

PREDICTIVE MODEL FOR MAIZE STEM BORERS' CLASSIFICATION IN PRECISION FARMING

Ezeofor Chukwunazo¹, Akpado Kenneth² and Ulasi Afamefuna³

¹Department of Electrical/Electronic Engineering,
University of Port Harcourt, Rivers State, Nigeria

²Electronic and Computer Engineering,
NnamdiAzikiwe University, Awka, Anambra State, Nigeria

³Department of Information Management, Federal University of Technology,
Owerri, Imo State, Nigeria

ABSTRACT

This paper presents Predictive Model for Stem Borers' classification in Precision Farming. The recent announcement of the aggressive attack of stem borers (Spodoptera species) to maize crops in Africa is alarming. These species migrate in large numbers and feed on maize leaf, stem, and ear of corn. The male of these species are the target because after mating with their female counterpart, thousands of eggs are laid which produces larvae that create the havoc. Currently, Nigerian farmers find it difficult to distinguish between these targeted species (Fall Armyworm-FAW, African Armyworm-AAW and Egyptian cotton leaf worm-ECLW only) because they look alike in appearance. For these reasons, the network model that would predict the presence of these species in the maize farm to farmers is proposed. The maize species were captured using delta pheromone traps and laboratory breeding for each category. The captured images were pre-processed and stored in an online Google drive image dataset folder created. The convolutional neural network (CNN) model for classifying these targeted maize moths was designed from the scratch. The Google Colab platform with Python libraries was used to train the model called MothNet. The images of the FAW, AAW, and ECLW were inputted to the designed MothNet model during learning process. Dropout and data augmentation were added to the architecture of the model for an efficient prediction. After training the MothNet model, the validation accuracy achieved was 90.37% with validation loss of 24.72%, and training accuracy 90.8% with loss of 23.25%, and the training occurred within 5minutes 33seconds. Due to the small amount of images gathered (1674), the model prediction on each image was of low confident. Because of this, transfer learning was deployed and Resnet 50 pre-trained model selected and modified. The modified ResNet-50 model was fine-tuned and tested. The model validation accuracy achieved was 99.21%, loss of 3.79%, and training accuracy of 99.75% with loss of 2.55% within 10mins 5 seconds. Hence, MothNet model can be improved on by gathering more images and retraining the system for optimum performance while modified ResNet 50 is recommended to be integrated in Internet of Things device for maize moths' classification on-site.

KEYWORDS

Fall Army Worm, African Army Worm, Egyptian Cotton Leaf Worm, Precision Farming, Convolutional Neural Network, Maize crop, ResNet, Stemborers

1. INTRODUCTION

The emergency of machine learning (ML) and big data technologies have open new approach of data mining and solving agricultural intensive problems [1]. In addition, techno stacks [2] posted report on machine learning as the trending technology used in modern agriculture sector to improve quality of the crops and productivity is machine learning. In agricultural sector (such as

digital farming, smart farming, precision farming), machine learning is one of the fastest growing areas. In precision farming, machine learning has gained serious attention and now applied to ensure optimal yield of crops in the farm such as yield prediction and quality assessment, species identification and classification, crop disease and weed detection, etc. ML is categorized into four types namely supervised, unsupervised, semi-supervised and reinforcement; each algorithm is chosen based on the Scientific and Engineering problems at hand. Since maize insect species identification and classification is one of the major problems all farmers face daily, supervised machine learning is preferred algorithm used to tackle such. In supervised machine learning, a machine is trained with set of a given input dataset in order to gain experience and carry out successful prediction. The Convolutional Neural Network (CNN) is chosen, designed and trained from the scratch because it is meant for object detection and classifications. The customized CNN algorithm (MothNet) is trained with large image dataset of maize species captured in the farm. Two Architectures of pre-trained CNN algorithm such as ResNet and MobileNet models are modified and subjected to learn and compared with MothNet model and best model selected for perfect prediction of these maize species.

2. REVIEW OF RELATED WORKS

Konstantinos P.F [3] presented plant disease detection and diagnosis using deep learning models. Various deep learning models were deployed based on convolutional neural network architecture for images of healthy and diseases plants. Eight-seven thousand, eight hundred and forty (87840) photographs comprising of 25 plant species in 58 classes of these plant images (both diseases and healthy) were used for the training. VGG model outperformed the other models with 99.53% accuracy and 0.4% loss. He also reported that the trained model used about 2 minutes on a single Graphic processing unit with low computational power to diagnose and detect any disease present in a plant leaves. He recommended further the development of an automated pesticide prescription system capable of performing plant disease diagnosis accurately for farmers' use of appropriate pesticides.

Simon et al [4] developed Fall Army Worm (FAW) image capturing and identification using machine-learning algorithm due to the havoc created by this said moth in the farm. They explained that their research was triggered because of stakeholders of Food and Agriculture Organization and University of Zambia demand to tackle the problem. They deployed transfer learning technique by retraining the pre-trained inception v3 model on the captured insect dataset via Google TensorFlow platform for the FAW image identification. Their trained model achieved train accuracy of 57-60%, cross entropy of 65-70% and validation accuracy of 34-50%. They recommended researchers to further improve on their system should enhance the training in Google TensorFlow and deployed to Raspberry Pi for onsite testing.

Belal et al [5] presented deep learning technique for plant seedlings classification. Dataset contained approximately 5000 images with 960 plants for different 12 species. They used convolutional neural network (CNN) algorithms applied to image recognition and achieved their target. The trained CNN model achieved an accuracy of 99.48% on a held-out test set. With this model, plant seedlings classification is possible to improve on the farm produce.

Kawche et al [6] presented machine-learning techniques for rice leaf diseases detection. They selected three different rice leaf diseases such as leaf smut, bacterial leaf blight and brown spot disease for the experiment. Four machine-learning algorithms (K-Nearest Neighbor, decision tree, logistic regression and Naïve Bayes) were used for the detection of the aforementioned rice leaf diseases. Decision tree outperformed other algorithms with 97.916% accuracy on test data. They recommended using ensemble-learning algorithm on the same dataset to check its performance and use to evaluate theirs.

