Computers and Electronics in Agriculture 164 (2019) 104906

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Original papers

Crop pest classification based on deep convolutional neural network and transfer learning

K. Thenmozhi, U. Srinivasulu Reddy*

Machine Learning and Data Analytics Lab, Department of Computer Applications, National Institute of Technology, Tiruchirappalli 620015, India

ARTICLE INFO

Keywords:
Insect classification
Field crops
Convolutional neural network
Deep learning
Transfer learning

ABSTRACT

The growth of most important field crops such as rice, wheat, maize, soybean, and sugarcane are affected due to attack of various pests and the crop production is reduced due to different types of insects. The classification and identification of all types of crop insects correctly is a difficult task for the farmers due to the similar appearance in the earlier stage of crop growth. To address this issue, Convolutional neural network (CNN) with deep architectures is being applied as it performs automatic feature extraction and learns complex high-level features in image classification applications. This study proposed an efficient deep CNN model to classify insect species on three publicly available insect datasets. The National Bureau of Agricultural Insect Resources (NBAIR) dataset used as first insect dataset that consists of 40 classes of field crop insect images, while the second and third dataset (Xie1, Xie2) contains 24 and 40 classes of insects respectively. The proposed model was evaluated and compared with pre-trained deep learning architectures such as AlexNet, ResNet, GoogLeNet and VGGNet for insect classification. Transfer learning was applied to fine-tune the pre-trained models. The data augmentation techniques such as reflection, scaling, rotation, and translation are also applied to prevent the network from overfitting. The effectiveness of hyperparameters was analysed in the proposed model to improve accuracy. The highest classification accuracy of 96.75, 97.47, and 95.97% was achieved in proposed CNN model for NBAIR insect dataset (40 classes), Xie1 (24 classes) insect dataset and Xie2 (40 classes) insect dataset respectively. The results demonstrated that the proposed CNN model is effective in classifying various types of insects in field crops than pre-trained models and can be implemented in the agriculture sector for crop protection.

1. Introduction

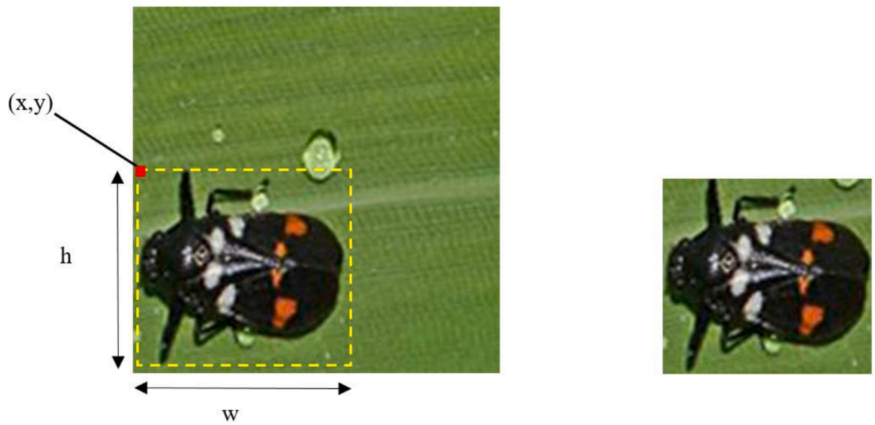
Crop pest identification and classification represent one of the major challenges in the agriculture field. Insects cause damage to crops and mainly affect the productivity of crop yield. Classification of insects is a difficult task due to the complex structure and having a high degree of similarity of the appearance between distinct species. It is necessary to recognize and classify insects in the crops at an early stage, especially to prevent the spread of insects, which cause crop diseases by selecting effective pesticides and biological control methods. Traditional manual identification of insects is typically labour-intensive, time-consuming and inefficient. The vision-based computerized system of image processing using machine learning was developed for accurate classification and identification of insects to overcome these problems in agriculture research field (Martineau et al., 2017).

Wen et al. (2009) proposed an effective local feature based insect classification for orchard insects using six machine learning algorithms. The maximum classification accuracy of 89.5% was observed and insect species are misclassified due to similar insect species, various poses of wings and bodies. An automatic insect identification system was developed by Wang et al. (2012) by defining seven geometrical features and the classification results of Artificial neural networks (ANNs) and support vector machine (SVM) provide good results only for less number of classes of insects. In machine learning, the classification accuracy mainly depends on the design of extracted features and only the best features are selected to pass over to the machine learning algorithm, which increases the computational complexity. Further, the accuracy is improved by applying deep learning, which is a branch of machine learning for classifying larger image data sets. Deep learning performs automatic feature extraction from raw data that reduces the challenges in handcrafted features and solve more complex problems, especially for image classification.

In recent years, deep learning models based on CNN are extensively used as a powerful class of models for classification of images in a variety of problems in agriculture field such as plant disease recognition, fruit classification, weed identification and crop pest classification

* Corresponding author.
E-mail address: usreddy@nitt.edu (U. Srinivasulu Reddy).

https://doi.org/10.1016/j.compag.2019.104906
Received 8 June 2019; Received in revised form 16 July 2019; Accepted 21 July 2019



(i) Input insect image *(Callitettix versicolor (Fabricius))* (ii) Cropped output

Fig. 1a. Pre-processed cropped insect image (NBAIR dataset).

(Kamilaris and Prenafeta-Boldu, 2018). Convolutional neural network models were developed to diagnose and identify plant diseases from the leaf images of healthy and diseased plants (Ferentinos, 2018). Rice diseases identification method was proposed by Lu et al. (2017) based on deep CNN (DCNN) techniques to identify ten common rice diseases, which increases both the convergence speed and recognition accuracy. Later, transfer learning was introduced to fine-tune the pre-trained deep networks to improve learning efficiency. Recently, Too et al., reported the analysis of state-of-the-art deep learning models for plant disease identification (Too et al., 2018). Fine-tuning is a concept of transfer learning which need a bit of learning, it is proved that much faster and more accuracy than built models (Mohanty et al., 2016). Ghazi et al., tested GoogLeNet, AlexNet, and VGGNet models using transfer learning to improve the plant species identification accuracy (Ghazi et al., 2017). Deep pre-trained models were implemented by Khan et al., to extract deep features for classifying six types of apple and banana fruit diseases with improved precision and accuracy of classification (Khan et al., 2018). In Liu et al. (2016), 8-layer CNN network was developed to learn powerful local features from the complex insect image dataset and achieved a high mean Accuracy Precision for classification of 12 important paddy insect species. Wang et al. (2017) applied LeNet-5 and AlexNet to classify crop pest images by analysing the effects of both the convolution kernel and the number of layers on the network. However, he reported only two pre-trained models to classify crop pest images. The advanced pre-trained and CNN models are needed to classify crop pest images for better accuracy.

In the present work, CNN model is proposed to provide high accuracy in insect classification task and pre-trained CNN models using transfer learning are applied for comparison of classification accuracy. Three insect datasets were selected, which are collected from different field crops. The first insect dataset is collected from NBAIR that contains 40 classes of insects from various field crops such as rice, maize, soybean, sugarcane and cotton crops (http://www.nbair.res.in/insectpests/pestsearch.php). The second (Xie1) and third insect dataset (Xie2) were adopted from (Xie et al., 2015) with 24 classes of insect pests and (Xie et al., 2018) with 40 classes of insects respectively. We explore and evaluate different DCNN models of AlexNet, ResNet, GoogLeNet and VGG with transfer learning and achieving significantly better classification performance. This work was implemented in *MATLAB 2018a* and utilized GPU parallelization for fast computation. Our main contributions in this paper are introducing an effective CNN model with improved performance for insect classification tasks than pre-trained models using transfer learning and investigating the effect of learning rate, the number of epochs and mini-batch size, which are important hyper parameters to achieve less classification error and avoid the model over-fitting. This proposed work used to recognize the different classes of insects in crop fields at early stage to improve the crop quality and increase the crop productivity.

