RECOGNITION OF FLY SPECIES-LOCUST BASED ON IMPROVED RESNET

Mr. j. Vetrimanikumar[1], P.M.Monica[2], S.Vinu deepthi [3], S.A.Padma bhagavathi[4], A.Loga dharshini[5]

[1]Assistant Professor, Department of Electronics and Communication Engineering, SSM Institute of Engineering and Technology, Tamilnadu , India.

[2,3,4,5] UG Scholar, Department of Electronics and Communication Engineering, SSM Institute of Engineering and Technology, Tamilnadu , India.

---***---

ABSTRACT

A dozen species of locusts (Orthoptera: Acrididae) are a major threat to food security worldwide. Their outbreaks occur on every continent except Antarctica, threatening the livelihood of 10% of the world's population. The locusts are infamous for their voracity, polyphagy, and capacity for long-distance migrations. For effective control, the insects need to be detected on the ground before they start to develop air borne swarms. Detection systems need to determine pest density and location with high speed and accuracy. Location of the swarms on the ground then enables their control by the application of pesticides and bio-pesticides. This work proposes a locust species recognition method based on Resnet50 - convolutional neural network (CNN). We experimentally compared the proposed method with other the state-of-the-art methods on the established dataset. Experimental results showed that accuracy of this method reached higher than the state-of-the-art methods. This method has a good detection effect on the fly species recognition.

***Key Words*: Orthoptera: Acrididae , CNN-Convolution neutral network, good detection.**

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.

1.1 Feature extraction

Feature extraction transforms the raw data into meaningful representations for a given classification task. Images are typically composed of millions of pixels with associated colour information each. The high dimensionality of these images is reduced by computing abstract features, i.e. a quantified representation of the image retaining relevant information for the classification problem (e.g. shape, texture or colour information) and omitting irrelevant. Traditionally, features to be extracted were designed by domain experts in a typically long term and rather subjective manual process. For instance, it was observed that humans are sensitive to edges in images. Many well-known computer vision algorithms follow this pattern and use edge or gradient based features, e.g. the scale invariant feature transform (SIFT). SIFT is a widely adopted approach for object detection and image comparison that efficiently detects and describes characteristic and scale invariant keypoints within images that provided a huge improvement over earlier approaches

1.2 Classification

Depending on the application, the score is either compared to a threshold solely deciding whether an object is present or not (e.g. presence of a plant or animal in the image), or it is compared to other scores to distinguish object classes (e.g. species name). Prominent classification methods are machine learning algorithms such as support vector machines, Random Forest and artificial neuronal network (ANN).

2. EXISTING SYSTEM

Automatic image segmentation techniques can be classified into four categories, namely, (1) Clustering Methods, (2) Thresholding Methods, (3) Edge-Detection Methods, and (4) Region-Based Methods .

1. Clustering Methods

Clustering is a process whereby a data set (pixels) is replaced by cluster; pixels may belong together because of the same color, texture etc. There are two natural algorithms for clustering: divisive clustering and agglomerative clustering. The difficulty in using either of the methods directly is that there are lots of pixels in an image. Also, the methods are not explicit about the objective function that is being optimized. An alternative approach is to write down an objective function and then build an algorithm. The K-means algorithm is an iterative technique that is used to partition an image into K clusters, where each pixel in the image is assigned to the cluster that minimizes the variance between the pixel and the cluster center and is based on pixel color, intensity, texture, and location, or a weighted combination of these factors. This algorithm is guaranteed to converge, but it may not return the optimal solution. The quality of the solution depends on the initial set of clusters and the value of K .

2. Thresholding Methods

Thresholding is the operation of converting a multilevel image into a binary image i.e., it assigns the value of 0 (background) or 1 (objects or foreground) to each pixel of an image based on a comparison with some threshold value T (intensity or color value). When T is constant, the approach is called global thresholding; otherwise, it is called local thresholding. Global thresholding methods can fail when the background illumination is uneven. Multiple thresholds are used to compensate for uneven illumination. Threshold selection is typically done interactively.

3. Edge Detection Methods

Edge detection methods locate the pixels in the image that correspond to the edges of the objects seen in the image. The result is a binary image with the detected edge pixels. Common algorithms used are Sobel, Prewitt, Robert, Canny and Laplacian operators. These algorithms are suitable for images that are simple and noise free; and will often produce missing edges, or extra edges on complex and noisy images.