Abhishek et al [7] presented Statistical Feature Extraction and Image Classification Methods for whitefly pest automatic detection. The proposed system would help in classifying white fly infected leaf images from normal leaf images. Gray Level Run Length Matrix (GLRLM) and

Gray Level Co-occurrence Matrix (GLCM) were the statistical feature extraction methods used to extract features from the leaf images. Also various classifiers such as Bayesian, Support Vector machine (SVM), Binary decision tree, Artificial Neural Network and k-Nearest neighbor were used to classify the infected leaf images. SVM outperformed the other classifier with an accuracy of 98.46%

3. ARTIFICIAL INTELLIGENCE SYSTEM DEVELOPMENT

Developing the system intelligence requires different stages (figure 1). When followed, these stages ensure successful development of the model for maize insect pests' recognition and classification in precision farming.

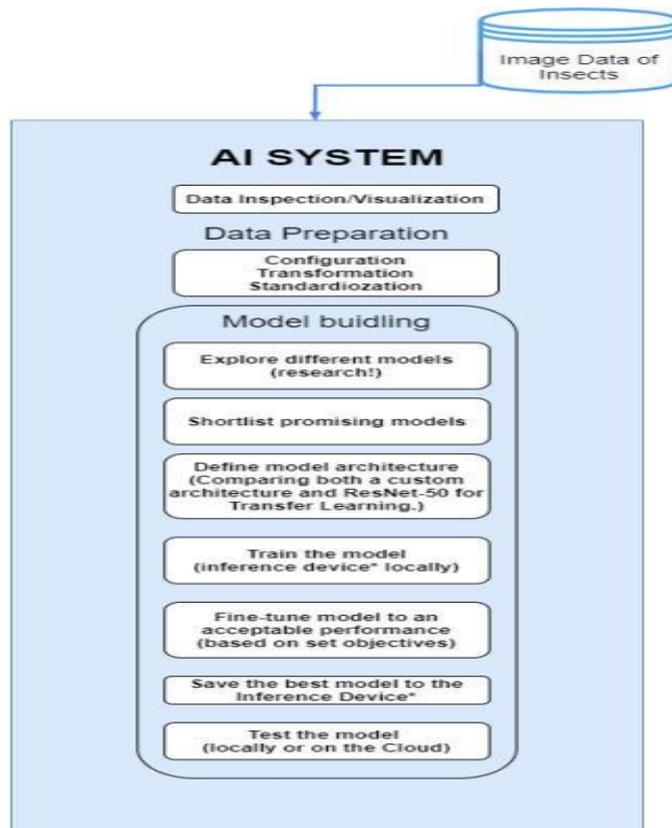


Figure 1. Artificial Intelligence development stages

3.1. Maize Moths Data Capturing Techniques

Two methods of data capturing were adopted. The first method used was to set-up delta pheromone trap in the farm for a catch while the second method, was to breed the moths larvae in the laboratory. Three different delta pheromone traps for three categories of the maize moths were set-up in the farm and labelled accordingly. The targeted and labelled maize moths were

Fall Army Worm (FAW), male Africa Army Worm (AAW), and male Egyptian Cotton Leaf Worm (ECLW). The installed delta traps with lures attract the male of these moths and catch them. The species are nocturnal moths, which made them to operate and create havoc mainly at night hours. When they are trapped, high definition phone camera was used to take the image of the moths including the video footage. The maize farm, trapped moth and three categories of the targeted moths are shown in figure 2.

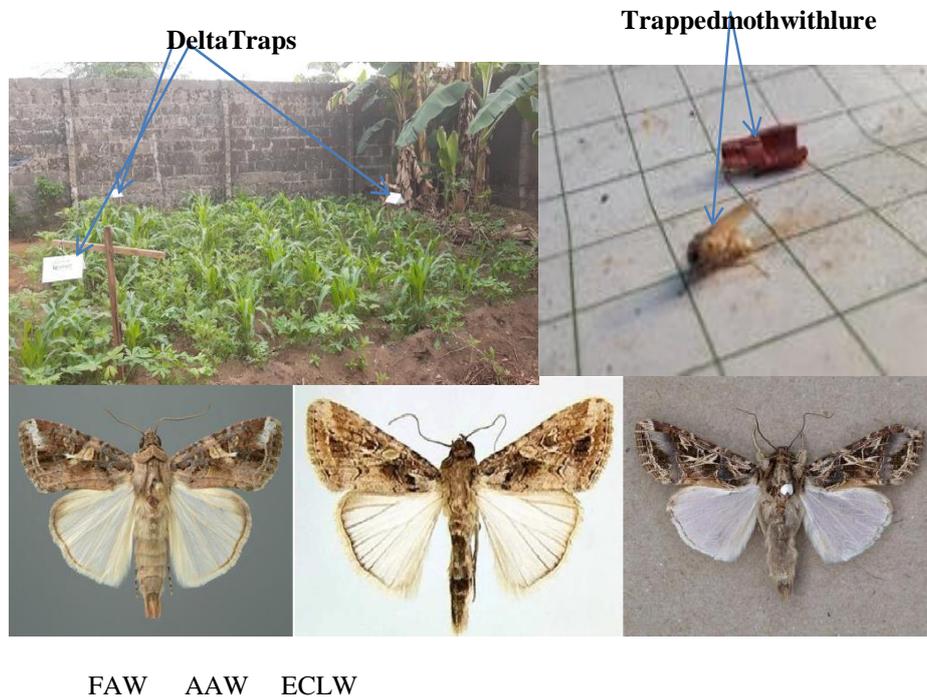


Figure 2. The set trap, trapped moth with lure and Spodoptera species

False environment was created in order to capture more maize moths for the system training. A rectangular glass box was used to breed the larvae of the species to adult male and female species. Cardboard sheet was spread on the inside-bottom part of the box and loamy soil poured on it 1cm distance from the bottom. Maize farm was visited and those maize plants already on attack from the larvae of the species were cut and placed inside the box. Water was spread inside the box to moisten the environment and the box covered with black cloth to create dark environment for the moths as shown in figure 3. The setup was visited daily to moisture the environment and enable carbon dioxide to escape and fresh oxygen circulate within the box for an hour. The process was repeated for three weeks (21 days) before the larvae pupated into full matured adult male and female. Then images were taken and added into the dataset to increase the numbers for efficient system training.



