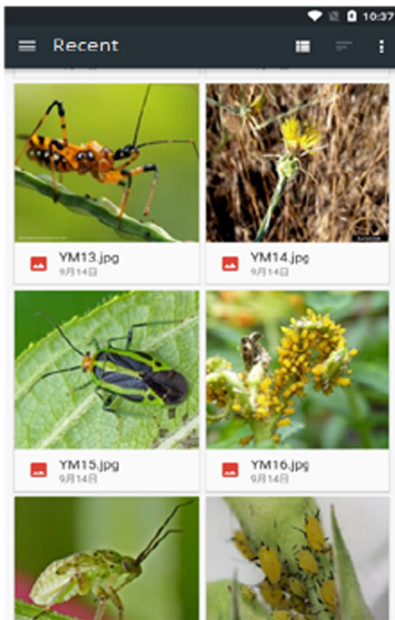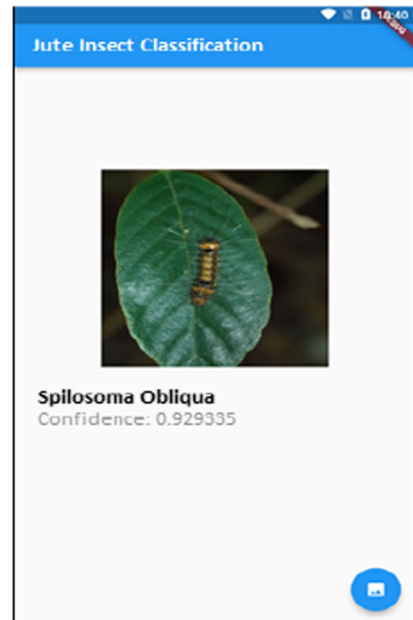
(a) Pest identification sample library　　(b) Main interface of the app

Fig. 10 Intelligent jute pest identification app for Android system

8 Conclusions

The production of jute, which has geographical importance to many of its producing countries, is largely hampered by pest attacks. Due to the matching appearance in the early stages, farmers have to struggle a lot while correctly classifying and identifying all the varieties of jute pests. This work proposes a model to solve this practical issue based on a DCNN (deep convolutional neural network) by integrating the transfer learning method using the VGG19 pre-trained network. Beforehand, we constructed a customized image dataset of vital tiny insects of jute, on which we trained our model. The higher accuracy rate and consistent indications of other performance metrics depict the validity of the real-life applicability of this model. The proposed model was incorporated into Android and IOS apps to accomplish the development goals of smart agriculture.

Declarations

Conflict of interest The authors declare that they have no known conflict of interest.

References

1. Rahman S, Kazal MMH, Begum IA, Alam MJ (2017) Exploring the future potential of jute in Bangladesh. Agriculture 7(12):96
2. Zhang L, Ma X, Zhang X, Xu Y, Ibrahim AK, Yao J et al (2021) Reference genomes of the two cultivated jute species. Plant Biotechnol J 19(11):2235–2248
3. Khan MMH (2018) Evaluation of different mutants against insect and mite pests with natural enemies in coastal jute ecosystem. J Asiatic Soc Bangladesh Sci 44(1):23–33
4. Li K, Yang QH, Zhi HJ, Gai JY (2010) Identification and distribution of soybean mosaic virus strains in southern China. Plant Dis 94(3):351–357
5. Wan FH, Yang NW (2016) Invasion and management of agricultural alien insects in China. Annu Rev Entomol 61:77–98
6. Balan V, Chiaramonti D, Kumar S (2013) Review of US and EU initiatives toward development, demonstration, and commercialization of lignocellulosic biofuels. Biofuels Bioprod Biorefin 7(6):732–759
7. Patrício DI, Rieder R (2018) Computer vision and artificial intelligence in precision agriculture for grain crops: a systematic review. Comput Electron Agric 153:69–81
8. Júnior TDC, Rieder R (2020) Automatic identification of insects from digital images: a survey. Comput Electron Agric 178:105784
9. Visalli F, Bonacci T, Borghese NA (2021) Insects image classification through deep convolutional neural networks. In: Progresses in artificial intelligence and neural systems. Springer, Singapore, pp 217–228
10. Thenmozhi K, Reddy US (2019) Crop pest classification based on deep convolutional neural network and transfer learning. Comput Electron Agric 164:104906
11. Sun C, Ma M, Zhao Z, Tian S, Yan R, Chen X (2018) Deep transfer learning based on sparse autoencoder for remaining useful life prediction of tool in manufacturing. IEEE Trans Ind Inf 15(4):2416–2425
12. Shao L, Zhu F, Li X (2014) Transfer learning for visual categorization: a survey. IEEE Trans Neural Netw Learn Syst 26(5):1019–1034
13. Lu J, Behbood V, Hao P, Zuo H, Xue S, Zhang G (2015) Transfer learning using computational intelligence: a survey. Knowl-Based Syst 80:14–23
14. Li X, Zhang W, Ding Q, Li X (2019) Diagnosing rotating machines with weakly supervised data using deep transfer learning. IEEE Trans Ind Inf 16(3):1688–1697
15. Guo L, Lei Y, Xing S, Yan T, Li N (2018) Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data. IEEE Trans Ind Electron 66(9):7316–7325
16. Meena KB, Tyagi V (2021) Distinguishing computer-generated images from photographic images using two-stream convolutional neural network. Appl Soft Comput 100:107025