2. Materials and methods

2.1. Insect dataset collection

In the experiment, the first insect dataset is collected from NBAIR that contains 40 pest types from field crops. The 24 insect classes of Xie1 and 40 insect classes of Xie2 are used as second and third insect dataset. The performance of the insect classification task is improved by applying the image pre-processing techniques to extract the insect from the original input image automatically before input in to the deep learning models. First, RGB insect image is converted in to gray scale image. Canny Edge Detection is applied to detect the edges in an insect image and to supress the noise. The extreme outer contours in an edge detected binary insect image are found. Bounding box is determined by four points (x,y,w,h) where (x,y) is taken as top-left coordinate of the bounding rectangle and (w,h) is its width and height. The up-right bounding rectangle for each contour is calculated likewise. If the bounding rectangle that contains the insect with width and height greater than 100 pixels, then the region of interest (ROI) of the insect is extracted according to the co-ordinates (x + w, y + h) from the original RGB insect image. Finally, the cropped insect image is stored as output. Fig. 1a shows the pre-processed cropped insect image from NBAIR insect dataset. Then, all the cropped insect images are resized to 227 × 227. The geometric transformation techniques such as scaling, transposing, rotation and flipping were applied to expand the number of insect samples in the datasets (Lopez et al., 2017). This type of image augmentation technique is relatively generic and computationally low cost to train the deep learning models effectively. The sample insects for 40 classes of NBAIR dataset (Fig. 1b) and the details of insect species are listed in Table 1. The class labels and insect names of Xie1 and Xie2 insect datasets are given in supporting information of S1 and S2.

2.2. Deep convolutional neural networks

Deep CNN is currently one of the most popular models and has exhibited their great performance on many image classification problems in agriculture field (Kamilaris and Prenafeta-Boldu, 2018). The concept of sharing weights in DCNN makes an effective image classification by discovering robust features in the images and reduce the vanishing gradient problem. The architecture of a typical CNN is given in Fig. 2. The structure of CNN includes convolution layer, pooling

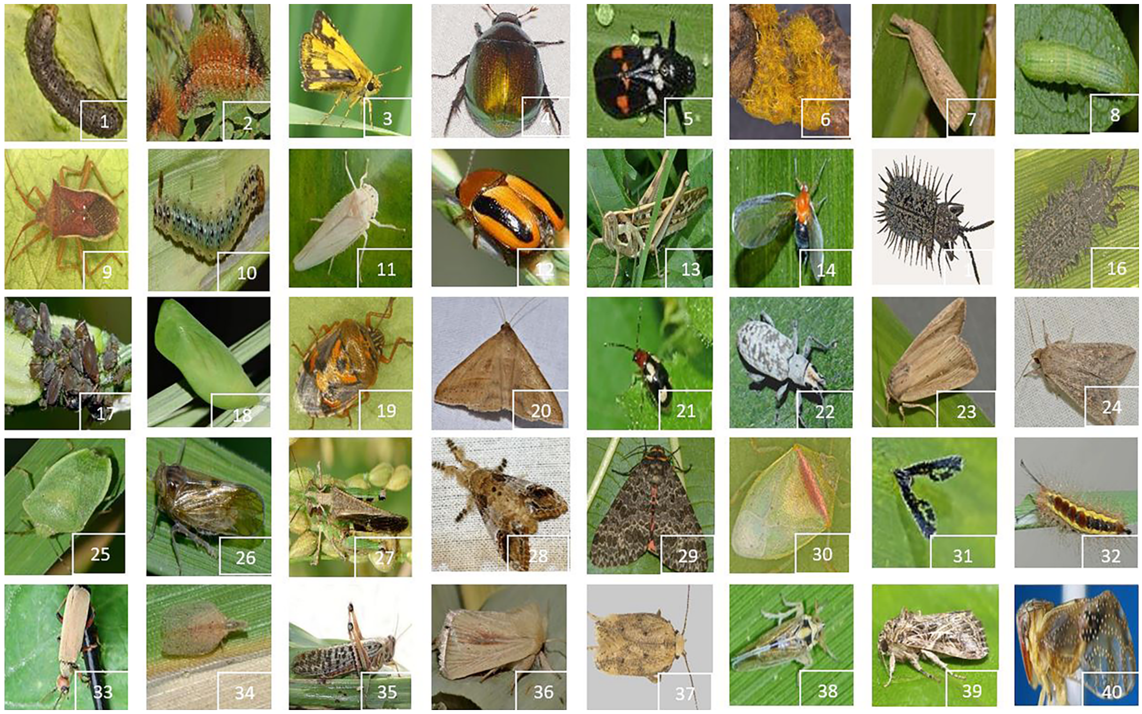


Fig. 1b. Examples of insect classes in NBAIR insect dataset.

layer, and fully connected layer. The convolutional layer acts as filters and the main task is to extract features from the insect images. The convolutional layer is followed by pooling layer, which performs down sampling and retains the most important information in the insect images. This layer reduces the spatial size of representation as well as the number of parameters and prevents overfitting which makes the model more efficient. The last layer is the fully connected layers that use a softmax activation function and takes the high-level features from the insect images for classifying them into various categories with labels.

2.3. Proposed CNN model for insect classification

We proposed an effective deep CNN model for the classification of field crop insects. It consists of 6 convolutional layers, 5 max pooling layers, one fully connected layer and the output layer with Softmax classifier. Fig. 3 presents the framework of our proposed model. Three different insect datasets with an image size of 227 × 227 was used in this experiment. The first convolution layer (Convolution 1) uses 8 initial convolution filters with a kernel size of 5 × 5 and rectified linear unit (ReLU) activation function is applied to activate neurons of the next layers to make an effective insect classification model. A 2 × 2 sized Max pooling layer is applied after every convolution layer from Convolution 1 to Convolution 5 with a stride of 2 and zero padding. The second convolution layer (Convolution 2) applies 16 convolution filters with 5 × 5 kernel size. Further, the number of convolutional kernels are increased from Convolution 3 to Convolution 6 as 32, 64, 128 and 256 with 3 × 3 kernel size (Fig. 3). We applied batch normalization between convolutional layers and ReLU layer to increase the speed of learning and the overall classification accuracy. The last fully connected layer has 'C' neurons, where 'C' represents the number of classes of insects and the classification results are fed as input to the Softmax

Table 1
NBAIR insect samples.

Insect label	Name of the insect	Insect label	Name of the insect
1	Agrotis sp	21	Monolepta signata Olivier
2	Aloa albistriga Walker	22	Myllocerus undecimpustulatus Faust
3	Ampittia dioscorides (Fabricius)	23	Mythimna loreyi (Duponchel)
4	Anomala dimidiata Hope	24	Mythimna separata (Walker)
5	Callitettix versicolor (Fabricius)	25	Nezara viridula (Linnaeus)
6	Cerococcus indicus (Maskell)	26	Nilaparvata lugens (Stal)
7	Chilo partellus (Swinhoe)	27	Oedaleus senegalensis (Krauss)
8	Chrysodeixis chalcites (Esper)	28	Olene mendosa Hubner
9	Cletus punctiger (Dallas)	29	Olepa ricini (Fabricius)
10	Cnaphalocrocis trapezalis (Guene)	30	Piezodorus hybneri (Gmelin)
11	Cofana spectra (Distant)	31	Proutista moesta (Westwood)
12	Cryptocephalus schestedti Fabricius	32	Psalis pennatula (Fabricius)
13	Cyrtacanthacris tatarica (L.)	33	Psalydolytta rouxi (Castelnau)
14	Diostrombus carnosa (Westwood)	34	Pyrilla perpusilla Walker
15	Hispa ramosa group	35	Schistocerca gregaria (Forskal)
16	Hispa stygia (Chapuis)	36	Sesamia inferens (Walker)
17	Hysteroneura setariae (Thomas)	37	Sitotroga cerealella (Olivier)
18	Melanitis leda Linnaeus	38	Sogatella furcifera (Horvath)
19	Menida versicolor (Gmelin)	39	Spodoptera litura (Fabricius)
20	Mocis frugalis (Fabricius)	40	Tropidocephala serendiba (Melichar)

