Journal of Applied and Natural Science
Special Issue: 14 (SI), 138 - 144 (July, 2022)
ISSN : 0974-9411 (Print), 2231-5209 (Online)
journals.ansfoundation.org

Research Article

Android application development for identifying maize infested with fall armyworms with Tamil Nadu Agricultural University Integrated proposed pest management (TNAU IPM) capsules

R. Prabha
Department of Agricultural Entomology, Agricultural College and Research Institute, Tamil Nadu Agricultural University, Coimbatore- 641003 (Tamil Nadu), India
J. S. Kennedy*
Department of Agricultural Entomology, Agricultural College and Research Institute, Tamil Nadu Agricultural University, Coimbatore- 641003 (Tamil Nadu), India
G. Vanitha
Department of Physical Science and Information Technology, Agricultural College and Research Institute, Tamil Nadu Agricultural University, Coimbatore- 641003 (Tamil Nadu), India
N. Sathiah
Department of Agricultural Entomology, Agricultural College and Research Institute, Tamil Nadu Agricultural University, Coimbatore - 641003 (Tamil Nadu), India
M. Banu Priya
Department of Agricultural and Information Technology, Agricultural College and Research Institute, Tamil Nadu Agricultural University, Coimbatore - 641003 (Tamil Nadu), India

*Corresponding author. Email: jskennedy@tnau.ac.in

Article Info
https://doi.org/10.31018/jans.v14iSI.3599
Received: March 10, 2022
Revised: May 9, 2022
Accepted: June 2, 2022

How to Cite
Prabha, R. *et al.* (2022). Android application development for identifying maize infested with fall armyworms with Tamil Nadu Agricultural University Integrated proposed pest management (TNAU IPM) capsules. *Journal of Applied and Natural Science*, 14 (SI), 138 - 144. https://doi.org/10.31018/jans.v14iSI.3599

Abstract
Several pests and diseases wreak havoc on maize crops worldwide. Novel and rapid methods for detecting pests and diseases in real-time will make monitoring them and designing effective management measures easier. In the recent past, maize has been imperilled by fall armyworms (*Spodoptera frugiperda*), which have caused substantial yield losses in maize. This study aimed to create an Android mobile application via DCNN (Deep Convolutional Neural Network)-based AI pest detection system for maize producers. Everyone benefits from the deployment of these CNN models on mobile phones, especially farmers and agricultural extension professionals because it makes them more accessible. Automatic diagnosis of plant pest infestations from captured images through computer vision and artificial intelligence research is feasible for technological advancements. Therefore, early detection of maize fall armyworm (FAW) infestation and providing relevant recommendations in maize could result in intensified maize crop yields. . An Android mobile application was created to identify fall armyworm infection in maize and included the recommendations given by Tamil Nadu Agricultural University proposed Integrated Pest Management (TNAU IPM) capsules in the mobile app on as to how to deal with such a problem. Digital and novel technology was chosen to address these issues in maize. Deep convolutional neural networks (DCNNs) and transfer learning have recently moved into the realm of just-in-time crop pest infestation detection, following their successful use in a variety of fields. The algorithm accurately detects FAW (*S. frugiperda)* infected areas on maize with 98.47% training accuracy and 93.47% validation accuracy.

Keywords: Artificial intelligence, Deep convolutional neural networks, Fall armyworm, Mobile app.

INTRODUCTION

Deep learning is a branch of machine learning that uses artificial neural networks to educate computers to perform acts that humans would consider natural. In deep learning, a computer learns to classify directly from images, text, or sound (Dyrmann et al., 2016). The convolutional neural network (CNN) approach is used to identify FAW (fall armyworms) infestation in corn plants, and this method can be utilized to identify fall armyworm pests (*Spodoptera frugiperda*) infestation in corn plants.

Artificial intelligence (AI) plays an essential role in precision agriculture because of its flexibility, high performance, accuracy, and cost-efficiency, and artificial intelligence (AI) plays an essential role in precision agriculture (Patricio and Rieder, 2018; Eli-Chukwu, 2019). Deep CNNs have shown potential in agricultural areas for detecting plant pests and diseases. Several studies have used image datasets to create CNN models for pest and disease detection in plants such as tomato, banana, apple, cassava, cherry, alfalfa, wheat, and grapevine (Barbedo et al., 2014; Mohanty et al., 2016; Sladojevic et al., 2016; Amara et al., 2017; Brahimi et al.,2017; Fuentes et al., 2017; Ramcharan et al., 2017; Mkonyi et al., 2020). The deployment of these CNN models on mobile phones would benefit everyone, especially farmers and agricultural extension staff, making them more accessible.