Figure 3. Adult Moths' breeding in the Lab

3.2. Data Preparation and Pre-processing

After capturing the images of the targeted three *Spodoptera* species (FAW, AAW, ECLW), data preparation and pre-processing were carried out in two phases namely data inspection, image dataset splitting and data visualization.

Maize Species Image Data Set Inspection: In data preparation, the images of these maize moths were filtered and selected by the Entomologist and educated farmers respectively to ensure that the true moths required are captured for the model-building phase of the machine learning development process. Duplicated and similar frames (images) were deleted using python scripts to ensure effective performance of the deep learning algorithm. Further processing was done to the images in the image dataset folder. Each image was resized using written python script to size of 500x500 pixels and 224 x224 pixels. This is to ensure that an efficient format acceptable by the Convolutional Neural Network model is generated. The final labeled image species of about one thousand, six hundred and seventy-four (1674) were saved in the image dataset folder. The size of the filtered labeled image dataset was over 4GB and was compressed using python script to reduce the size because the images required during model building do not need to be super-high resolution for them to work with our deep learning algorithm. The gathered image dataset was uploaded to the online Google drive for storage and use for the model training.

Image Data Splitting: the proposed predictive model (MothNet) for maize species classification would be evaluated to ensure that the targeted system is robust. This requires that the image dataset be split into training set, validation/development test, and testing set. The training set would be used by the deep learning algorithm to learn the distinguishing patterns across the various species, and finally make predictions based on the input data. The validation set would be used to evaluate our experiments, and then the test set would be used to evaluate the performance of the trained Model for generalizability. Since 1674 species images could be gathered, 70:30 rule of thumb was applied to ensure perfect data set splitting. This simply means that 70% of image data set would be for training set and 30% for validation set. The 30% for the validation set was further split into 20% for validation/development set, and 10% for test set. In splitting the image data set, the `tf.keras.preprocessing.image_dataset_from_directory` function was used. The batch size is set to 20 images per batch to help the system perform optimally during training phase. This means that the image data is fed to the algorithm eighty-four (84) times to complete 1674 images in the dataset during training.

Image Data Visualization: Data visualization aids the system developer to view the variation of images present in the data set. This helps to ensure that the images are properly formatted and the

moths' appearance are good before feeding them into the algorithm. Matplotlib.pyplot library was used to achieve this visualization and part of the labeled images (figure 4) in the data set was imported randomly using matplotlib.

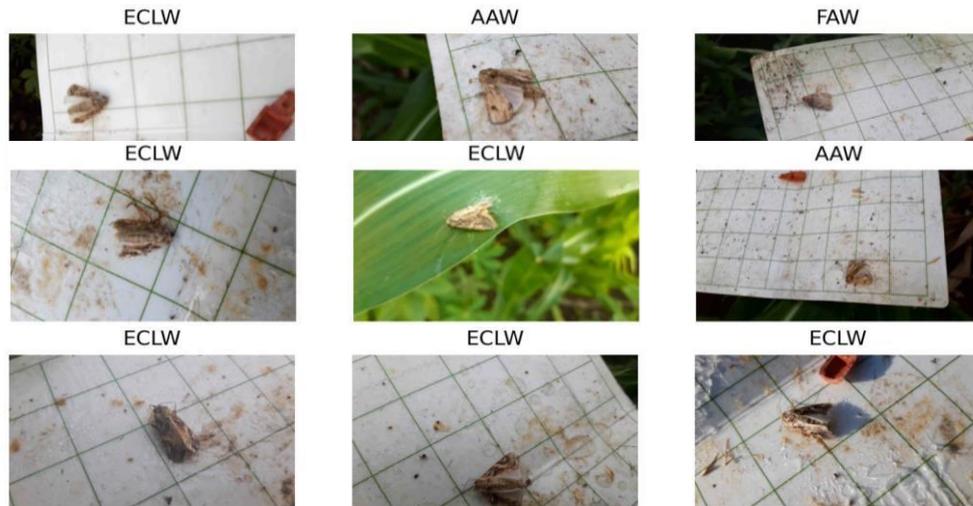


Figure 4. Sampled resized images of the insects in

3.3. Selecting and Building Neural Network Model from Scratch

In selecting and building a model suitable to handle the insect image classification problem at hand, the best deep learning algorithm for image recognition and classification over the years is chosen. A Convolutional Neural Networks (CNNs) is a deep learning algorithm design to recognize and classify features in images for computer vision. It is a multi-layer neural network designed to analyze visual inputs and perform tasks such as image classification, segmentation and object detection, which is useful for Maize species detection and recognition. TensorFlow is used to develop machine learning systems, and Keras, a deep learning application programming interface (API) that runs on top of the TensorFlow, is used for model building and experimentation.

Custom CNN Model Architecture: The developed CNN Model architecture named “MothNet” is made up of input layer, rescaling layer, 2-dimensional convolutional layer and a max-pooling layer, a second 2D convolutional layer and a max-pooling layer, a third 2D convolutional layer and a max-pooling layer, and then a flatten layer, a dense layer, and an output layer as shown in figure 5. The input layer accepts an image size of 500 height x 500 width with RGB color channels moth image and sends it to the rescaling layer. The rescale- layer normalizes the image pixel values from 0 to 255 to 0 to 1, as deep learning algorithms perform much better when a value is within this range. The output is sent to the convolutional layer1 which convolves the input image with sixteen 3x3 kernels. The sixteen outputs are generated and sent to max-pooling layer1 for reduction. The reduced layers are sent to convolutional layer 2 to convolve with a higher thirty-two 3x3 kernels in order to pick more features from the image. The outputs are further reduced by passing them to max-pooling layer2. The output layers are passed to convolutional layer 3 which convolves with sixty-four 3x3 kernels for more features extractions. The sixty-four outputs are further reduced with max- pooling layer 3. The final outputs in 2D pattern is transformed into 1-D by passing them through flatten layer. The dense layer uses Softmax to classify the insects into 3 categories.