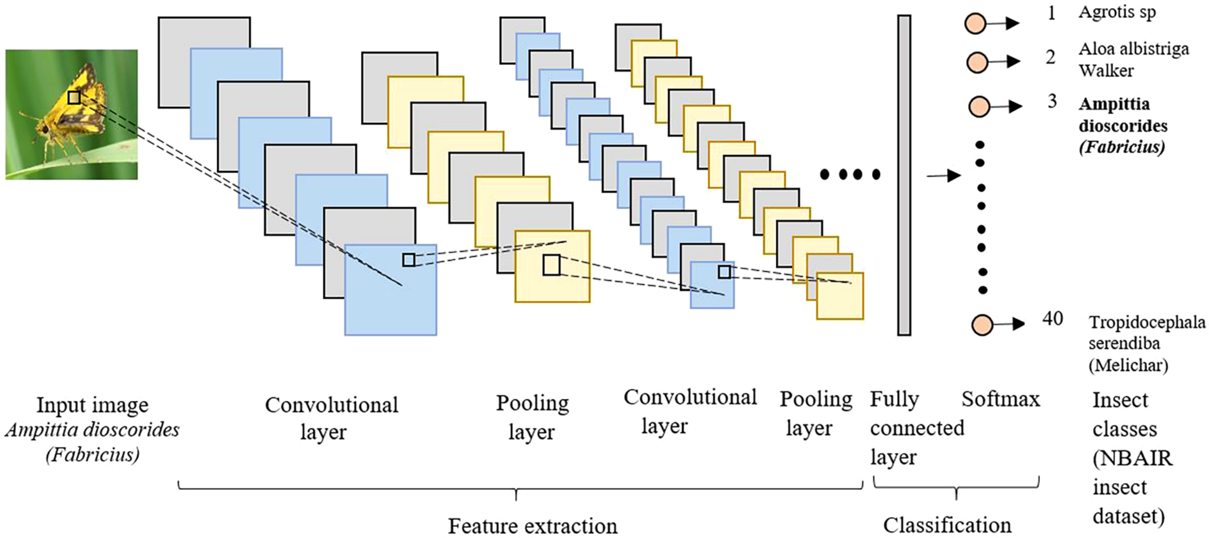


Fig. 2. Basic CNN model architecture.

classifier. In this model, categorical cross entropy is used as a loss function and Stochastic gradient descent (SGD) is used as the optimizer. Table 2 presents the details of each layer of our CNN model. Further, the results of our proposed model are compared with mostly used pre-trained models such as AlexNet, ResNet, GoogLeNet and VGG using transfer learning approach.

2.4. Transfer learning of DCNNs

In this study, transfer learning approach was applied to retrain deep learning models and the insect classification tasks are evaluated in terms of accuracy and efficiency. Deep learning models contain layered architecture with different layers to learn complex features of the insect images and finally, all these layers are connected to a fully connected layer to get the final results. In transfer learning, this layered architecture is allowed to use the pre-trained models such as AlexNet, ResNet and VGG without its final classification layer as fixed feature extractor to achieve better insect classification performance with less training time. We mainly explore four deep learning models based on CNN such as AlexNet, ResNet, GoogLeNet and VGGNet for classification of field crop insects via transfer learning (Leonardo et al., 2018). The contribution of this work is an effective learning methodology, which is used to tackle the insect classification problem. The deep learning models that are adopted to train on different insect datasets using transfer learning are explained detail in the next subsections.

2.4.1. AlexNet

The original AlexNet consists of 5 convolutional layers, 3 max pooling layers and 3 fully connected layers (Krizhevsky et al., 2012). The image input layer requires the insect image of size 227 × 227 × 3.

Table 2
Proposed CNN model parameters and Output.

Layers	Output size	Kernel size	Stride	Pad
Input image	(3,227,227)			
Convolution 1	(8,223,223)	5	1	0
Max pooling 1	(8,111,111)	2	2	0
Convolution 2	(16,107,107)	5	1	0
Max pooling 2	(16,52,52)	2	2	0
Convolution 3	(32,50,50)	3	1	0
Max pooling 3	(32,25,25)	2	2	0
Convolution 4	(64,23,23)	3	1	0
Max pooling 4	(64,11,11)	2	2	0
Convolution 5	(128,9,9)	3	1	0
Max pooling 5	(128,4,4)	2	2	0
Convolution 6	(256,2,2)	3	1	0
Fully connected	(*C)			

*C – Number of classes of insects.

ReLU (Rectified Linear Unit) is applied after every convolution and the fully connected layer which increases the non-linear properties of the network model. Cross channel normalization is applied before the first and second max pooling layer. A dropout ratio of 0.5 is applied to the first and second fully connected layers that contain 4096 neurons each. Dropout is used to reduce training time and control overfitting due to a large number of parameters. The final fully connected layer (C) has 1000 neurons which are configured to classify 1000 classes in ImageNet dataset which is fed to softmax function. For our insect classification task, the pre-trained AlexNet model is adapted by fine-tuning the last three layers. These last three layers of the model are replaced with fully connected layer, softmax layer, and output classification layer. The new fully connected layer is now set to 40, 24 and 40 classes of insects for

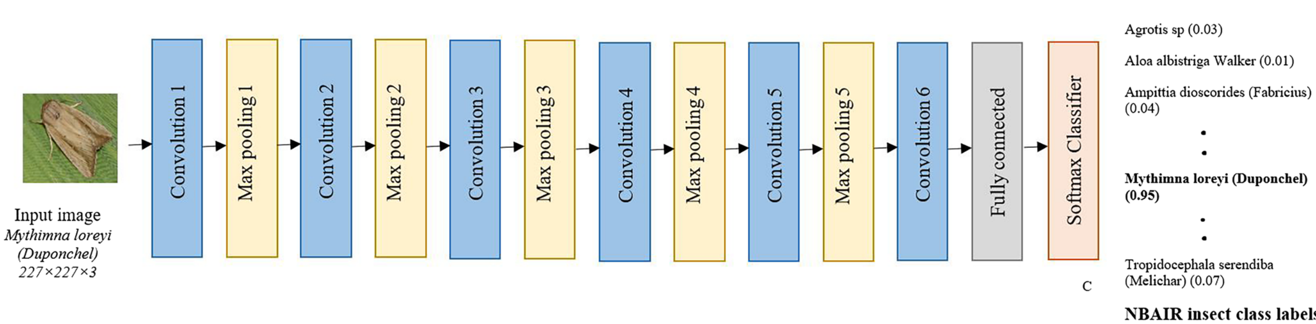


Fig. 3. Proposed deep CNN model architecture.