Fall armyworm infestations on maize plants can be identified early and in real-time, which can help farmers manage pests and make better decisions. This research uses a mobile app called Agri app and a deep CNN model to detect and segment a maize fall armyworm (*Spodoptera frugiperda*) infestation on maize plants.

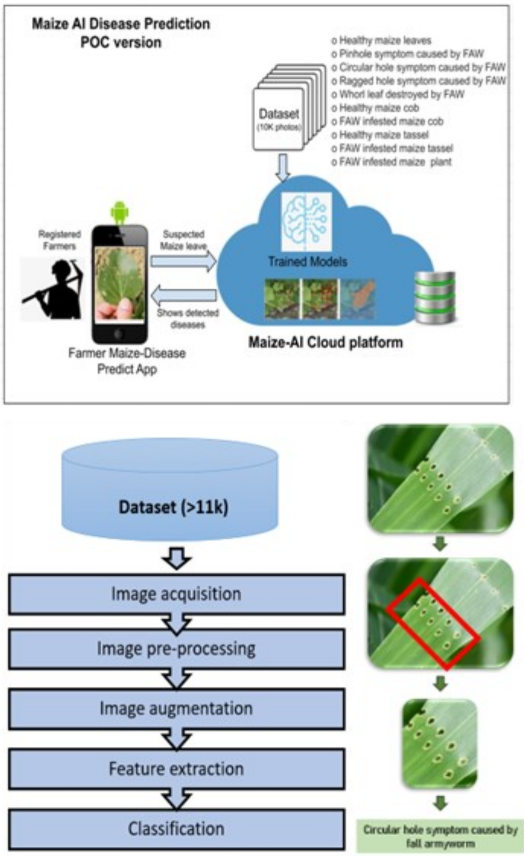

Fig. 1. *Artificial Intelligence (AI) framework for deep learning model development*

MATERIALS AND METHODS

Artificial Intelligence (AI) frameworks for deep learning model development

Image collection is the first and most crucial phase in developing an Android mobile app. More than 10,000 photographs of FAW-infested maize were taken using a digital camera (Nikon D 7500 P- Digital camera. It was then divided into 12 different classifications based on the score ratings, i.e., i). Health leaves, ii)Pinhole Symptom, iii) Circular hole symptom, iv) Ragged hole symptom, v) Whorl leaf, vi) damage, vi) Healthy cob, vii) <25% Cob area infested by FAW, viii) 25-50% Cob area infested by FAW, ix) 50-75% Cob area infested by FAW, x) >75% area infested, by FAW, xi) Healthy tassel, xii) Fall armyworm infested tassel. Google Collaboratory was utilized, which is an online notebook where Python algorithms were created to train an image dataset. Finally, the trained model was deployed to an Android phone using Android Studio. The developed mobile application allows farmers to identify FAW-infested maize images and bestows the relevant TNAU IPM recommendations. Image acquisition, image preprocessing, image augmentation, feature extraction and classification are all five steps to developing a deep learning convolutional neural network. The artificial intelligence (AI) framework is depicted in Fig. 1.

Convolutional neural network

CNNs are artificial neural networks that are used for image identification and processing pixel data for a specific design. Through its multilayered structure, it is compelling and evaluates graphical input images and finally extracts the relevant features from the input image. The convolution layer is the primary building block and performs computational tasks based on the convolution function. The pooling layer is arranged next to the convolution layer and is used to reduce the size of inputs by removing unnecessary information . A fully connected layer is arranged next to a series of convolution and pooling layers and classifies inputs into various categories. The fully connected layer employs the softmax classifier, a well-known input classifier, to recognize and categorize maize fall armyworm infestation. MobileNet V2 is a deep learning framework model to train our datasets. It also seeks to perform well on our mobile devices. Other framework models, such as Resnet and Inception, were also considered for model development. However, the accuracy level was very low in the case of ResNet and Inception. Therefore, the MobileNet V2 model is chosen to train the datasets.