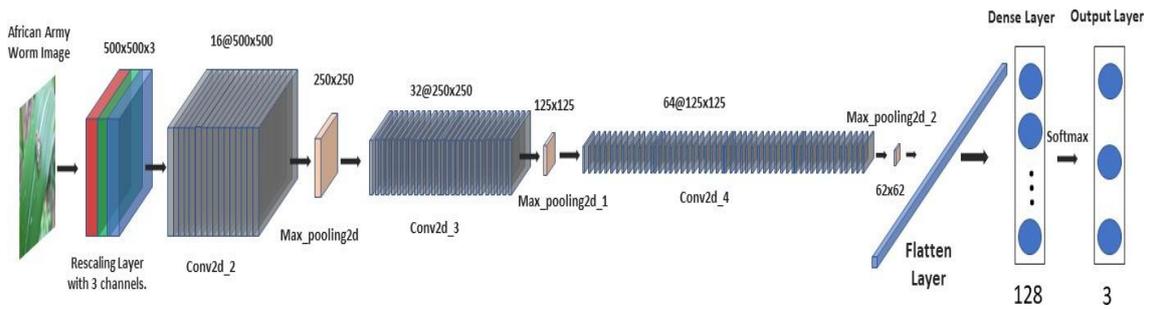


Figure 5. Diagram of Custom CNN (MothNet) Model Architecture

Custom CNN Model Training: The customized CNN model (MothNet) training platform used is online Google Colab with python programming language. The image dataset (1674 images) was loaded from Google drive to Google Colab. The dataset converted to tensors using TensorFlow API to help work with the data. During training, the TensorBoard visualization tool was used to monitor the progress of the training. During training, the system learns following nine stages of designed convolutional pattern. Python scripts were written to achieve stage 1 through stage 9.

Stage 1: The captured insect image was rescaled into 500x500x3 size (figure 6)) through rescaling layer; then, normalized and sent to the convolutional layer 1.



Figure 6. Rescaled insect image for processing

Stage 2: The convolution layer 1 applies sixteen (16) different 3x3 kernels to the input image (500x500) to generate sixteen feature maps image of 500x500 (figure 7), indicating the strength of a detected feature in each of the input image. The activation function called rectified linear unit (ReLU) is applied to the various generated feature maps in order to overcome the vanishing gradient problem, allowing models to learn faster and perform better. The layer outputs the input directly if it is positive, and zero if it's negative. The final feature maps are further scaled down by passing them to max-pooling layer in stage 3.

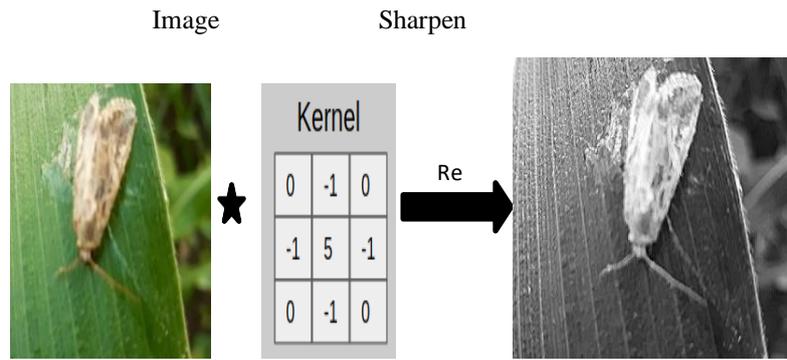


Figure 7. 500x500x16 Image feature map extraction

Stage 3: Max-pooling layer calculates the largest value in each patch of the sixteen feature maps and generates 250x250x16, then send the result to convolutional layer 2.

Stage 4: The convolutional layer 2 further applies thirty-two (32) filters to the resultant feature maps 250x250 in order to detect more features in the insect image. An activation function (ReLU) is applied to eliminate the negative values in the system and result sent to max-pooling layer of stage 5.

Stage 5: Max-pooling layer further calculates the maximum value in each patch of the thirty- two feature maps of 250x250 and generates 125x125 and sends the result to convolutional layer 3.

Stage 6: The convolutional layer 3 applies sixty-four (64) filters to the resultant feature maps 125x125 in order to deeply detect remaining features in the insect image. An activation function is applied to eliminate the negative values in the system and result sent to max-pooling layer of stage 7.

Stage 7: Max-pooling layer calculates the maximum value in each patch of the sixty-four feature maps of 125 x125 to generate 62x62 and send the result to the flatten layer

Stage 8: The flatten layer converts convolutional layer 2-D image feature outputs of 62x62x64 into 1-D fully connected layer of 246016 outputs in a single column and submit same to dense layer.

Stage 9: The dense layer with 128 neurons uses activation function called Softmax to classify the maize insect pests into FAW, ECLW, and AAW. The trained MothNet model was finally optimized, compiled and transferred into the Jetson Nano memory for image recognition of maize insects. The learning flow diagram of the MothNet model is shown in figure 8.