4. Region-Based Methods

The goal of region-based segmentation is to use image characteristics to map individual pixels in an input image to sets of pixels called regions that might correspond to an object or a meaningful part of one. The various techniques are: Local techniques, Global techniques and Splitting and merging techniques. The effectiveness of region growing algorithms depends on the application area and the input image. If the image is sufficiently simple, simple local techniques can be effective. However, on difficult scenes, even the most sophisticated techniques may not produce a satisfactory segmentation. Edge-based techniques are based on the assumption that pixel values change rapidly at the edge between two regions Operators such as Sobel or Roberts operators can be used to detect the edges. And some post procedures such as edge tracking, gap filling can be used to generate closed curves. Regionbased techniques are based on the assumption that adjacent pixels in the same region should be consistent in some properties. Namely, they may have similar characteristic such as grey value, color value or texture. The deformable models are based on curves or surfaces defined within an image that moves due to the influence of certain forces. And the global optimization approaches use a global criterion when segmenting the image.

3. PROPOSED SYSTEM

CNN Overview

This network structure was first proposed by Fukushima in 1988 . It was not widely used, however, due to limits of computation hardware for training the network. In the 1990s,It applied a gradient-based learning algorithm to CNNs and obtained successful results for the handwritten digit classification problem. After that, researchers further improved CNNs and reported state-of-the-art results in many recognition tasks. CNNs have several advantages over

DNNs, including being more like the human visual processing system, being highly optimized in the structure for processing 2D and 3D images, and being effective at learning and extracting abstractions of 2D features. The max pooling layer of CNNs is effective in absorbing shape variations. Moreover, composed of sparse connections with tied weights, CNNs have significantly fewer parameters thana fully connected network of similar size. Most of all, CNNs are trained with the gradient-based learning algorithm and suffer less from the diminishing gradient problem. Given that the gradientbased algorithm trains the whole network to minimize an error criterion directly, CNNs can produce highly optimized weights.

Figure shows the overall architecture of CNNs consists of two main parts: Feature extractors and a classifier. In the feature extraction layers, each layer of the network receives the output from its immediate previous layer as its input and passes its output as the input to the next layer. The CNN architecture consists of a combination of three types of layers: Convolution, max-pooling, and classification. There are two types of layers in the low and middle-level of the network: Convolutionallayers and max-pooling layers. The even numbered layers are for convolutions and the oddnumbered layers are for max-pooling operations. The output nodes of the convolution and maxpooling layers are grouped into a 2D plane called feature mapping. Each plane of a layer is usually derived from the combination of one or more planes of previous layers. The nodes of a plane are connected to a small region of each connected planes of the previous layer. Each node of the convolution layer extracts the features from the input images by convolution operations on the input
nodes.

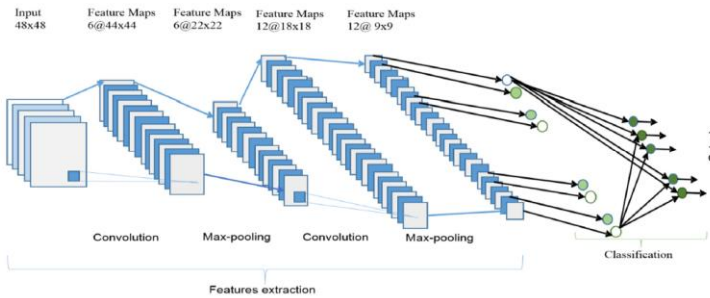

Figure The overall architecture of the Convolutional Neural Network (CNN) includes an input layer, multiple alternating convolution and max-pooling layers, one fully-connected layer and one classification layer.

Higher-level features are derived from features propagated from lower level layers. As the features propagate to the highest layer or level, the dimensions of features are reduced depending on the size of the kernel for the convolutional and max-pooling operations respectively. However, thenumber of feature maps usually increased for representing better features of the input images forensuring classification accuracy. The output of the last layer of the CNN is used as the input to a fullyconnected network which is called classification layer. Feed-forward neural networks have been usedas the classification layer as they have better performance . In the classification layer, the extracted features are taken as inputs with respect to the dimension of the weight matrix of the final neural network. However, the fully connected layers are expensive in terms of network or learning parameters. Nowadays, there are several new techniques, including average pooling and globalaverage pooling that is used as an alternative of fully-connected networks. The score of the respective class is calculated in the top classification layer using a soft-max layer. Based on the highest score, the classifier gives output for the corresponding classes. Mathematical details on different layers of CNNs are discussed in the following section.