Convolution operation

A 6*6 input picture was taken for the convolution procedure. To convert the 6*6 input image into a 4*4 output matrix, the 3*3 filter was chosen. The filter was slid

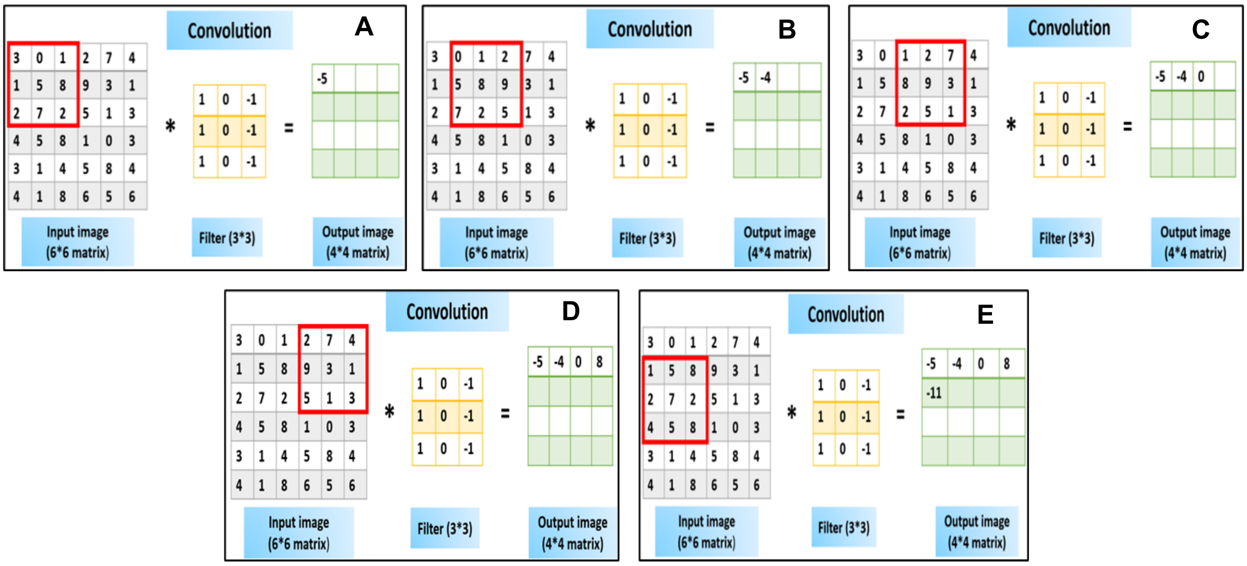

Fig. 2. *Convolution operation of input image data A. First mathematical algorithm convolution operation B. Second mathematical algorithm convolution operation C. Third mathematical algorithm convolution operation D. Fourth mathematical algorithm convolution operation, E.Fifth mathematical algorithm convolution operation*

over the top left corner of the pixel and began computing. The filter will compute the input pixel value by simply calculating it as 3*1+. As a result, the computed pixel value will be written here. The first convolutional mathematical algorithm operation was performed. The second convolutional operation was the same as the first one. The filter will slide over the next 3*3 pixel values. The pixel values will be computed as previously done. The second convolutional operation was done. Likewise, all the input pixel values are computed, and finally, the output pixel values are written in a 4*4 output matrix. The convolutional operation is illustrated in Fig. 2.

Pooling operation

The output of the convolution operation should be resized before it will be fed into another convolution operation. The input image will be resized from 4 * 4 to 2*2. The first 2*2 element was chosen. The one with the highest pixel value was chosen in max-pooling operations. In the output matrix, the highest pixel value will be written as the first output element. Then, the highest value of the next two pixels was taken. Similarly, in the output matrix, all of the highest values will be written. The max-pooling operation was also done. In an average pooling operation, the average of 4-pixel values will be written. In the case of the min pooling operation, the min pixel value of all 4-pixel values will be written. The Pooling operation is depicted in Fig. 3.