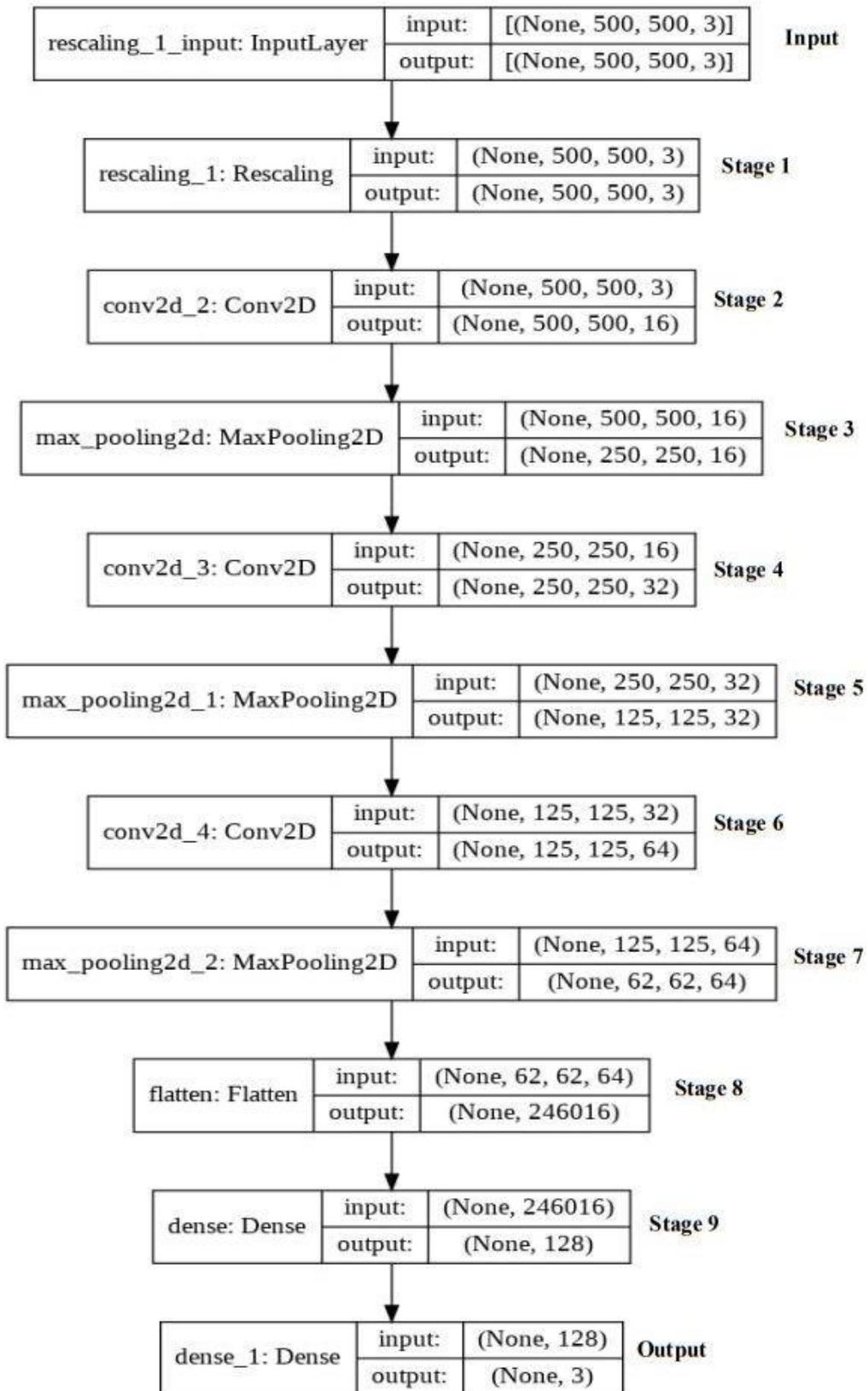


Figure 8. Flow Chart steps for MothNet model training

MothNet Model Training performance: The system was trained using 1674 images in the dataset. The training and validation accuracy and loss plots are shown in figure 9. There was an over fitting noticed on both accuracy and loss graphs. The training accuracy was 100% accuracy but the validation accuracy 93% which showed an over-fitting. The same happened to the training and validation loss. In loss graph, the training achieved 0% loss but the validation achieved 23% loss which is not efficient after training. This was mostly due to the small image dataset captured and the network model needed some hyper-parameter tuning as well. The system was improved upon by applying dropout and data augmentation.

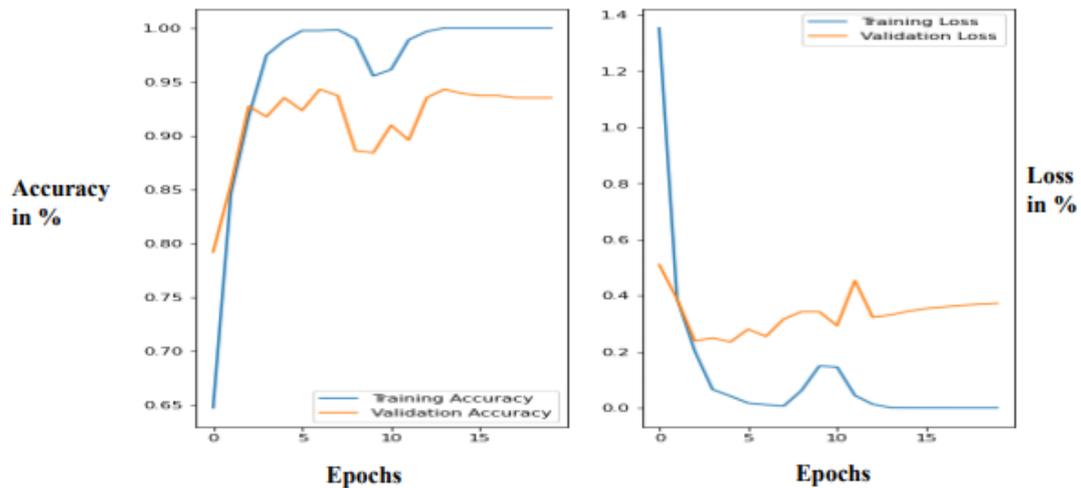


Figure 9. MothNet Training and validation accuracy and loss

Dropout and Data Augmentation: The image size of 500x500 was rescaled and used for the system training but noticed that the system overfits when viewed online via TensorBoard Development environment, which makes the model not efficient. In order to learn relevant features from the image and improve on the MothNet model, dropout layer with 20% was added between the max-pooling layer 3 and flatten layer. This means that 20% of the learnable parameters are dropped. The effect of adding the random dropout is that the function forces the network to learn less redundant representations of image dataset thereby making it more robust and less prone to overfitting.

Augmenting image data set help avoid overfitting the training data since limited images were captured. Also, noting that the dataset did not account for all the future predictions of the targeted maize insect pests such as image on a fly, image on different angles, etc. Image random transformation is achieved through data augmentation which added more styles and position to the images in the data set. This augmentation is done during training which rotates the image as it passes through the model's stages. After the introduction of the drop out layer in the architecture and performing data augmentation, the enhanced training and validation accuracy and loss graphs plot for 20 epochs are shown in figure 10.