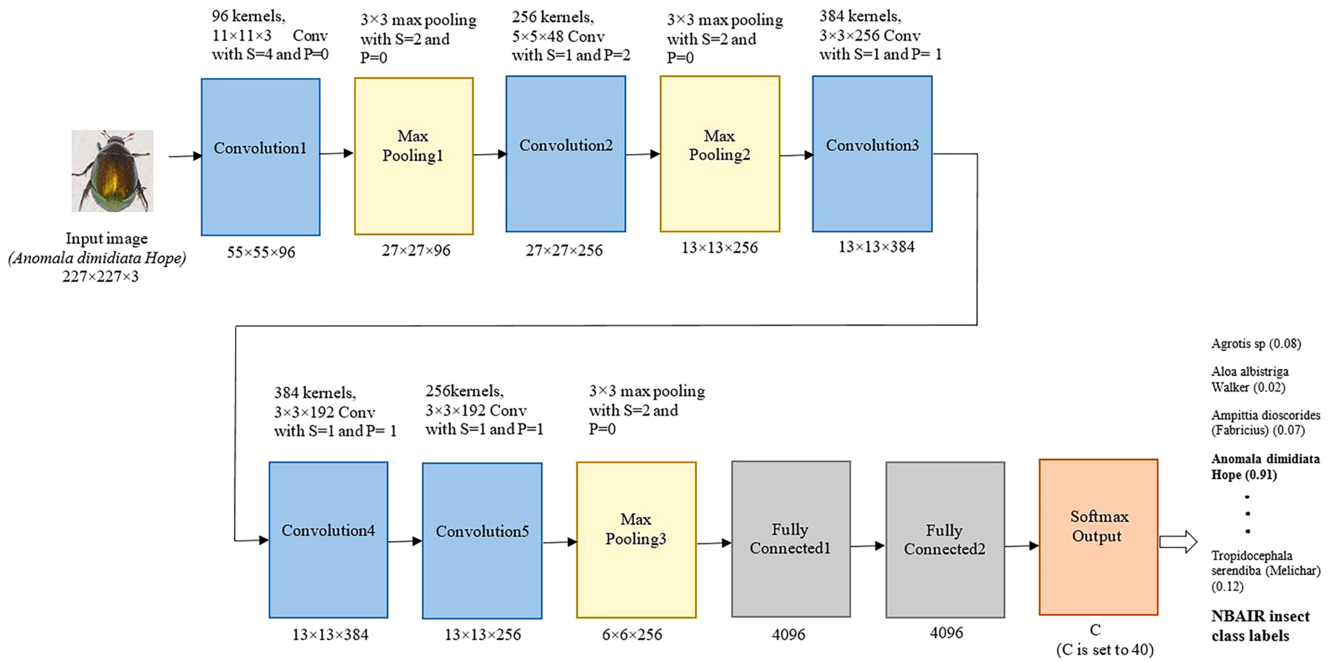


Fig. 4. Modified AlexNet Network architecture using transfer learning. C in the softmax output represents the number of classes in each dataset. (C is set to 40, 24 and 40 for NBAIR, Xie1, and Xie2 dataset, respectively; Conv – Convolution, S – Stride and P – Padding).

NBAIR, Xie1, Xie2 dataset respectively. In a fully connected layer, both weight learn rate factor and bias learn rate factor are increased to speed up the learning in the new final layers. The network is trained using the insect data by setting training parameters such as learning rate, maximum epochs, mini-batch size and validation data. The classification accuracy is the percentage of insect images that the network predicts correctly which is calculated on the testing set. The transfer learning architecture with modified AlexNet is shown in Fig. 4.

2.4.2. ResNet

Deep residual network also called ResNet, which has good performance with very deep architectures and creates a more direct path for propagating information throughout the network. The residual networks won the first ranking in the 2015 ILSVRC (ImageNet Large Scale Visual Recognition Challenge) image classification task on ImageNet test set and 2015 COCO (Common Objects in Context) competition for COCO detection and COCO segmentation (He et al., 2016). In deep networks, degradation problem occurs due to the increase of network layers and the accuracy starts to saturate which degrades rapidly. In ResNet, backpropagation does not encounter the vanishing gradient problem. A residual neural network has shortcut connections (or skip connections) which are parallel to the normal convolutional layers that help the network to understand global features. The shortcut connection is added to add the input x to the output after a few weight layers (Fig. 5). Those shortcut connections allow the network by skipping the layers that are not useful while training and results in optimal tuning of the number of layers for faster training. Mathematically, the output $H(x)$ can be defined by,

$$H(x) = F(x) + x \tag{1}$$

The weight layers are actually to learn a kind of residual mapping which is given by,

$$F(x) = H(x) - x \tag{2}$$

and $F(x)$ represents the stacked non-linear weight layers.

In this work, we evaluate the two residual network of ResNet-50 and ResNet-101 for the classification of crop field insects. ResNet-50 model is a 50 layer deep state of the art convolutional network and ResNet-101 model which consists of 101 parametrized layers with recurrent

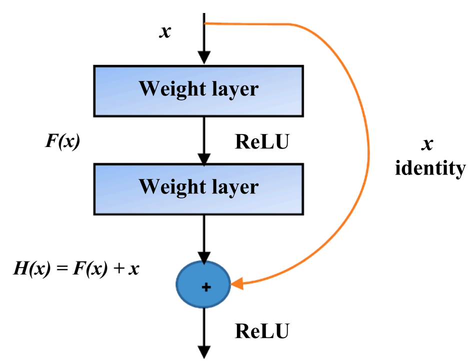


Fig. 5. Single Residual block.

connections using transfer learning are shown in Fig. 6. The ResNet-50 contains a 7 × 7 convolution layer with 64 kernels, a 3 × 3 max pooling layer with stride 2, 16 residual building blocks, a 7 × 7 average pooling layer with stride 7 and a new fully connected layer before the softmax output layer. The softmax output layer is set to 40, which indicates the number of insect classes in NBAIR dataset. The layer structure of ResNet-101 is similar to that of ResNet-50 except it has a total of 33 residual blocks. The residual blocks reduce the output size and increase the depth of the network.

2.4.3. GoogLeNet

GoogLeNet (Szegedy et al., 2015) is a deep CNN model that achieved good classification results with improved computational efficiency in several applications using transfer learning (Ghazi et al., 2017; Suh et al., 2018). GoogLeNet otherwise called as Inception model has won in ILSVRC 2014 competition and achieved a top-5 error rate of 6.67%. The GoogLeNet architecture consists of 22 layers in deep that contains 2 convolution layers, 4 max pooling layers, 9 inception modules which are linearly stacked and one average pooling which is applied at the end of last inception module (Fig. 7a). A 1 × 1 convolution is used in each inception module that performs dimension reduction before expensive large 3 × 3 and 5 × 5 convolutions (Fig. 7b). In GoogLeNet, the number of parameters is reduced due to efficient

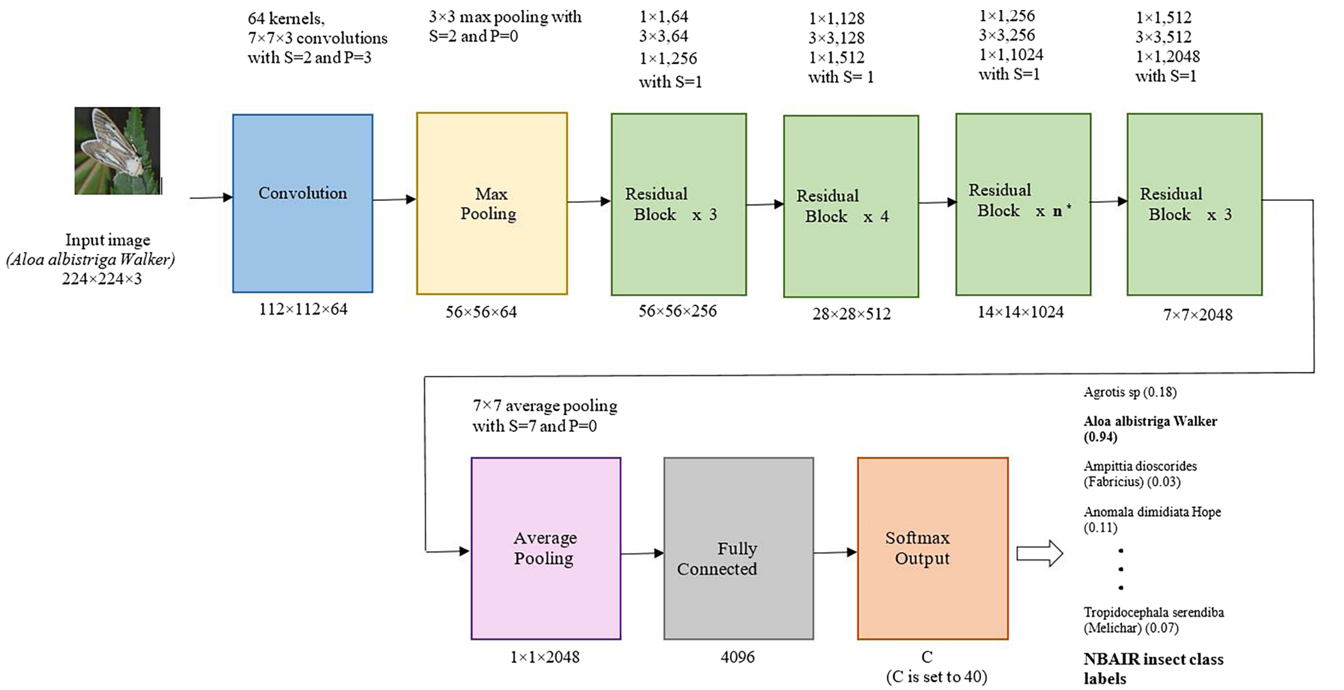


Fig. 6. Transfer learning architecture with modified ResNet-50 and ResNet-101 model ($n^* = 6$ for ResNet-50 and $n = 23$ for ResNet-101; S – Stride and P – Padding).

inception model by very small convolution and the computational cost is less than two times of AlexNet.

2.4.4. VGG network

In VGG (Visual Geometry Group) net (Simonyan and Zisserman, 2014), the depth of architecture is increased to 16 and 19 layers and number of parameters are reduced by using very small (3 × 3) convolution filters, which was named as VGG-16 and VGG-19. VGG architecture has secured first and second places on the image classification and localization tasks at ImageNet Challenge in 2014 respectively. VGG model consists of convolution layers with many consecutive 3 × 3 convolutions and 2 × 2 max pooling layer are then followed by two fully connected layers with the final layer as the Softmax output. In this work, we used both VGG-16 and VGG-19 to perform insect classification task. The structure of the modified VGG-16 model using transfer learning is shown in Fig. 8. VGG-19 model is same as VGG-16 model except that it contains four 3 × 3 convolution layers in each Convolution 3, 4 and 5 blocks of layers which is indicated by red dotted blocks (Fig. 8).

3. Results and discussion

In this work, the experiments were conducted on three different insect datasets: NBAIR, Xie1, and Xie2. Each insect dataset is divided

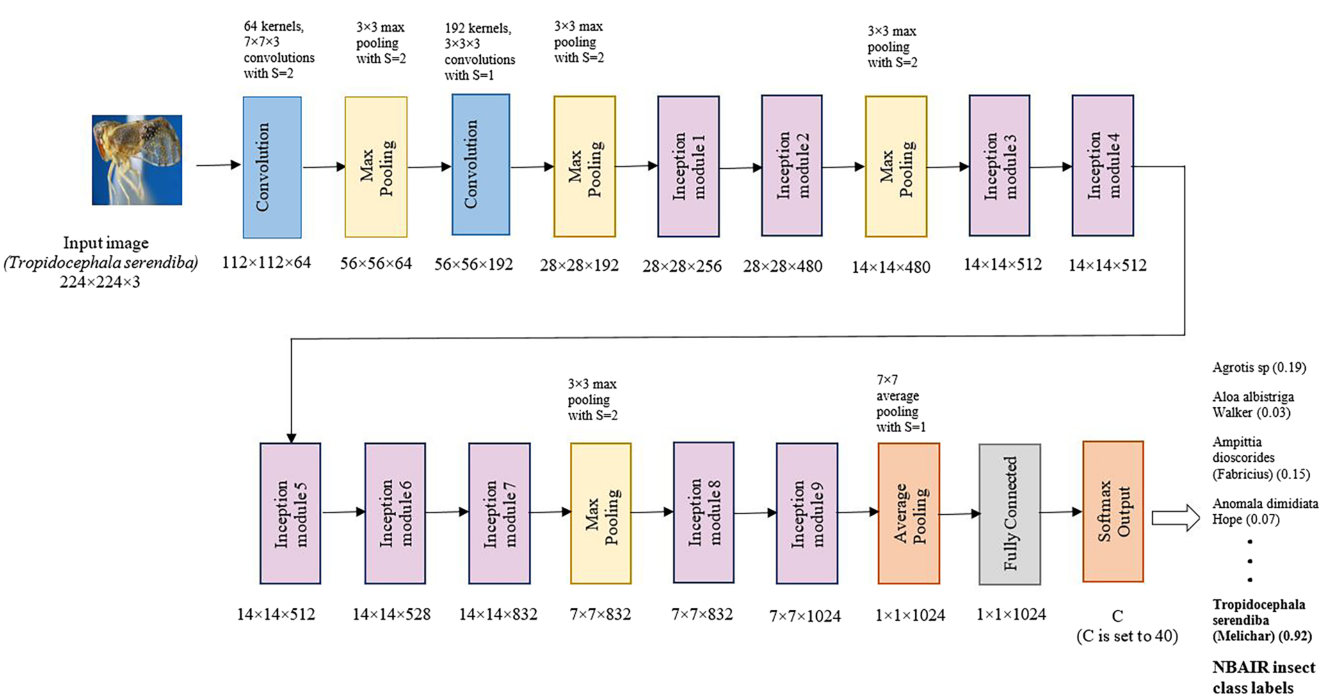


Fig. 7a. GoogLeNet Architecture using transfer learning. (S-Stride).

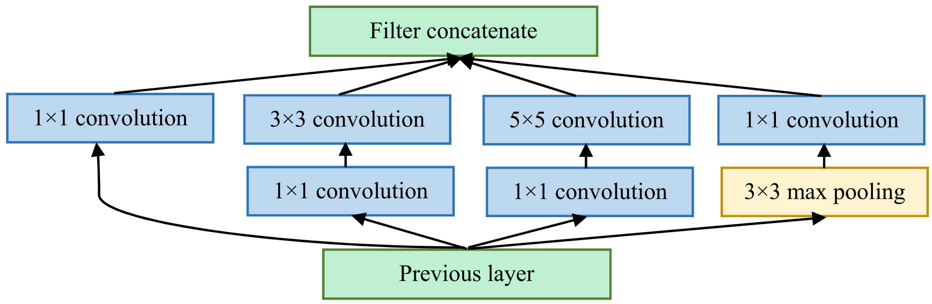


Fig. 7b. Inception module.

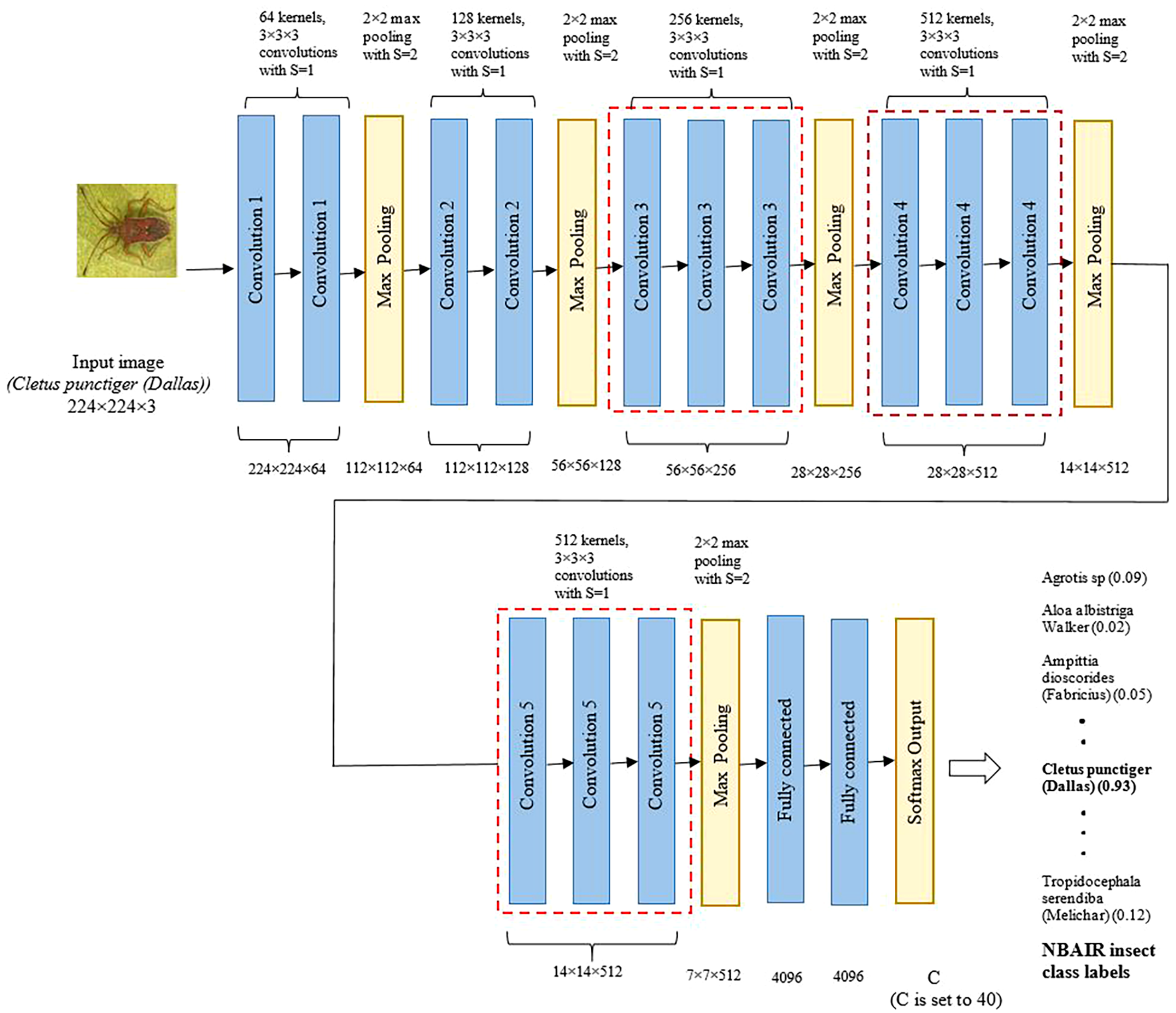


Fig. 8. Modified VGG-16 model configuration. (S-Stride).

into two categories i.e., training and testing data sets, such that 70% of insect images in each class are used as training dataset and the remaining 30% of insect images are used as a testing set (Liu and Cocea, 2017). The framework that is used to implement deep learning models is Matlab2018a using GPU NVIDIA Quadro K2200 with 4 GB of VRAM.

3.1. Proposed CNN model parameters

The proposed CNN model is trained using the SGD as optimizer with 0.9 momentum and learning rate is varied as 0.00005, 0.0001, 0.0005 and 0.001. The learning rate defines the learning progress of the proposed model and updates the weight parameters to reduce the loss function of the network. The maximum number of epochs is varied up to 10 and a mini-batch size of 10, 16, 32, 64 and 128 are used in this experiment. An epoch is the full training cycle over the entire training insect dataset and subset of training insect dataset is called as mini-batch for evaluating gradient descent loss function and also updating the weights. The model is trained on the insect training dataset and the accuracy is calculated on the testing dataset in a regular interval with validation frequency of 30. Classification accuracy is an important evaluation metric used in our model and it is given by,

$$Accuracy = \frac{Number\ \ of\ \ insects\ \ predict\ \ correctly}{Total\ \ number\ \ of\ \ input\ \ insect\ \ samples} \quad (3)$$

Categorical cross entropy is used as the loss function, which have softmax activations in the output layer and is given by,

$$Loss = \sum_{i=1}^{N} \sum_{j=1}^{K} t_{ij} \ln y_{ij} \quad (4)$$

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

3.2. Effects of convolution and pooling layers in the proposed CNN model

In this work, CNN model is constructed by selecting an appropriate number of convolution and pooling layers to improve the classification accuracy. If the model contains less number of convolution layers, the essential features of the insect image cannot be learned by the model and adding more number of layers that extracts more features result in overfitting problem. The following five combinations of convolution and pooling layers are studied in this work with the learning rate of 0.0001, number of epochs of 10 and mini-batch size of 64: *(conv-7, pool-6), (conv-6, pool-5)(conv-5, pool-4),(conv-4, pool-3) and (conv-3, pool-2)*. The accuracy is increased on adding convolution and pooling layers from 3 to 6 and 2 to 5 respectively. For the combination of 7 convolution and 6 pooling layers, the accuracy gets reduced. Therefore, a further increase of convolution and pooling layers does not cause a significant effect on classification performance. Cheng et al. (2017b) investigated the layer depth of CNN model and varied from 9 to 11 layers for the 10 classes of Xie1 insect dataset. The accuracy is degraded with an increase in the number of layers and less error is observed for a layer depth of 10 to avoid the occurrence of degradation. From Fig. 9, it is observed that the effectiveness of the proposed method is verified with good classification results for 6 convolution and 5 pooling layers and effectively reduce overfitting for NBAIR, Xie1 and Xie2 insect datasets.

3.3. Effects of hyperparameters of the proposed model

The following section analyses the important hyperparameters such as learning rate, number of epochs and mini batch size.

3.3.1. Effects of learning rate

Learning rate is an important factor that affects the efficiency of the CNN model. The higher learning rate speed up the learning process that results in an increase of loss function and low learning rate makes the loss function to decrease slowly. It is necessary to select the optimal learning rate to minimize the loss function for insect classification problem. We trained our proposed model with a learning rate of 0.00005, 0.0001, 0.0005 and 0.001. After 10 epochs with a mini-batch size of 64, the accuracy rate for different learning rates is shown in Fig. 10. Based on the results, better classification accuracy results were achieved when the learning rate was set to 0.0001. Similarly, Xia et al. varied the learning rate from 0.0006 to 0.0014 and the highest insect classification and accuracy is obtained for the learning rate of 0.001(Xia et al., 2018). In our model, the lower learning rate gradually decreases the error and avoid the over-fitting problem.

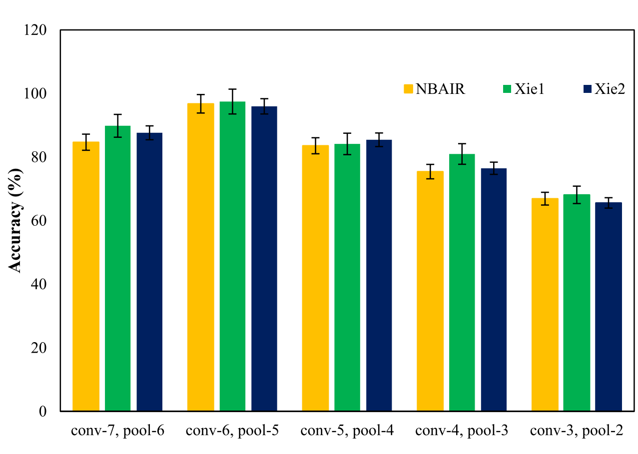


Fig. 9. Classification accuracy for different convolution and pooling layers.

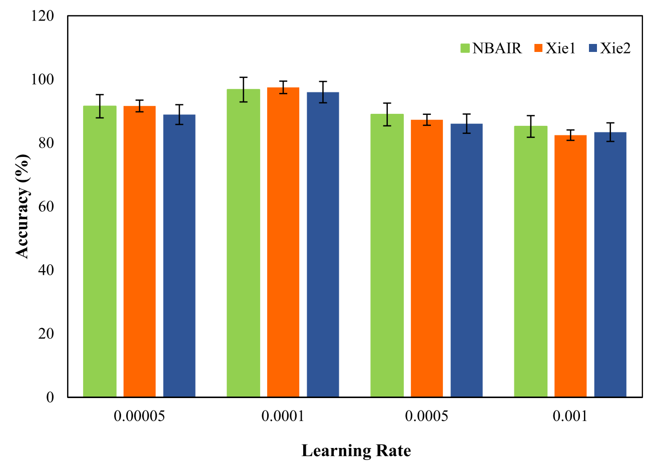


Fig. 10. Variation of accuracy with Learning Rate.