Image preprocessing

The acquired images are usually messy and unnormalized. To feed them to an algorithm, it should be cleaned. The image information was obtained by writing the simple code image-info. Colour space conversion was also done. The algorithm can be applied only to 2D (2-dimensional) or 3D (3-dimensional) matrices. RGB conversion is a 3-dimensional image, whereas a grayscale image is a 2-dimensional image. The histogram is the best method for image enhancement. It provides better quality images without the loss of any

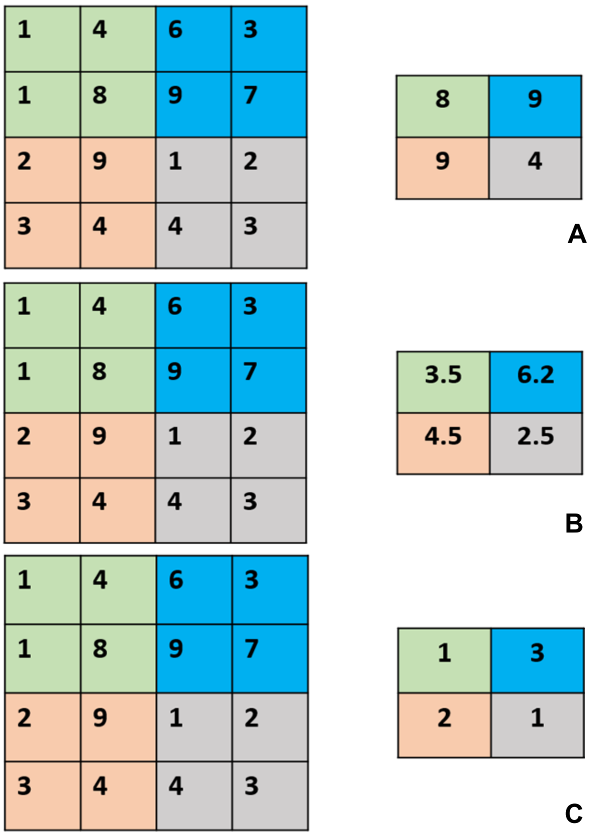

Fig. 3. *Pooling operations A. Max pooling operation. B. Average pooling operation. C. Min pooling operation*