Convolutional Layer

In this layer, feature maps from previous layers are convolved with learnable kernels. The output of the kernels goes through a linear or non-linear activation function, such as sigmoid, hyperbolic tangent, Softmax, rectified linear, and identity functions) to form the output feature maps. Each of the output feature maps can be combined with more than one input feature map. In general, we have that

$$x_j^l = f\left(\sum_{i \epsilon M_j} x_i^{l-1} * k_{ij}^l + b_j^l\right),$$

Where xjlis the output of the current layer, xil-1is the previous layer output, kijlis the kernel for the present layer, and bjlare the biases for the current layer. mjrepresents a selection of input maps. For each output map, an additive bias bis given. However, the input maps will be convolved with distinct kernels to generate the corresponding output maps. The output maps finally go through a linear or non-linear activation function (such as sigmoid, hyperbolic tangent, Softmax, rectified linear, or identity functions).

Sub-sampling Layer

The subsampling layer performs the down sampled operation on the input maps. This is commonly known as the pooling layer. In this layer, the number of input and output feature maps does not change. For example, if there are Ninput maps, then there will be exactly Noutput maps. Due to the down sampling operation, the size of each dimension of the output maps will be reduced, depending on the size of the down sampling mask. For example, if a 2 × 2 down sampling kernel is
used, then each output dimension will be half of the corresponding input dimension for all the images. This operation can be formulated as

$$\mathrm{x}_j^l = \mathrm{down}\left(\mathrm{x}_j^{l-1}\right),$$

wheredown(.) represents a sub-sampling function. Two types of operations are mostly performed in this layer: Average pooling or max-pooling. In the case of the average pooling approach, the function usually sums up over N × N patches of the feature maps from the previous layer and

selects the average value. On the other hand, in the case of max-pooling, the highest value is selected from
the N×N patches of the feature maps. Therefore, the output map dimensions are reduced by n times. In some special cases, each output map is multiplied with a scalar. Some alternative sub-sampling layers have been proposed, such as fractional max-pooling layer and sub-sampling with convolution.

Classification Layer

This is the fully connected layer which computes the score of each class from the extracted features from a convolutional layer in the preceding steps. The final layer feature maps are represented as vectors with scalar values which are passed to the fully connected layers. The fully connected feed-forward neural layers are used as a soft-max classification layer. There are no strict rules on the number of layers which are incorporated in the network model. However, in most cases,
two to four layers have been observed in different architectures, including LeNet , AlexNet , and VGG Net. As the fully connected layers are expensive in terms of computation, alternative approaches have been proposed during the last few years. These include the global average pooling layer and the average pooling layer which help to reduce the number of parameters in the network si
gnificantly.
In the backward propagation through the CNNs, the fully connected layer updates followingthe general approach of fully connected neural networks (FCNN). The filters of the convolutional layers are updated by performing the full convolutional operation on the feature maps between the convolutional layer and its immediate previous layer. Figure 10 shows the basic operations in the convolution and sub-sampling of an input image.

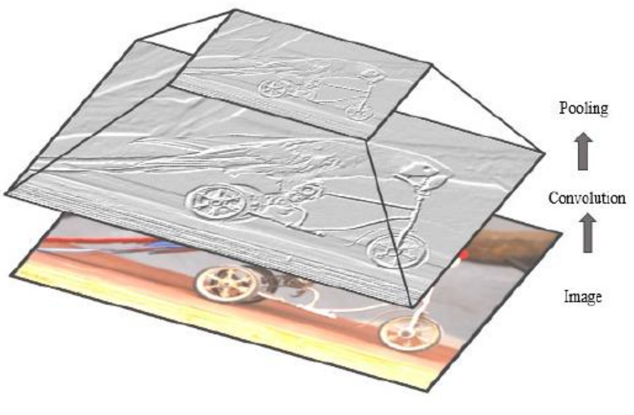

Figure .Feature maps after performing convolution and pooling operations.