17. Dey N, Zhang YD, Rajinikanth V, Pugalenthi R, Raja NSM (2021) Customized VGG19 architecture for pneumonia detection in chest X-rays. Pattern Recogn Lett 143:67–74

18. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556

19. Jogin M, Madhulika MS, Divya GD, Meghana RK, Apoorva S (2018) Feature extraction using convolution neural networks (CNN) and deep learning. In: 2018 3rd IEEE international conference on recent trends in electronics, information & communication technology (RTEICT). IEEE, pp 2319–2323

20. Tajbakhsh N, Shin JY, Gurudu SR, Hurst RT, Kendall CB, Gotway MB, Liang J (2016) Convolutional neural networks for medical image analysis: full training or fine tuning? IEEE Trans Med Imaging 35(5):1299–1312

21. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4510–4520

22. Yu W, Yang K, Bai Y, Xiao T, Yao H, Rui Y (2016) Visualizing and comparing AlexNet and VGG using deconvolutional layers. In: Proceedings of the 33rd international conference on machine learning

23. Ma N, Zhang X, Zheng HT, Sun J (2018). Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European conference on computer vision (ECCV), pp 116–131

24. Dai Q, Cheng X, Qiao Y, Zhang Y (2020) Agricultural pest super-resolution and identification with attention enhanced residual and dense fusion generative and adversarial network. IEEE Access 8:81943–81959

25. Lu Y, Yi S, Zeng N, Liu Y, Zhang Y (2017) Identification of rice diseases using deep convolutional neural networks. Neurocomputing 267:378–384

26. Mohanty SP, Hughes DP, Salathé M (2016) Using deep learning for image-based plant disease detection. Front Plant Sci 7:1419

27. Tang P, Wang H, Kwong S (2017) G-MS2F: GoogLeNet based multi-stage feature fusion of deep CNN for scene recognition. Neurocomputing 225:188–197

28. Bi Z, Yu L, Gao H, Zhou P, Yao H (2021) Improved VGG model-based efficient traffic sign recognition for safe driving in 5G scenarios. Int J Mach Learn Cybern 12(11):3069–3080

29. Kulwa F, Li C, Zhang J, Shirahama K, Kosov S, Zhao X, et al (2022) A new pairwise deep learning feature for environmental microorganism image analysis. Environ Sci Pollut Res, pp 1–18

30. Chen JR, Chao YP, Tsai YW, Chan HJ, Wan YL, Tai DI, Tsui PH (2020) Clinical value of information entropy compared with deep learning for ultrasound grading of hepatic steatosis. Entropy 22(9):1006

31. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767

32. Ha I, Kim H, Park S, Kim H (2018) Image retrieval using BIM and features from pretrained VGG network for indoor localization. Build Environ 140:23–31

33. Guan Q, Wang Y, Ping B, Li D, Du J, Qin Y et al (2019) Deep convolutional neural network VGG-16 model for differential diagnosing of papillary thyroid carcinomas in cytological images: a pilot study. J Cancer 10(20):4876

34. Tan W, Liu P, Li X, Liu Y, Zhou Q, Chen C et al (2021) Classification of COVID-19 pneumonia from chest CT images based on reconstructed super-resolution images and VGG neural network. Health Inf Sci Syst 9(1):1–12

35. Wang SH, Fernandes S, Zhu Z, Zhang YD (2021) AVNC: attention-based VGG-style network for COVID-19 diagnosis by CBAM. IEEE Sensors J

36. Wang SH, Zhou Q, Yang M, Zhang YD (2021) ADVIAN: Alzheimer's disease VGG-inspired attention network based on convolutional block attention module and multiple way data augmentation. Front Aging Neurosci 13:313

37. Vigneshwaran B, Maheswari RV, Kalaivani L, Shanmuganathan V, Rho S, Kadry S, Lee MY (2021) Recognition of pollution layer location in 11 kV polymer insulators used in smart power grid using dual-input VGG convolutional neural network. Energy Rep 7:7878–7889

38. Jahangeer GSB, Rajkumar TD (2021) Early detection of breast cancer using hybrid of series network and VGG-16. Multimed Tools Appl 80(5):7853–7886

39. Islam MM, Ali MS (2017) Agronomic research advances in jute crops of Bangladesh. AASCIT J Biol 3(6):34–46

40. Ferdous J, Hossain MS, Alim MA, Islam MM (2019) Effect of field duration on yield and yield attributes of Tossa Jute varieties at different agroecological zones. Bangladesh Agron J 22(2):77–82

41. Akter S, Sadekin MN, Islam N (2020) Jute and jute products of Bangladesh: contributions and challenges. Asian Bus Rev 10(3):143–152

42. Kasinathan T, Singaraju D, Uyyala SR (2021) Insect classification and detection in field crops using modern machine learning techniques. Inf Proc Agric 8(3):446–457

43. Abeywardhana DL, Dangalle CD, Nugaliyadde A, Mallawarachchi Y (2022) An ultra-specific image dataset for automated insect identification. Multimed Tools Appl 81(3):3223–3251
44. He H, Ma Y (2013) Imbalanced learning: foundations, algorithms, and applications. Wiley-IEEE Press