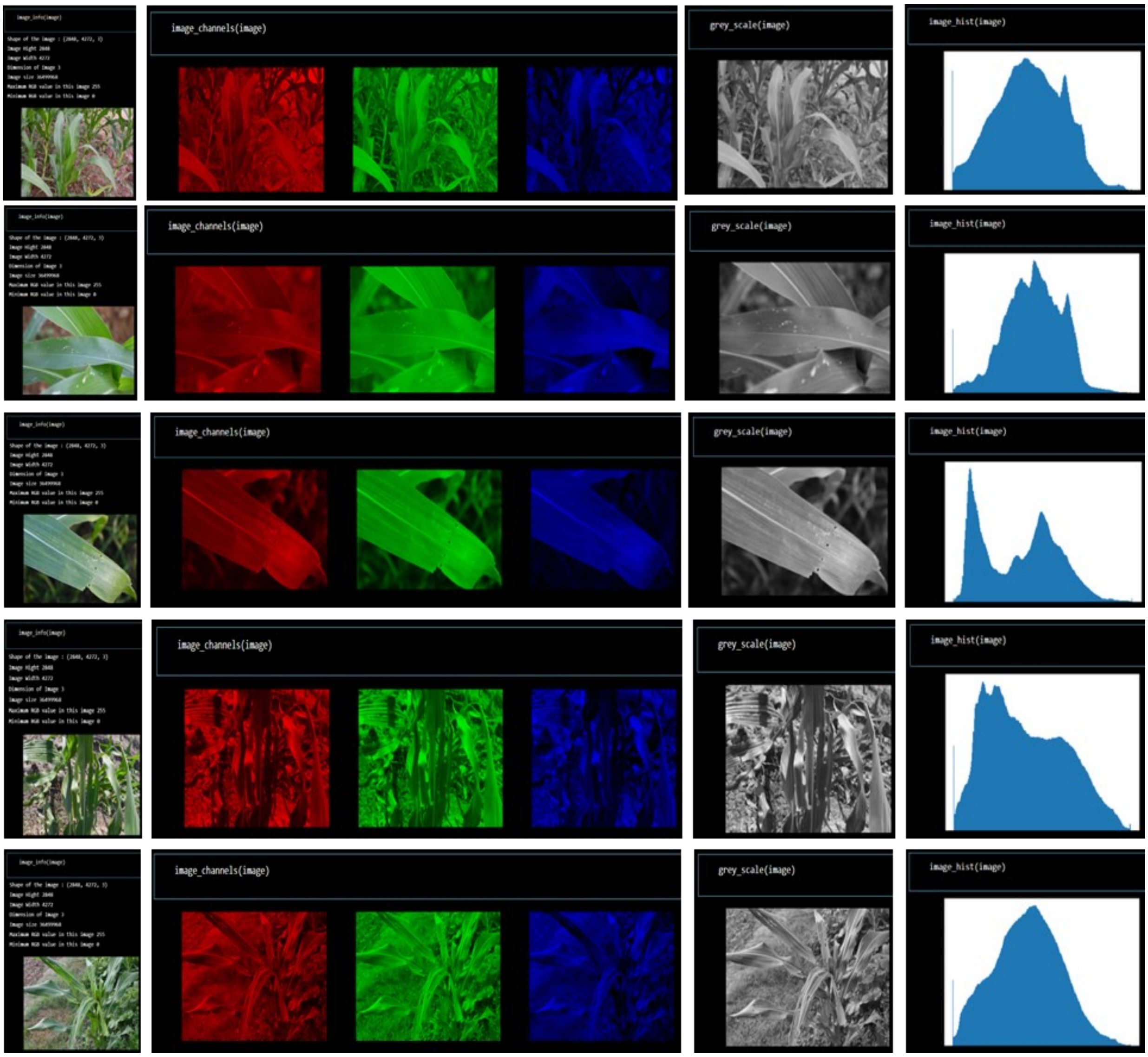

Fig. 4. *Image Preprocessing showing the original image, RGB (Red, Green, Blue) image, gray scale image, image histogram plot analysis to know the pixel values of the image*

information. To see the distribution of intensities in the image, a histogram plot was created by calling the imhist function. For all 12 classifications, the same image processing was performed and is depicted in Fig 4.

RESULTS

Output of the model

The number of epochs, learning rate and batch size was changed. No epochs were set as 50, the learning rate was set as 0.003, and the batch size was set as 512. Accuracy was also high in this model, and loss was also very low in this model. A training accuracy of 98.47% was obtained, and 93.42% validation accuracy was obtained in this deep learning model. Training loss of 0.052 percent was obtained, and validation loss of 0.20 percent was obtained. Therefore, this model is neither overfitting nor underfitting. Therefore, it is thought to be the best model for developing an Android mobile application. The accuracy in the mobile net V2 model was similar to the accuracy of 99.62% in other training datasets on 13 crop diseases and pest detection (Kulkarni, 2018). They worked on a mobile network using datasets from Plantvilla, an open-source dataset containing 54,306 photos of crop leaves sorted into 38 different classes. The collection includes 13 different crop species and 26 different illnesses. The training parameters for the deep learning model are illustrated in Table 1. The model accuracy and loss of the training and validation datasets are depicted in Fig. 5.

Deep learning framework comparisons

The mobile net V2 model was compared against resnet and inception framework models for training accuracy, validation accuracy, training loss, validation loss, and number of epochs. When compared to the resnet

Table 1. Training parameters for deep learning (CNN) model development

Training parameters for	Value
No of epochs	50
Batch size	512
Learning rate	0.003
Training accuracy	98.47
Validation accuracy	93.42
Training loss	0.0522
Validation loss	0.2065
Training data size	25853 with augmentation
Validation data size	8068 with augmentation

(28.90%) and inception (28.35%) framework models, mobile net V2 had a high level of accuracy (98.47%). Therefore, it was chosen for model development. A deep learning model comparison is illustrated in Table 2.

Android application development through android studio

The developed deep learning model was deployed in Android mobile phones using Android Studio, which is an official ide (integrated development environment) for Google and android operating systems and specifically designed for Android development. Its version was 4.1.2. Configuration decisions and plugin installation options were made to obtain flutter-based mobile applications. Two plugin choices were installed, i.e., flutter and dart. Flutter is an open-source user interface software development kit from Google. It is used to create cross-platform apps for Android, iOS, Linux, Mac, and Windows with a single codebase. Android Studio is an excellent alternative for learning Dart and developing Flutter apps. Dart is a programming language that may be used to create web and mobile apps depicted in Fig. 6. Only the image datasets and our unquantized model were imported and designed; therefore, an asset folder was created for them (Fig. 7). Three files were included under this: a laberl.txt folder, an icon folder, and an unquantized model folder. Finally, ADB WiFi from the plugin option was installed. It is **used** to bridge communication between an Android device and the background running process (server). It is also used for installing or debugging a device and running various commands on devices. Once the ADB WiFi is installed, connect our device by enabling USB debugging and developer options in our Android phone on which the