3.3.2. Effects of epochs

The optimum value of the number of epochs was determined with respect to classification accuracy. The proposed CNN model is trained up to 10 epochs for three insect datasets such as NBAIR, Xie1, and Xie2.

The learning rate and mini-batch size are set to 0.0001 and 64 respectively. As it can be seen from Fig. 11, the classification accuracy increases rapidly in the earlier epochs from 1 to 2 and then slowly increase from epoch 4 to 10. It is cleared that the increase in the number of epochs improves accuracy performance. After 10th epoch onwards, the accuracy remains constant for a further increase in the number of epochs. This is the case for all the three insect datasets. Therefore, the optimum value of the number of epochs was set to 10. The highest classification accuracy of 97.47% for Xie1 insect dataset is observed when compared to NBAIR and Xie2 insect dataset. This observed higher

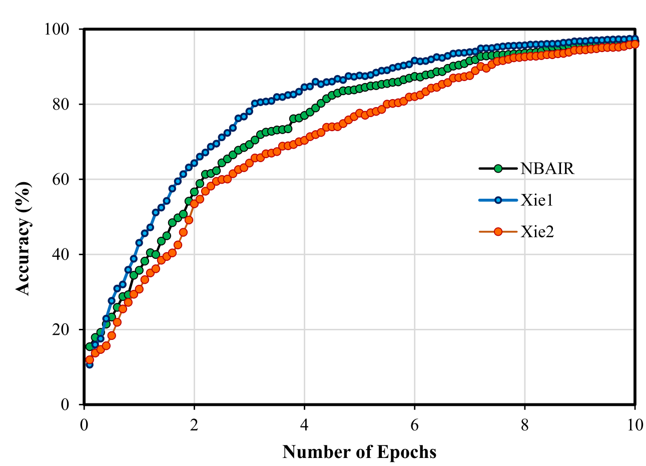


Fig. 11. Variation of accuracy with Number of Epochs.

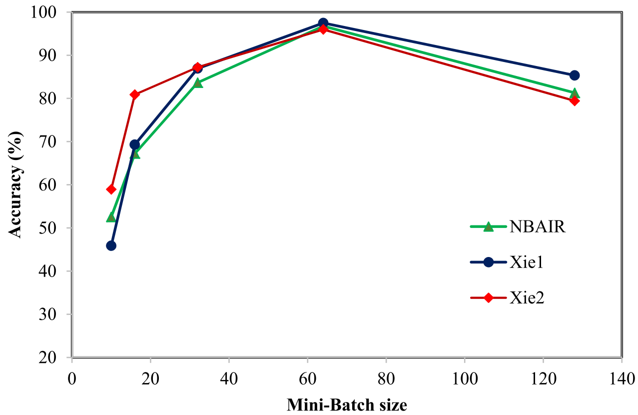


Fig. 12. Variation of accuracy with Mini-Batch size.

accuracy is due to the deep layer structure of the proposed model, which provide better insect classification for various field crops with different insect datasets.

3.3.3. Effects of mini-batch size

The size of mini-batch is an important parameter that influences the classification accuracy of the model. The larger batch size makes the model to run for a long period of time with constant weights that decreases the overall performance and affects the requirement of memory.

Therefore, the appropriate size of mini-batch is selected to improve the quality of models (Lee and Xing, 2018). The proposed model is evaluated with a mini-batch size of 10, 16, 32, 64 and 128. The accuracy of the model is improved with the increase in mini-batch size from 10 to 64 and then it decreased for 128. The model performance for three insect datasets with different sizes of mini-batch was compared in Fig. 12. The model was run for 10 epochs with a learning rate of 0.0001. Based on the experimental results, a mini-batch size of 64 is selected to train the model that increases the convergence precision. It is also clear that further increase in mini-batch size does not make improvement in accuracy. The selected mini-batch size of 64 supported the proposed model to achieve better final accuracy. Hence, the proposed model for insect classification is now more accurately identifiable for insects with a complex background.

3.3.4. Overall classification performance of the proposed model and pre-trained models with transfer learning

Pre-trained models are fine-tuned with transfer learning to learn the features quickly to perform classification on different insect datasets. In pre-trained models, the fully connected layer multiplies the input by a weight matrix and then adds a bias vector. Hence, both weight learn rate factor and bias learn rate factor values of the fully connected layer are increased to 20 to speed up the learning in the new final layers. All the pre-trained models using transfer learning were evaluated for the same training options, which were used in the proposed CNN model. The following combination of best hyper parameter values are used in this experiment that discussed in Section 3.3: Learning rate − 0.0001, Number of Epochs − 10 and mini-batch size − 64. Fig. 13 shows the comparative performance for both proposed CNN model and pre-trained models on the three insect datasets. Among the pre-trained models, it can be seen that ResNet-101 provides high performance for NBAIR (95.02%) and Xie2 (93.99%) insect dataset where GoogLeNet performed well for Xie1 (96.25%) insect dataset. For all the datasets, the performance of VGG-16 and VGG-19 are highly similar and superior to AlexNet. Further, the improvement in accuracy is observed in our proposed CNN model with 6 convolution layers and 5 pooling layers than all the pre-trained model results. From Fig. 13, it is cleared that the proposed model has achieved a higher accuracy of 96.75, 97.47, and 95.97% for NBAIR, Xie1 and Xie2 insect datasets respectively. This is due to the number of convolution and pooling layers and hyperparameters applied in the proposed CNN model. Batch normalization layer is used between every convolutional layers and ReLU layers in our CNN model that maximize the training and reduce overfitting. The error loss is minimized by training our model with SGD and shuffling of training data and validation data is performed for every epoch. The overall classification accuracies were improved up to 0.8% in this work, only when the candidate regions of insects were extracted from the image before applied into our deep learning models. The more detailed features from the ROI of insects make the model to increase the accuracy and also reduce the amount of processing time for training.

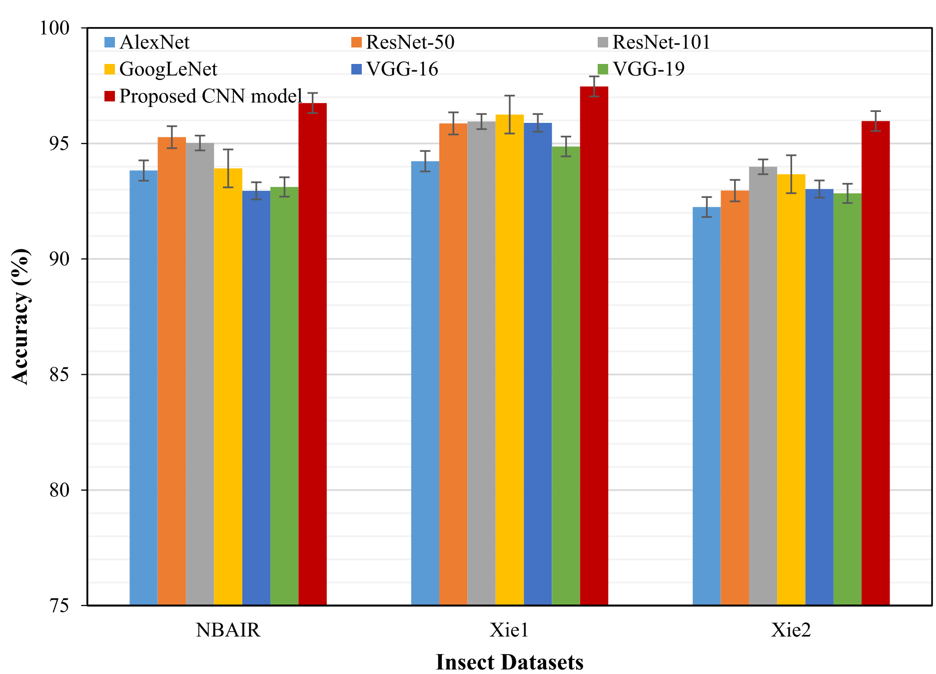


Fig. 13. Overall classification accuracies (%).