Network Parameters and Required Memory for CNN

The number of computational parameters is an important metric to measure the complexity of a deep learning model. The size of the output feature maps can be formulated as follows:

$$M = \frac{(N - F)}{S} + 1,$$

whereNrefers to the dimensions of the input feature maps, Frefers to the dimensions of the filters or the receptive field, Mrefers to the dimensions of output feature maps, and stands for the stride
length. Padding is typically applied during the convolution operations to ensure the input and output feature map have the same dimensions. The amount of padding depends on the size of the kernel.
Equation 17 is used for determining the number of rows and columns for padding.

$$P = (F - 1)/2,$$

herePis the amount of padding and Frefers to the dimension of the kernels. Several criteria are considered for comparing the models. However, in most of the cases, the number of network parameters and the total amount of memory are considered. The number of parameters (Parml)of lthlayer is the calculated based on the following equation:

$$Parm_l = (F \times F \times FM_{l-1}) \times FM_l.$$

If bias is added with the weights, then the above equation can be written as follows:

$$Parm_l = (F \times (F + 1) \times FM_{l-1}) \times FM_l,$$

here the total number of parameters of lthlathe yer can be represented with Pl ,FMlis for the totalnumber of output feature maps, and FMl-1is the total number of input feature maps or channels.

4.2 Residual Network (ResNet in 2015)

The winner of ILSVRC 2015 was the Residual Network architecture, ResNet.Resnet was developed by Kaiming He with the intent of designing ultra-deep networks that did not suffer from the vanishing gradient problem that predecessors had. ResNet is developed with many different numbers of layers; 34, 50,101, 152, and even 1202. The popular ResNet50 contained 49 convolution layers and 1 fully connected layer at the end of the network. The total number of weights and MACs
for the whole network are 25.5M and 3.9M respectively. The basic block diagram of the ResNet architecture is shown in Figure 16. ResNet is a traditional feedforward network with

a residual connection. The output of a residual layer can be defined based on the outputs of (l− 1)thwhich comes from the previous layer defined as xl-1. $\mathscr{F}$(xl-1) is the output after performing various operations (e.g., convolution with different size of filters, Batch Normalization (BN) followed by an activation function, such as a ReLU on xl-1). The final output of residualthe unit is xlwhich can be defined with the following equation:xl = $\mathscr{F}$(xl-1) + xl-1. (15)

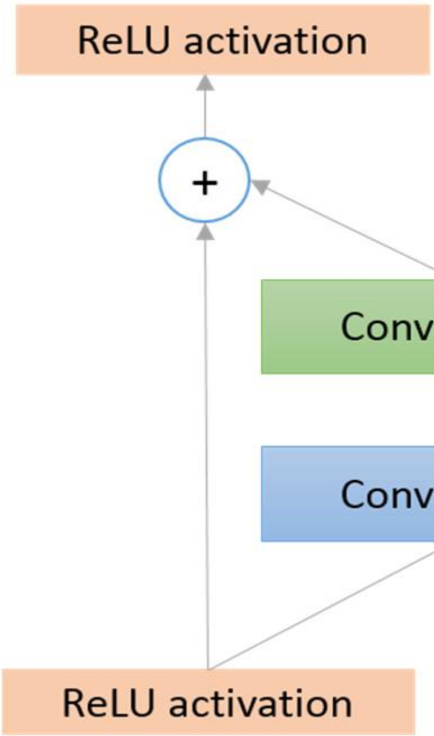

Figure Basic diagram of the Residual block.

The residual network consists of several basic residual blocks. However, the operations in the residual block can be varied depending on the different architecture of residual networks . The wider version of the residual network was proposed by Zagoruvko el at. , another improved residual network approach known as aggregated residual transformation . Recently, some other variants of residual models have been introduced based on the Residual Network architecture. Furthermore, there are several advanced architectures that are combined with Inception and
Residual units. The basic conceptual diagram of Inception-Residual unit is shown in the following Figure 17.

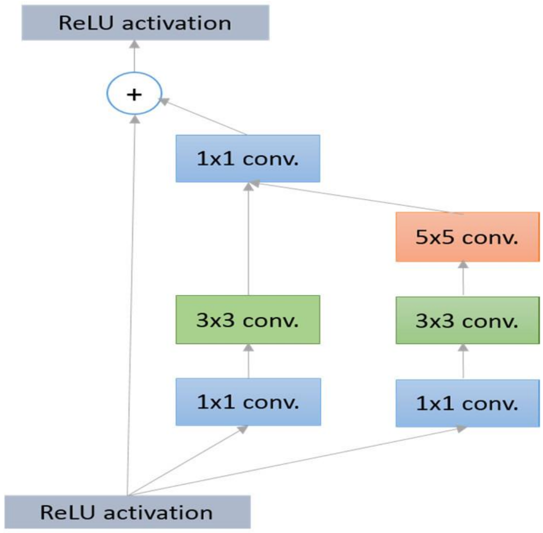

Figure The basic block diagram for Inception Residual unit.