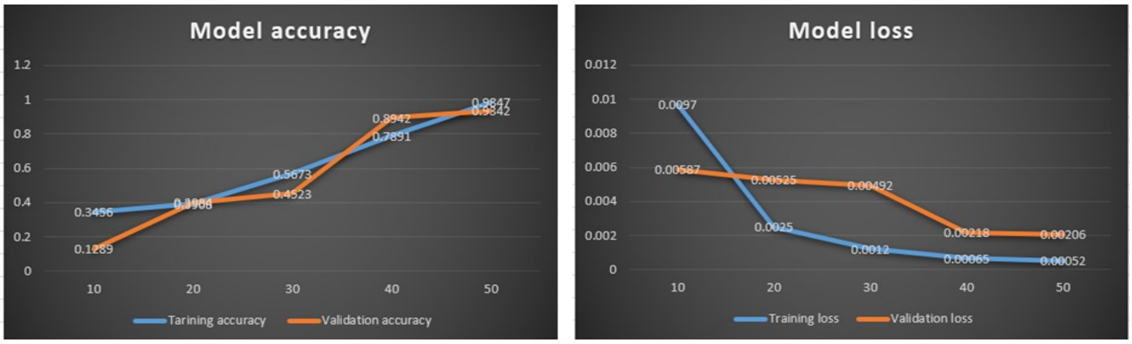

Fig. 5. *Model accuracy and loss of training and validation dataset*

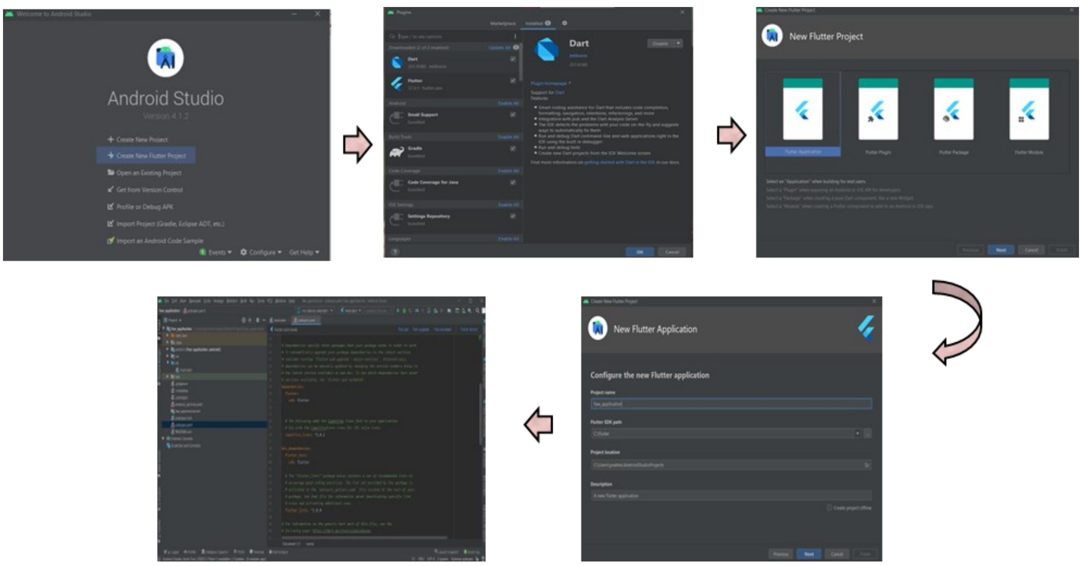

Fig. 6. *Android studio plugin options*

Table 2. Model comparison of deep learning framework

Deep learning framework	Number of epochs	Training accuracy	Validation accuracy	Training loss	Validation loss
Mobile Net V2	50	0.9847	0.9342	0.0522	0.2065
Inception	25	0.2835	0.2234	180.7142	178.5931
Resnet	25	0.2890	0.2877	2.0497	1.9305

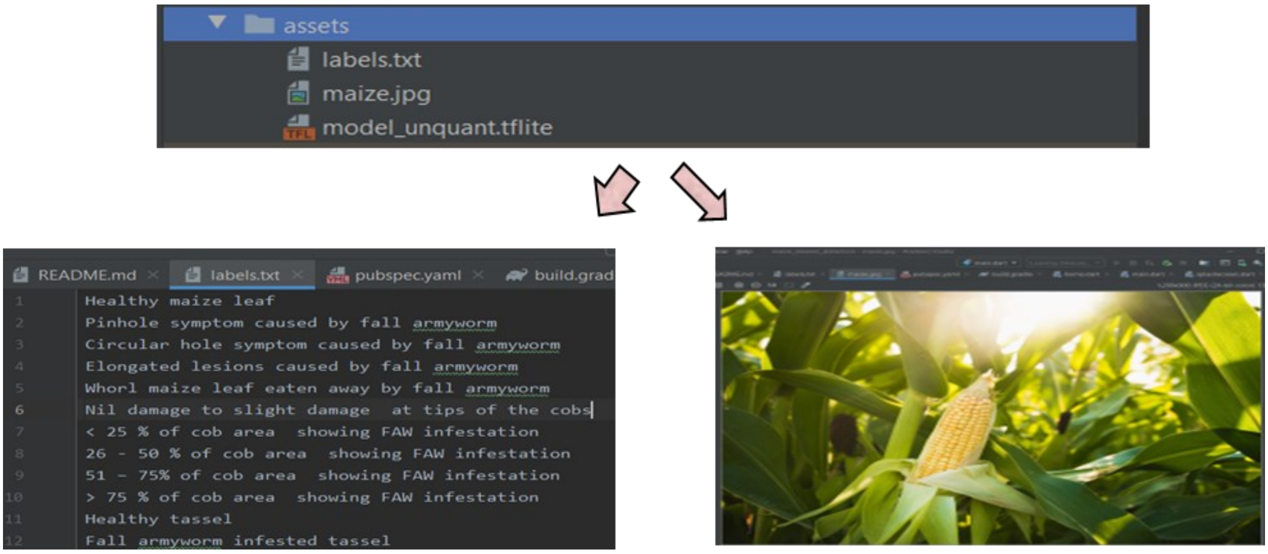

Fig. 7. *Assets folder in android studio*

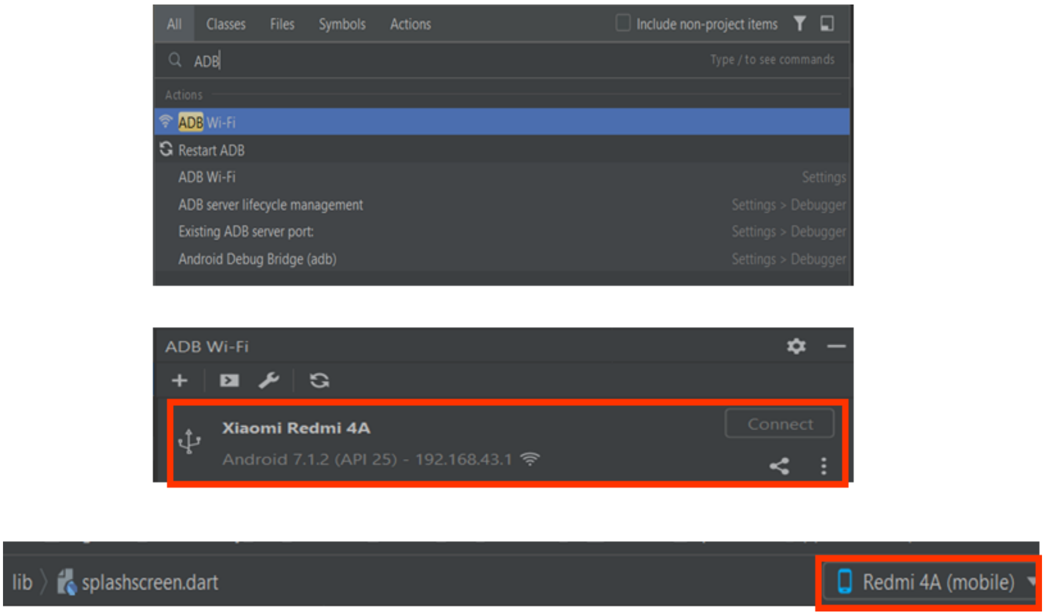

Fig. 8. *ADB WiFi option in android studio*

supplication will be running. It was then linked to the loading file. Then, we can be able to simply access our project from our phone (Fig. 8)