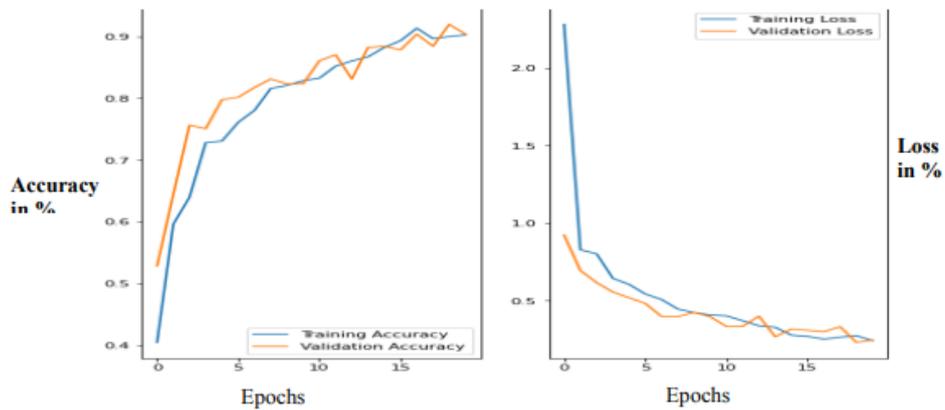


Figure 10. Enhanced Training and Validation accuracy and losses

Looking at the graph, the new training accuracy achieved was 42% at 0 epoch and progressed to 90% at 20 epochs while validation accuracy achieved was 53% at 0 epochs and progressed to 90% at 20 epochs whereas the training and validation losses reduced to 10% at 20 epochs. This shows that the system did not overfit and has improved during training, for optimal recognition of the species. The validation accuracy achieved is 90.37% and validation loss is less than 10% while training accuracy was 90.8% with training loss of 23.25% are achieved, and training occurred within 5minutes 33seconds.

MothNet Model Test Result: Five images were picked from each categories of the maize moths reserved from the remaining 30% image dataset. Each image of these species was passed to MothNet model via online Google Colab and the model returned its prediction on each image. The MothNet model prediction accuracies on Fall Army Worm adults (FAWs), African Army Worm adults (AAWs) and Egyptian Cotton Leaf Worm adults (ECLWs) are showcased in figure 11, figure 12 and figure 13 respectively. In figure 11, on the predicted class column, only one moth was wrongly predicted as FAW while the remaining predictions are correct even with low percentage accuracies.

Actual Class Images (Fall Army Worm)	Predicted Class	African Army Worm (AAW) %	Egyptian Cotton-Leaf Worm (ECLW) %	Fall Army Worm (FAW) %
	Fall Army Worm (FAW)	-6.56%	8.17%	10.12%
	Fall Army Worm (FAW)	0.61%	-1.61%	5.11%
	Fall Army Worm (FAW)	-4.33%	1.96%	5.91%
	Fall Army Worm (FAW)	-1.68%	0.33%	4.65%
	Egyptian Cotton Leaf Worm (ECLW)	-4.12%	5.86%	3.67%

Figure 11. Fall Army Worm (FAW) prediction by the MothNet model

In figure 12, on the predicted class column, only one moth was wrongly predicted as AAW while the remaining predictions are correct even with low percentage accuracies.

Actual Class Images (African Army Worm)	Predicted Class	African Army Worm (AAW) %	Egyptian Cotton-Leaf Worm (ECLW) %	Fall Army Worm (FAW) %
	African Army Worm (AAW)	8.69%	5.21%	-11.1%
	African Army Worm (AAW)	13.96%	14.65%	-17.33%
	African Army Worm (AAW)	7.89%	6.22%	-9.41%
	Fall Army Worm (FAW)	-3.98%	8.50%	9.32%
	African Army Worm (AAW)	7.02%	2.89%	-7.98%

Figure 12. Predicted AAWs with their accuracies

In figure 13, on the predicted class column, only two moths were wrongly predicted as ECLW while the remaining predictions are correct even with low percentage accuracies.

Actual Class Images (Egyptian Cotton-Leaf Worm)	Predicted Class	African Army Worm (AAW) %	Egyptian Cotton-Leaf Worm (ECLW) %	Fall Army Worm (FAW) %
	Egyptian Cotton Leaf Worm (ECLW)	6.41%	19.74%	-17.95%
	Egyptian Cotton Leaf Worm (ECLW)	-5.13%	5.32%	3.13%
	Fall Army Worm (FAW)	-2.39%	3.04%	3.59%
	Egyptian Cotton Leaf Worm (ECLW)	0.60%	3.27%	-0.11%
	African Army Worm (AAW)	3.92%	1.53%	-1.44%

Figure 13. Predicted ECLWs with their accuracies

4. TRANSFER LEARNING

The need for transfer learning cannot be overemphasized because of the small dataset of the maize moth images gathered. Supervised learning algorithm requires large dataset for perfect learning and prediction. The custom CNN (MothNet) model might not be perfect in its prediction due to the few (1674) images captured and used for the system training. Pre-trained models were modified and deployed for better predictions. The custom CNN model is compared with the pre-trained models (ResNet 50 & MobileNetV2) in order to select the most optimal and generalizable

model for the system. ResNet-50 and MobileNetV2 were chosen because they have been trained on ImageNet and possess high inference speed for computer vision tasks.

4.1. Residual Neural Network (ResNet 50)

ResNet 50 is one of the imageNet pre-trained models of CNN architecture [8], selected for image recognition which has been found excellent in doing such tasks. It has bottleneck architecture and contains 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 without fully- connected layer head. Since the captured image dataset of the maize species is small (the number of images is 1674), ResNet architecture is modified to suite the work as follows:

- i. Remove the fully connected nodes at the end of the network (that is, where the actual class label predictions are made).
- ii. Replace the fully connected nodes with freshly initiated ones (that is., using custom CNN defined FC layer of MothNet with 3 classes as represented in the dataset).
- iii. Freeze CONV layers in the network before the FC layers (ensuring that any previous robust features learned by the CNN are not destroyed).
- iv. Re-train only the added custom FC layer heads.
- v. Unfreeze all of the CONV layers in the network and perform a second pass of training. The final modified ResNet-50 architecture for maize species recognition is shown in figure 14.

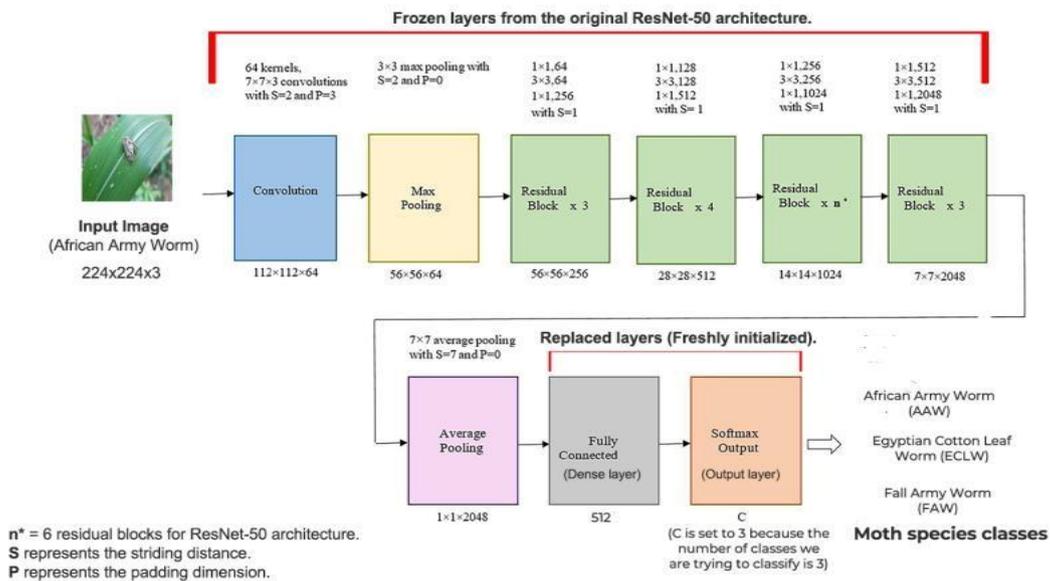


Figure 14. Modified ResNet 50 Architecture for Maize species Recognition [8]

With this development, the input image is now resized from $500 \times 500 \times 3$ to $224 \times 224 \times 3$ to be in the acceptable format that the modified pre-trained ResNet model expects.

Training & Validation Accuracy and Loss of ResNet Model: Looking at the graph shown in figure 15, the training accuracy achieved 82.5% and its validation accuracy 92.5% on 1 epoch. Both proceed to 98% after completing 20 epochs with no overfitting. Also, training losses achieved 45% and reduced to 2% and validation losses 27% and reduced to 2% after iterating for 20 epochs. This shows that the system did not overfit and has smooth learning with less error.

After training the modified ResNet-50 model, validation accuracy achieved 99.21% and loss of 3.79% and training accuracy of 99.75% with loss of 2.55% within 10mins 5 seconds.

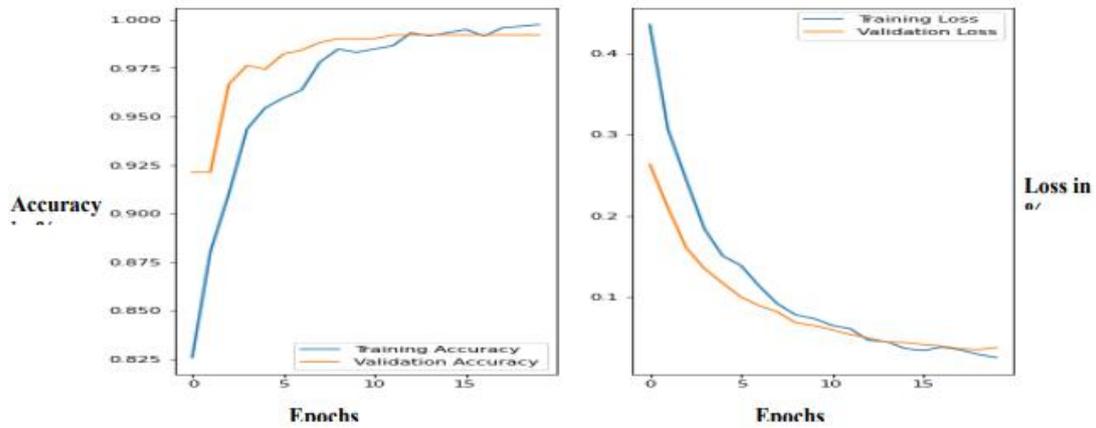


Figure 15. Training and Validation accuracy & losses on ResNet Model

4.2. Modified ResNet 50 Model Test Result

The same five images were picked from each category of the maize moths to test the predictive power of the modified ResNet 50. Each image of these species was passed to the model via online Google Colab and the model returned its prediction on each image. The MothNet model prediction accuracies on Fall Army Worm adults (FAWs), African Army Worm adults (AAWs) and Egyptian Cotton Leaf Worm adults (ECLWs) are showcased in figure 16, figure 17 and figure 18 respectively. In figure 16, all the FAW images were predicted correctly with percentage accuracies.

Actual Class Images (Fall Army Worm)	Predicted Class	African Army Worm (AAW) %	Egyptian Cotton-Leaf Worm (ECLW) %	Fall Army Worm (FAW) %
	Fall Army Worm (FAW)	0.12%	0.26%	99.61%
	Fall Army Worm (FAW)	0.25%	0.26%	99.48%
	Fall Army Worm (FAW)	0.01%	1.22%	98.76%
	Fall Army Worm (FAW)	0.002%	0.43%	99.55%
	Fall Army Worm (FAW)	0.004%	0.31%	99.68%

Figure 16. Correctly predicted FAWs and accuracies

In figure 17, all the AAWs images sent were correctly predicted with percentage accuracies.

Actual Class Images (African Army Worm)	Predicted Class	African Army Worm (AAW) %	Egyptian Cotton-Leaf Worm (ECLW) %	Fall Army Worm (FAW) %
	African Army Worm (AAW)	99.65%	0.34%	0.001%
	African Army Worm (AAW)	99.09%	0.90%	0.001%
	African Army Worm (AAW)	96.18%	3.80%	0.01%
	African Army Worm (AAW)	99.58%	0.40%	0.006%
	African Army Worm (AAW)	97.02%	2.94%	0.03%

Figure 17. Correctly predicted AAWs and accuracies

In figure 18, on predicted class column, only one image was wrongly predicted as FAW but others correctly predicted as ECLWs with percentage accuracies.

Actual Class Images (Egyptian Cotton-Leaf Worm)	Predicted Class	African Army Worm (AAW) %	Egyptian Cotton-Leaf Worm (ECLW) %	Fall Army Worm (FAW) %
	Egyptian Cotton Leaf Worm (ECLW)	0.3%	99.67%	0.001%
	Egyptian Cotton Leaf Worm (ECLW)	2.07%	97.91%	0.009%
	Fall Army Worm (FAW)	0.12%	16.14%	83.72%
	Egyptian Cotton Leaf Worm (ECLW)	7.79%	79.15%	13.04%
	Egyptian Cotton Leaf Worm (ECLW)	0.10%	95.49%	4.40%

Figure 18. Correctly and wrongly predicted ECLWs and accuracies

5. CONCLUSION

In this work, MothNet model was developed for the aggressive maize moths' prediction. The design of MothNet model was based on Convolutional Neural Network architecture with fine-tuned hyper parameters. Images of the targeted maize moths were captured via the use of delta pheromone traps set up in the maize farm. The captured images of three different targeted insects' categories were pre-processed and total of 1674 images stored in the image dataset folder and uploaded to online Google drive. Online Google Colaboratory platform (Python language based) was used to train the designed and customized CNN model called MothNet for the insects' classification. Due to the small size of the images captured and processed, data augmentation and dropout were carried out in order to get better prediction accuracy. The training and validation ration of 70:30 was used to avoid underfitting/overfitting and ensure minimal data loss. The trained MothNet model achieved accuracy of 91% with loss of 18.38% and when applied to real time testing, the predictions were not confident. The MothNet model predicted with low percentage accuracies on each image received which led to using transfer learning to improve on the prediction accuracies. ResNet 50 model was selected, modified and fine-tuned to give better prediction accuracies required. It achieved accuracy of 99% with loss of 4.35% and when tested, the confident was excellent. Its prediction gave high percentage accuracies to almost all the moths inputted. Hence, the developed MothNet model requires retraining with more images to do better prediction while modified ResNet 50 model is selected and recommended to be integrated in an embedded device for on-site proper detection and classification of these maize insect moths in the farm.

ACKNOWLEDGEMENTS

Special thanks to Zakka Usman, an associate professor and entomologist from crop and soil science, University of Port Harcourt, for initiating this research work and supervised till the end. We thank the funding body, PETROLEUM TECHNOLOGY DEVELOPMENT FUND SCHOLARSHIP BOARD, for their total support to ensure its success. Thanks to Steven Oladele, graduate from University of Port Harcourt who greatly assisted in the model development.

REFERENCES

- [1] Konstantinos G. L., Patrizia B., Dimitrios M., Simon P., & Dionysis B., (2018) "Machine Learning in Agriculture: A Review," *Sensors-MDPI*. [Online] Available: https://www.researchgate.net/publication/327029380_Machine_Learning_in_Agriculture_A_Review.
- [2] Technpstacks (2018), "Role of Machine Learning in Modern Age Agriculture," [Online] Available: <https://technostacks.com/blog/machine-learning-in-agriculture>
- [3] Konstantinos P. F.(2018) "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture* vol.145, pp. 311–318. 2018.
- [4] Simon H. C., Mayumbo N., Jackson P., Monde M. K., Philip O. Y. N., & Philemon H. S.(2019) "Machine Learning Algorithms for Automated Image Capture and Identification of Fall Armyworm (FAW) Moths," *Zambia Information Communication Technology (ICT) Journal*, vol.3, no. 1, pp. 1-4..
- [5] Belalm A. M., Ashqar B. S., & Abu-Nasser S. S. (2019) "Plant Seedlings Classification Using Deep Learning," *International Journal of Academic Information Systems Research (IJAIR)*, vol. 3, no.1, pp.7-14.
- [6] Kawcher A., Tasmia R. S., Syed Md I. A., & Sifat M.(2019) "Rice Leaf Disease Detection Using Machine Learning Techniques," *International Conference on Sustainable Technologies for Industry*, pp. 24-25.

- [7] Abhishek D., Debasmita B., & Kashi. N. D.(2016) “Automatic Detection of Whitefly Pest using Statistical Feature Extraction and Image Classification Methods,” International Research Journal of Engineering and Technology (IRJET), vol 3, no.9, pp. 950-959.
- [8] Thenmozhi K., & Srinivasulu Reddy U. (2019) “Crop pest classification based on deep convolutional neural network and transfer learning,” published in Elsevier-Computers and Electronics in Agriculture.

AUTHORS

Ezeofor, Chukwunazo Joseph is a PhD scholar at NnamdiAzikiwe University Awka (NAU) and a lecturer in the Department of Electrical/Electronic Engineering, University of Port Harcourt (UNIPORT) both in Nigeria. He obtained his B.Eng in Computer Engineering (ESUT) and M.Eng in Communication Engineering (FUTO). His doctorate field of study is in Computer and Control System Engineering and research interest in Artificial Intelligence, Robotics, Machine Learning, Automation and Control, and Neural Networks. He is a registered member of Council for the Regulation of Engineering in Nigeria (COREN) and Nigerian Society of Engineers (NSE). Ezeofor is happily married with four children.



Akpado, Kenneth Aghaegbunam is an Associate Professor in the Department of Electronic and Computer Engineering, NnamdiAzikiwe University Awka, Anambra State, Nigeria. He obtained his B.Eng in Electrical/Electronic Engineering (FUTO), M.Eng in Electronic and Engineering and Ph.D in Electronic and Computer Engineering (UNIZIK) both in Nigeria. Akpado specialties and interest is in Computer/Control Systems, Artificial Intelligence, Renewable Energy and Robotics. Computer Engineering (UNIZIK) with specialties in Computer and Control



Ulası Afamefuna Godfrey is a PhD student at Federal University of Technology Owerri, Imo State, Nigeria. (FUTO) He obtained his B.Eng in Computer Engineering (ESUT) and M.Tech in Information Management Technology (FUTO). His research interest is in Network security, Data mining, Machine learning etc. He is a registered member of Council for the Regulation of Engineering in Nigeria (COREN) and Nigerian Society of Engineers (NSE). Ulası is happily married with children.