RESULT & DISCUSSION

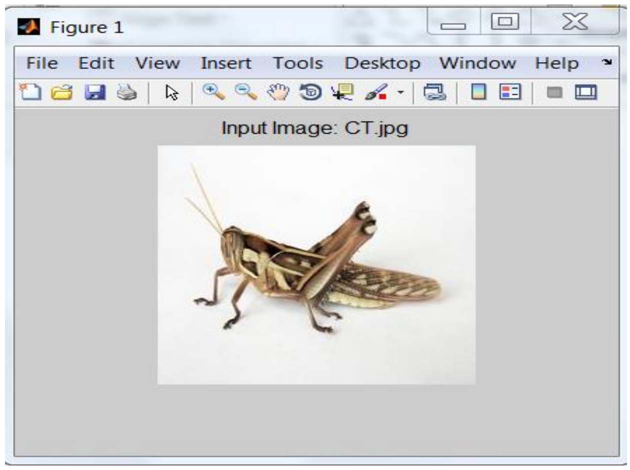

Figure image: Input image

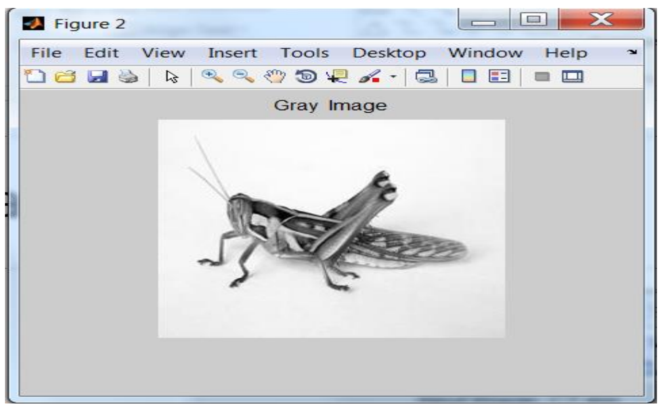

Figure:Grey image

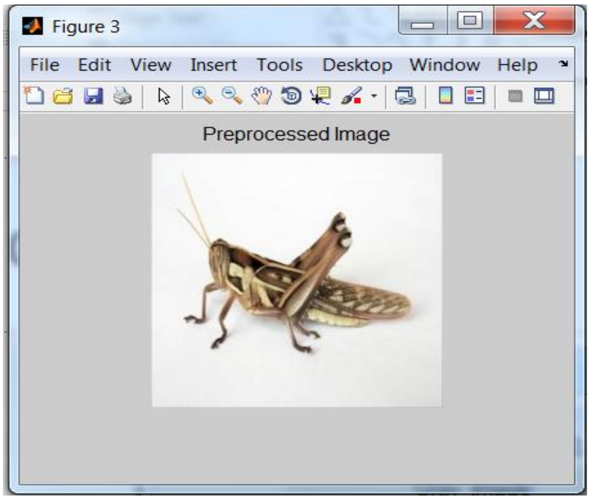

Figure : Preprocessed Image

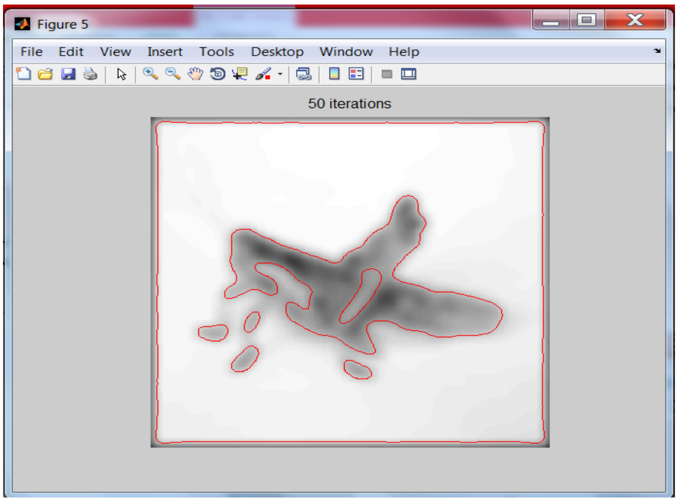

Figure:Feature Extracted Image

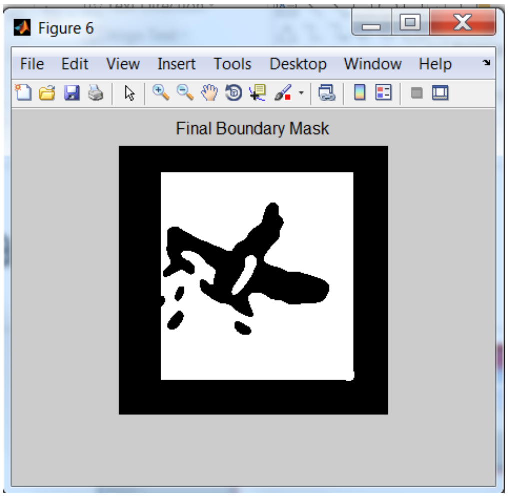

Figure: Figure Segmentation

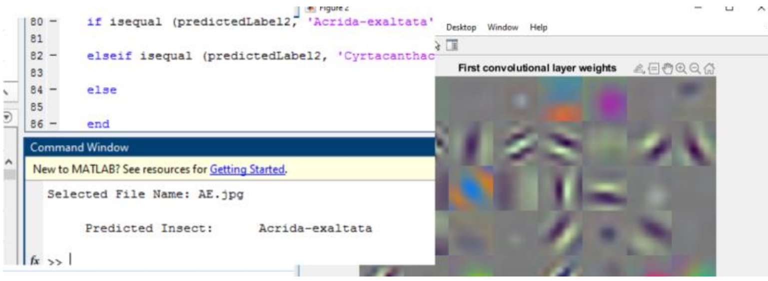

Figure: Output Classification.

CONCLUSION

In this project, we propose a locust recognition method based on improved ResNet, which accurately locates and recognizes flies. We designed the learning structure and introduced a bottom-up path augmentation to improve the low-level features semantic information and the high-level features location ability. The experimental results show that our proposed method have better performance compared with the state-of-the-art methods for fly species recognition. This is of great significance for the species recognition.

REFERENCE

[1] Nie, L., Wang, K., Fan, X., &Gao, Y. (2017). Fine-Grained Butterfly Recognition with Deep Residual Networks: A New Baseline and Benchmark. 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA).

[2] QuocBao Truong; Tan Kiet Nguyen Thanh; Minh Triet Nguyen; ,Shallow and Deep Learning Architecture for Pests Identification on Pomelo Leaf, 2018 10THINTERNATIONAL CONFERENCE ON KNOWLEDGE AND SYSTEMS ENGINEERING (KSE)

[3] Shi, Z., Dang, H., Liu, Z., & Zhou, X. (2020). Detection and identification of stored-grain insects using deep learning: a more effective neural network. IEEE Access, 1–1.