Specification of an android mobile application

The apk file size is 22mb. API (application programming interface) was incorporated. it is worked well only in online. It is also having the flutter environment. The MySQL database management system was employed in this POC version to save all of the pest information and ground-truth photographs for crop protection. TNAU recommendations were added to the POC (proof of concept) version. The refined IPM capsule (TNAU) has been added in version. The mobile application icon and interface are shown in Fig. 9. Testing accuracy was 99 percent. Field level validation is not yet done. It will be carried out by three levels of validation (extension officials, scientists and farmers). The present application (AIPES) development was also similar to the corn plant disease detection application development (Hidayat et al., 2019; Sibiya & Sumbwanyambe, 2019). They created a model for a variety of maize diseases, and the symptoms of each disease appear to be unique. This study focuses entirely on pest infestations rather than diseases. They developed the model and have yet to convert it into an Android app. The present study did work on the classification of several types of symptom infestation caused by Fall armyworm, a recently invasive pest in India, using CNN and Mobilenet framework approach. Finally, using Android Studio, the study created a mobile application that will be released commercially as soon as all of the necessary verification (at the level of farmers, extension officials, and agricultural scientists) is completed.

Conclusion

In the present research, one new android application was developed for the classification of fall armyworm infestation symptoms in leaves, cobs and as well as for tassels. The deep learning model was created via training the twelve different classification image datasets in Google collaboratory and tested three models, including Resnet, Inception and Mobilenet model framework. Among them, mobile net V2 had a high level of accuracy (98.47%) also implemented in the mobile application.

Conflict of interest

The authors declare that they have no conflict of interest.

REFERENCES

1. Amara,J., Bouaziz, B & Algergawy, A. (2017). A Deep Learning-based Approach for Banana Leaf Diseases Classification,". *Database systems for business, technology and web*,79–88.
2. Barbedo. A. (2014). An Automatic Method to Detect and Measure Leaf Disease Symptoms Using Digital Image Processing. *Plant Disease*. 98 (12), 1709–1716. https://doi.org/10.1094/PDIS-03- 14-0290-RE.
3. Brahimi, M., Boukhalfa,K & Moussaoui, A. (2017). Deep Learning for Tomato Diseases: Classification and Symptoms Visualization, *Applied Artificial Intelligence,* 31(4)., 299–315.https://doi.org/10.1080/08839514.2017.13 15516.
4. Dyrmann, M., Karstoft, H & Midtiby, H.S. (2016). Plant species classification using deep convolutional neural network, *Biosyst. Eng*, 72–80.
5. Eli-Chukwu. C. (2019). Applications of Artificial Intelligence in Agriculture: A review, Engineering, Technology & Applied Science Research. 9(4), 4377–4383, https://doi.org/10.48084/etasr.2756.
6. Fuentes, A., Yoon, S., Kim, S.C & Park, D.S. (2017). A Robust Deep Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. *Sensors*, 17 (9). https://doi.org/10.3390/s17092022.
7. Hidayat, U., Darusalam & Irmawati, I. (2019). Detection of disease on corn plants using convolutional neural. *Journal of Computer Science and Information*, 12(1), 51-56.
8. Kulkarni, O., (2018). Crop disease detection using deep learning. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 1-4, IEEE.
9. Mkonyi, L., Rubanga, D., Richard, M., Zekeya, N., Sawahiko, S., Maiseli, B. & Machuve, D., (2020). "Early identification of Tuta absoluta in tomato plants using deep learning," *Scientific African*. https://doi.org/10.1016/j.sciaf.2020.e00590.
10. Mohanty, P., Hughes, P & Salathe, M. (2016). "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, 7. https://doi.org/10.3389/fpls.2016.01419.
11. Patricio, I & Rieder, R. (2018). Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review, *Computers and Electronics in Agriculture*, 53, 69–81. https://doi.org/10.1016/j.compag.20 18.0 8.001.
12. Ramcharan, K., Baranowski, P., McCloskey, B., Ahmed, J., Legg & Hughes, P. (2017). Deep Learning for Image-Based Cassava Disease Detection, *Frontiers in Plant Science*, 8. https://doi.org/10.3389/fpls.2017.01852.
13. Sibiya, M & Sumbwanyambe, M. (2019). A computational procedure for the recognition and classification of maize leaf diseases out of healthy leaves using convolutional neural networks, *Agri Engineering*, 1(1), 119-131.
14. Sladojevic,S., Arsenovic, M., Anderla, A., Culibrk, D. & Stefanovic, D. (2016). Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification, *Computational Intelligence and Neuroscience*. https://doi.org/10.1155/2016/3289801.