Table 3
Related work and accuracy results (%) summary.

Data set	Xie et al. (2015) (24 classes)	Cheng et al. (2017a) (10 classes)		Cheng et al. (2017b) (10 classes)		Xia et al. (2018) (24 classes)	Xie et al. (2018) (40 classes)	Proposed methods					
								Proposed CNN model*	Pre-trained models with transfer learning				
	MKL	AlexNet	VGG-16	ResNet-50	ResNet-101	VGG-19 + RPN	MKB		AlexNet	ResNet-50	ResNet-101	VGG-16	VGG-19
Xie1 (24 classes)	91.2	86.67	95.33	94.67	98.67	89.22	–	**97.47**	94.23	95.87	95.95	96.25	95.89
Xie2 (40 classes)	–	–	–	–	–	–	89.3	**95.97**	92.25	92.96	93.99	93.67	93.03

* Accuracy values of proposed CNN model.

3.4. Comparison results summary

The performance comparison of the proposed method with other existing methods for Xie1 and Xie2 insect datasets is shown in Table 3. Xie et al. (2015) applied multiple-task sparse representation with multiple-kernel learning (MKL) using multiple features sets and achieved a classification accuracy of 91.2% for 24 classes of insect pests. It requires a feature description scheme and takes longer training time to investigate the insect recognition problem. In the proposed model, an accuracy of 97.47% was achieved for the same Xie1 insect dataset with deep CNN architecture which learns the deep features of the insect images and gives better results than handcrafted features. In Cheng et al. (2017a,b), the authors applied AlexNet, VGG-16, ResNet-50 and ResNet-101 models to perform pest identification for only 10 classes of Xie1 insect dataset. Xia et al. (2018) adopted the pre-trained VGG19 model with Region Proposal Network(RPN) for Xie1 insect dataset and requires detailed information about the classification of insects. The automatic field crop pests classification is proposed by Xie et al. (2018) use multi-level learning features with multiple-level fusion classification model for 40 classes of insects. In their method, more processing time is taken and the performance needs to be improved for insects with complex background. From the experimental results, our proposed CNN model and pre-trained models with transfer learning which is used in our method provide good accuracy results for both Xie1 and Xie2 with all 24 and 40 classes of insects respectively.

4. Conclusion

The quality and quantity of field crops are affected by pest attacks. In this study, we proposed a CNN model and focussed the development of pre-trained models for field crop insect classification. Transfer learning approach is utilized to use the pre-trained models such as AlexNet, ResNet-50, ResNet-101, VGG-16, and VGG-19 for our insect classification problem and the performances were compared with the proposed model. The effect of hyper parameters is also investigated in this work. The experimental results show that our proposed CNN model with good architecture can classify different field crop insects robustly and achieved good performance than pre-trained models. The classification accuracy of 96.75, 97.47, and 95.97% was attained in proposed CNN model for insect dataset of NBAIR, Xie1 and Xie2 respectively. In future, the investigated CNN model will focus on more number of insect classes and categorization of sub-classes of insects and to improve on the computational time. In addition, proposed work will more helpful to farmers for early identification and classification of insects in filed crops and to take fast decisions to use significant amounts of pesticides in the affected crops in order to improve the quality of the crops.

Acknowledgments

Author K. Thenmozhi acknowledge to Department of Science and Technology, India for financial support of the work under women scientist scheme B (WOS-B), grant number: DST/Disha/SoRF-PM/059/2013. The authors also gratefully acknowledge infrastructural supports from Machine Learning and Data Analytics Lab, Massively Parallel Programming Lab, Department of Computer Applications, National Institute of Technology, Tiruchirappalli, India.

Appendix A. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compag.2019.104906.

References

Cheng, X., Zhang, Y.H., Wu, Y.Z., Yue, Y., 2017a. Agricultural pests tracking and identification in video surveillance based on deep learning. In: International Conference on Intelligent Computing. Springer, Cham, pp. 58–70.

Cheng, X., Zhang, Y., Chen, Y., Wu, Y., Yue, Y., 2017b. Pest identification via deep residual learning in complex background. Comput. Electron. Agric. 141, 351–356.

Ferentinos, K.P., 2018. Deep learning models for plant disease detection and diagnosis. Comput. Electron. Agric. 145, 311–318.

Ghazi, M.M., Yanikoglu, B., Aptoula, E., 2017. Plant identification using deep neural networks via optimization of transfer learning parameters. Neurocomputing 235, 228–235.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

Kamilaris, A., Prenafeta-Boldu, F.X., 2018. Deep learning in agriculture: a survey. Comput. Electron. Agric. 147, 70–90.

Khan, M.A., Akram, T., Sharif, M., Awais, M., Javed, K., Ali, H., Saba, T., 2018. CCDF: Automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep CNN features. Comput. Electron. Agric. 155, 220–236.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105.

Lee, M., Xing, S., 2018. A Study of Tangerine Pest Recognition Using Advanced Deep Learning Methods.

Leonardo, M.M., Carvalho, T.J., Rezende, E., Zucchi, R., Faria, F.A., 2018. Deep feature-based classifiers for fruit fly identification (Diptera: Tephritidae). In: 2018 31st SIBGRAPI Conference on Graphics, Patterns, and Images (SIBGRAPI). IEEE, pp. 41–47.

Liu, H., Cocea, M., 2017. Semi-random partitioning of data into training and test sets in granular computing context. Granul. Comput. 2 (4), 357–386.

Liu, Z., Gao, J., Yang, G., Zhang, H., He, Y., 2016. Localization and classification of paddy field pests using a saliency map and deep convolutional neural network. Sci. Rep. 6, 20410.

Lopez, A.R., Giro-i-Nieto, X., Burdick, J., Marques, O., 2017. Skin lesion classification from dermoscopic images using deep learning techniques. In: 2017 13th IASTED International Conference on Biomedical Engineering (BioMed). IEEE, pp. 49–54.

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

Martineau, M., Conte, D., Raveaux, R., Arnault, I., Munier, D., Venturini, G., 2017. A survey on image-based insect classification. Pattern Recogn. 65, 273–284.

Mohanty, S.P., Hughes, D.P., Salathé, M., 2016. Using deep learning for image-based plant disease detection. Front. Plant Sci. 7 (September), 1–7.

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

Suh, H.K., IJsselmuiden, J., Hofstee, J.W., van Henten, E.J., 2018. Transfer learning for the classification of sugar beet and volunteer potato under field conditions. Biosyst. Eng. 174, 50–65.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 1–9.

Too, E.C., Yujian, L., Njuki, S., Yingchun, L., 2018. A comparative study of fine-tuning deep learning models for plant disease identification. Comput. Electron. Agric. 161, 272–279.

Wang, J., Lin, C., Ji, L., Liang, A., 2012. A new automatic identification system of insect images at the order level. Knowl.-Based Syst. 33, 102–110.

Wang, R., Zhang, J., Dong, W., Yu, J., Xie, C.J., Li, R., Chen, T., Chen, H., 2017. A crop pests image classification algorithm based on deep convolutional neural network. Telkomnika 15 (3).

Wen, C., Guyer, D.E., Li, W., 2009. Local feature-based identification and classification for orchard insects. Biosyst. Eng. 104 (3), 299–307.

Xia, D., Chen, P., Wang, B., Zhang, J., Xie, C., 2018. Insect detection and classification based on an improved convolutional neural network. Sensors 18 (12), 4169.

Xie, C., Wang, R., Zhang, J., Chen, P., Dong, W., Li, R., Chen, T., Chen, H., 2018. Multi-level learning features for automatic classification of field crop pests. Comput. Electron. Agric. 152, 233–241.

Xie, C., Zhang, J., Li, R., Li, J., Hong, P., Xia, J., Chen, P., 2015. Automatic classification for field crop insects via multiple-task sparse representation and multiple-kernel learning. Comput. Electron. Agric. 119, 123–132.