[4] Everton CastelãoTetila , Bruno Brandoli Machado , GeazyVilharvaMenezes, A Deep-Learning Approach for Automatic Counting of Soybean Insect Pests, IEEE GEOSCIENCE AND REMOTE SENSING LETTERS

[5] Roldan-Serrato, L., Baydyk, T., Kussul, E., Escalante-Estrada, A., & Rodriguez, M. T. G. (2015). Recognition of pests on crops with a random subspace classifier. 2015 4th International Work Conference on Bioinspired Intelligence (IWOBI).

[6] Ren, F., Liu, W., & Wu, G. (2019). Feature Reuse Residual Networks for Insect Pest Recognition. IEEE Access, 7, 122758–122768.

[7] Souza, W. S. R., Alves, A. N., & Borges, D. L. (2019). A Deep Learning Model for Recognition of Pest Insects in Maize Plantations. 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC).

[8] Long, D., Yan, H., Hu, H., Yu, P., &Hei, D. (2019). Research on Image Location Technology of Crop Diseases and Pests Based on Haar-Adaboost. 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS).

[9] Reza, M. T., Mehedi, N., Tasneem, N. A., &AshrafulAlam, M. (2019). Identification of Crop Consuming Insect Pest from Visual Imagery Using Transfer Learning and Data Augmentation on Deep Neural Network. 2019 22nd International Conference on Computer and Information Technology (ICCIT).

[10] Song, Y., Duan, X., Ren, Y., Xu, J., Luo, L., & Li, D. (2019). Identification of the Agricultural Pests Based on Deep Learning Models. 2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI).