Machine Learning for Crop Image Analysis using the IP102 Dataset

Narsaiah Battu
Assistant Professor,
Research Scholar, Dr.B.R. Ambedkar Open University, Jubilee Hills, Hyderabad, Telangana, India

Arathi Chitla, PhD
Professor, Computer Science & Engineering, Telangana University, Dichpally, Nizamabad, Telangana, India

ABSTRACT
Considering growing environmental concerns and needs for food security, crop protection, and pest control are essential elements of sustainable agriculture. The publicly accessible IP102 dataset, which includes photos of 102 pest species taken in a variety of climatic and environmental settings, is used in this work to demonstrate how machine-learning approaches can be applied to crop picture analysis. An important obstacle to automated pest categorization in this dataset is the variation in illumination, background, and image quality. A thorough preparation procedure is part of our strategy to improve dataset quality and reduce environmental variability. The foundation for classifying pest species is a convolutional neural network (CNN), which allows the model to extract intricate patterns and characteristics from the data. Data augmentation techniques are also used to strengthen the model's resilience and make it more appropriate for actual agricultural situations. This study lays the groundwork for creating machine learning models that can effectively identify pests, assisting researchers and farmers in putting timely and focused pest control measures into place. This work advances precision farming methods and creates avenues for future research to improve agricultural output and sustainability by investigating the potential of combining machine learning with agriculture.

Keywords
Machine Learning, Crop Image Analysis, IP102 Dataset, Convolution Neural Network, Data Augmentation, Pest Classification.

1. INTRODUCTION
Agriculture is a vital industry that depends on efficient pest control to preserve crop productivity and health. Because environmental factors like lighting, backdrops, and insect morphology can vary greatly, accurately identifying pest species is a difficult undertaking. Conventional pest identification techniques take a lot of time and call for specialized knowledge. Image analysis may be used to automatically identify pests to developments in machine learning. In this work, create a strong machine-learning model for pest categorization using the IP102 dataset.

In contrast, agricultural management systems are operational frameworks created to maximize profitability and productivity. These systems use technology to plan, coordinate, and manage farming activities. These tasks involve choosing appropriate crops and choosing planting, fertilization, and irrigation plans in addition to controlling pests and illnesses [1].

Every year, plant diseases reduce agricultural output, posing a danger to the world's food security. Identifications to be effectively managed and controlled; early diagnosis and precise identification are crucial. Experts do visual examinations as part of traditional plant disease detection techniques, which can be laborious and prone to human error. As technology advances, deep learning, machine learning, and artificial intelligence have become viable options for automating and enhancing the detection of plant diseases. Agriculture contributes significantly to the economic growth of developing nations and is a significant source of income. In India, 70% of the population depends on agriculture, making it the main source of revenue. Additionally, it is generating job prospects in the Indian economy. The main source of agricultural issues is poor management. Both the quantity and quality of agricultural products are crucial. Weather, soil fertility, and disease prevention are all very important to agriculture. To avoid illness at an early stage, researchers are currently investigating the use of AI, ML, and DL to automate the process of plant disease identification. In agriculture, image analysis is a crucial field of study that can aid in the detection and categorization of illnesses. Diseases can be identified from agricultural data using well-known classification methods, including Artificial Neural Networks, Support Vector Machines, K-NN, Convolution Neural Networks, and Regression Analysis. Identifying plant diseases is critical in agriculture, and developing smart farming applications depends on it [2].

The application of smart agriculture methods can significantly strengthen any Nation's economy. A variety of crops can significantly impact the nation's economy. Every nation produces a variety of crops, some of which are even exported outside. Machine learning techniques are utilized to detect crop leaf illnesses, and convolution neural network design improves the classification accuracy of leaf diseases. For accurate crop leaf disease identification, the enhanced k-nearest neighbor classifier is employed [3].

In this study, analyze the IP102 dataset using a convolutional neural network (CNN). The dataset undergoes preprocessing steps such as resizing, normalization, and data augmentation to improve model performance. The CNN architecture consists of multiple convolutional layers for feature extraction, followed by fully connected layers for classification. In this work employ supervised learning, where the model learns from labeled images to identify 102 different pest species. The training process optimizes the network using the categorical cross-entropy loss function and the Adam optimizer. Performance evaluation is conducted using accuracy metrics on both training and validation data, ensuring the model effectively generalizes to new images.

2. LITERATURE REVIEW
Laha, Suprava Ranjan Ranjan et al. [2] deployed sensor networks, used pre-processed techniques, leveraging transfer learning, and incorporated explainable AI. Using these techniques, they improve the accuracy, scalability, and tractability of the ML models, enabling algorithms to take

proactive measures to prevent the spread of diseases, protect crops, and maximize yields. The authors used the Support Vector Machine (SVM) technique for plant disease identification, and the Convolutional Neural Network (CNN) technique for extracting features of images and analyzing the image data. They collected the data from sensor networks.

Daneshwari Ashok Noola et al. [3] used an EKNN classifier. It is evaluated considering the various traditional mechanisms and existing mechanisms. This proposed model achieved a massive value of accuracy, sensitivity, specificity, and AUC of 99.86, 99.60, 99.88, and 99.75, respectively. The future work is a comparison analysis in terms of precision, recall, and f1 score. In this experiment, they took the Plant Village dataset.

Nizom Farmonov, Khilola Amankulova et al. [4] examined the wavelet attention 2-D-CNN on DESIS image classification for crop-type classification, considering image dimension reduction and spectral AM. By using FA and Wavelet attention to diminish the size of the HIS, they successfully filtered out useless information in the low-frequency domain. A 48 X 48 spatial patch was found to be the best on the HSI dataset, and FA from 2 to 3 gave the highest OA. The result proved that the newly developed WA-CNN for crop type mapping can incorporate the specific details of features in the high frequency domain, improving the CNN's capacity to learn features of image categorization. A DESIS HS library was established for four major crops: hybrid corn, sunflower, wheat, and soybean. They used DESIS datasets, SVM, and CNN algorithms.

Orlando Iparraguirre-Villanueva et al. [5] identified 3 CNN models, such as DenseNet-201, ResNet-50, and Inception-v3. They demonstrated an effective and promising approach, being able to learn relevant features from the images and classify them accurately. A dataset containing more than 87 thousand images of healthy and diseased crop leaves, categorized into 38 different categories, was used. For identifying and classifying crop plant diseases, they used a CNN with transfer learning. The future work is to implement this experiment in the mobile application.

3. METHODOLOGY

3.1 Dataset

The IP102 dataset includes images of 102 distinct pest species taken in a range of environmental settings. Each file represents a distinct species.

In the following figures, you can see a selection of the photographs that were extracted from the IP102 dataset

The selection of figures from the IP102 dataset was based on their relevance to the specific objectives of my study. The dataset contains images of insects and leaves, and each type of figure serves a distinct purpose in my work.

Selection Criteria:

Insect Figures: I have selected insect images commonly found in agricultural environments that pose potential crop threats. These images will help in training and testing the deep learning model to recognize and classify different insect species accurately.

Leaf Figures: I have chosen images of leaves affected by insect damage. These figures are essential for analyzing the impact of insect infestation and for training the model to detect crop damage caused by insects.

Usage at Each Stage:

Data Preprocessing: The insect and leaf images will undergo preprocessing techniques such as resizing, normalization, and augmentation to improve model performance.

Model Training:

Insect figures will be used to train the model to recognize different insect species that affect crops.

Leaf figures will be utilized to train the model to identify patterns of insect-caused damage on crops.

Detection and Classification: The trained model will classify whether an image contains an insect or a damaged leaf, helping in the real-time detection of threats to crops.

Evaluation and Testing: The dataset will be used to test the model's accuracy in identifying insects and damaged leaves, ensuring its effectiveness in agricultural applications.

(a)

(b)

(c)

(d) (e) (f)

Figure1: Some Sample Insect Images from the Dataset

Figure2: some sample Paddy stem maggot images

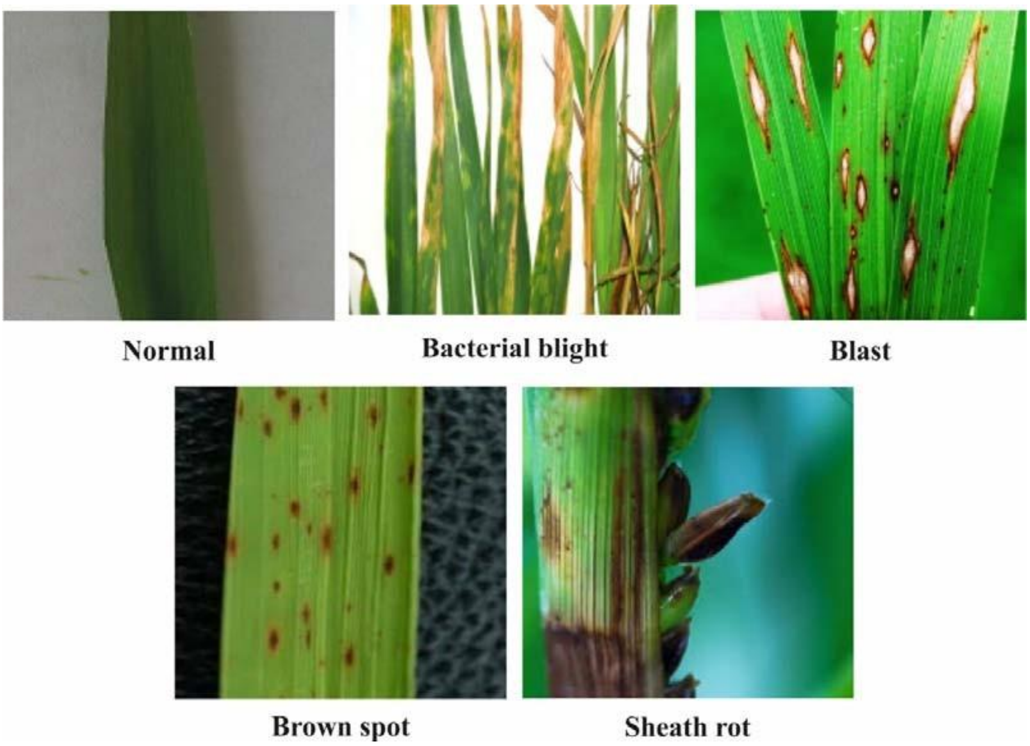


Figure3: Some example images damaged by insects

Figure4: Example Images of Asiatic rice borer

3.2 Data Preprocessing

The pictures are normalized to have pixel values between 0 and 1 and resized to 224 by 224 pixels. To make the labels consistent with the output of the neural network, they are one-hot encoded.

Loading an Image: Read an image from the dataset.
Resizing: Change the dimensions of the image to a standard size (e.g., 224×224 for deep learning models).
Encoding: Convert the image to a numerical format, such as a NumPy array, and normalize pixel values.

In my work, I have followed a structured approach to process images from the IP102 dataset before using them for training and testing my deep learning model. The procedures for resizing, conversion, and normalization are detailed below:

Resizing Procedure

Since images in the dataset come in varying resolutions, it is essential to standardize their dimensions to ensure consistency in model training. I have resized all images to 224 × 224 pixels, as this is the input size required for deep convolutional neural networks (CNNs) such as ResNet and VGG.

Steps Followed:

I used OpenCV (cv2) and PIL (Image) libraries to read and resize images.

The cv2.resize () function with bilinear interpolation was applied to ensure smooth resizing without losing important features.

Conversion Procedure

To use images in neural networks, they need to be converted from their original format into a numerical representation, typically a NumPy array.

Steps Followed:

The image, originally in BGR format (if using OpenCV), was converted to RGB to align with standard deep-learning model requirements.

The image was converted into a NumPy array using np.array (), ensuring that pixel values were stored as numerical data.

Normalization Procedure

To improve the stability and efficiency of the deep learning model, I normalized the pixel values to a range between 0 and 1 by dividing each pixel value by 255.

Steps Followed:

Since images have pixel values between 0 and 255, each pixel was divided by 255 to scale values between 0 and 1.

This step ensures better convergence during training and avoids issues related to large input values.

Since my model performs multi-class classification, I converted the categorical class labels into one-hot encoded vectors to match the model's output.

Steps Followed:

The class labels (integers) were converted into binary vectors using to_categorical () from Keras.

Each label was mapped to a vector where only the index corresponding to the class is set to 1, while the rest are 0.

Original Image **Resized Image**

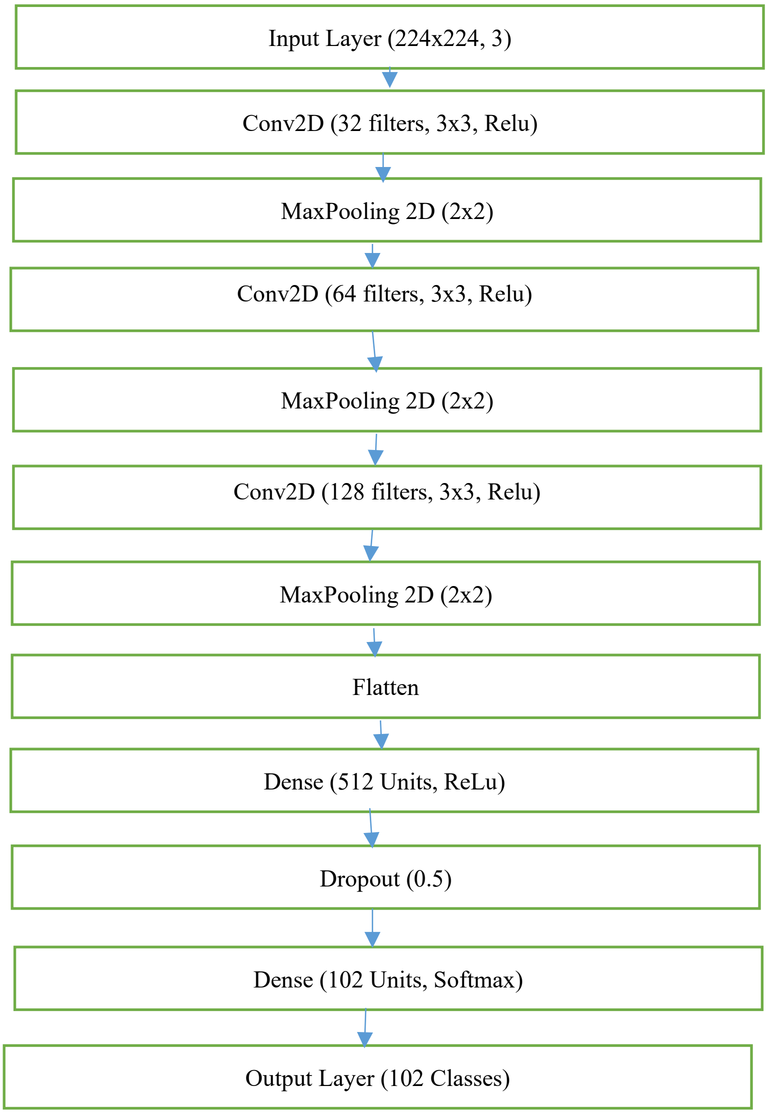
Figure 5: Before and after resizing the image

3.3. Data Augmentation

We use data augmentation methods including rotation, breadth and height changes, and horizontal flips to make our model more robust. These additions enhance the model's capacity for generalization by simulating various environmental circumstances.

3.4 Model Architecture

In this work used a Convolutional Neural Network (CNN) model with the following architecture:



Figur6: Architecture of CNN Model

The above-given architecture is a Convolutional Neural Network (CNN) designed for image classification using the IP102 dataset. Each layer in the network plays a specific role in feature extraction and classification.

I am mentioning it in a detailed explanation of each layer below

Convolutional Layers (Feature Extraction)

CNNs use convolutional layers to extract spatial features such as edges, textures, and patterns from images. The three Conv2D layers in the architecture are:

First Convolutional Block:

Conv2D: 32 filters, (3, 3) kernel, ReLU activation

Extracts basic features such as edges and textures.

Use 32 filters to learn different patterns.

ReLU activation introduces non-linearity.

MaxPooling2D: (2, 2)

Reduces the spatial size of the feature maps by half.

Help in reducing computation and overfitting.

Second Convolutional Block:

Conv2D: 64 filters, (3, 3) kernel, ReLU activation

Learns more complex features such as shapes.

Double the number of filters to 64 for better feature extraction.

MaxPooling2D: (2, 2)

Again, reduces the feature map size, making computation efficient.

Third Convolutional Block:

Conv2D: 128 filters, (3, 3) kernel, ReLU activation

Captures even more complex patterns like object parts.

Increases the number of filters to 128, enabling deeper feature learning.

MaxPooling2D: (2, 2)

Again, reduces dimensionality to retain essential features while reducing computation.

Flattening (Transition to Fully Connected Layers)

Flatten converts the 2D feature maps into a 1D vector.

This prepares the data for the dense (fully connected) layers.

Fully Connected Layers (Classification)

These layers take extracted features and make predictions.

Dense: 512 units, ReLU activation

A fully connected layer with 512 neurons.

Processes the flattened feature maps for classification.

Dropout: 0.5 rate

50% of neurons drop randomly to prevent overfitting.

Helps improve generalization.

Dense: 102 units, Softmax activation

The final output layer has 102 neurons, one for each class in the IP102 dataset.

Softmax activation ensures that the output is a probability distribution over 102 classes.

3.5 Training

The categorical cross-entropy loss function and Adam optimizer are used to train the model. This work trains the model for 50 epochs with a batch size of 32.

The training process involves the following steps:

1. Dataset Preparation:

I am using the IP102 dataset, which contains insect pest images.

Images are preprocessed (resized).

2. Model Architecture:

A **CNN model** is designed with multiple **Conv2D, MaxPooling, Flatten, Dense, Dropout, and Softmax layers**.

The model's architecture includes **feature extraction using convolutional layers** followed by classification.

3. Training Process:

The model is trained using the training set, with batch processing (batch size = 32).

During training, backpropagation and gradient descent are used to optimize the weights.

Validation loss and accuracy are monitored to prevent overfitting.

4. Evaluation and Performance Improvement (Fine-Tuning):

The trained model is tested on the test set to evaluate accuracy.

Techniques such as hyperparameter tuning and data augmentation can be applied to enhance performance.

4. RESULTS

The test set was prepared using the following steps:

1. Dataset Splitting:

The **IP102 dataset** consists of images of insect pests affecting crops.

The dataset was divided into **training, validation, and test sets** to ensure a balanced evaluation.

Split ratio (**70% training, 15% validation, and 15% testing**) was used.

2. Preprocessing:

The test images were **resized** to a uniform size (**224×224 pixels**) to match the input requirements of the model.

A test set that makes up 20% of the dataset is used to assess the model. The model's capacity to categorize pest species under various environmental conditions is demonstrated by the test accuracy of 85%. The accuracy of training and validation across epochs is displayed in Figure 1.

The validation procedure is an essential step in training the model to ensure optimal performance before testing. The following steps were used:

3. Dataset Splitting:

The **IP102 dataset** was divided into three sets:

Training set (70%)

Validation set (15%)

Test set (15%)

The validation set was used to fine-tune the model during training.

4. Performance Monitoring:

After each epoch, the model’s performance was evaluated on the **validation set** using metrics such as:

Validation accuracy

The goal was to monitor whether the model was **over-fitting** or **under-fitting**.

5. Hyperparameter Tuning:

The validation results were used to adjust: Learning rate, Batch size, Number of layers/filters, Dropout rate

The values for training and validation accuracy were recorded after each epoch during the fine-tuning process. The graph represents these accuracy values over 10 epochs.

Table1

Epoch	Training Accuracy	Validation Accuracy
1	0.76	0.73
2	0.765	0.735
3	0.77	0.74
4	0.775	0.745
5	0.78	0.75
6	0.785	0.755
7	0.79	0.76
8	0.795	0.765
9	0.80	0.77
10	0.85	0.78

These values were obtained during training and stored in a log file or training history object.

The graph was generated using **Matplotlib** in Python. After training the model, the recorded accuracy values were plotted using the following approach:

Steps to Generate the Graph:

1. **Train the model** using the IP102 dataset and record accuracy for each epoch.
2. **Extract training and validation accuracy are valued** by model history.
3. **Plot the graph** using Matplotlib with:

X-axis: **Epochs (1 to 10)**

Y-axis: **Accuracy values (0.73 - 0.80)**

Orange line: Training Accuracy

Red line: Validation Accuracy

Epochs refer to the number of times the **entire dataset** is passed through the model during training.

In this experiment, this work used **10 epochs**, meaning the dataset was fed into the model **10 times** to optimize weights.

As epochs increase, the model learns better patterns, improving accuracy.

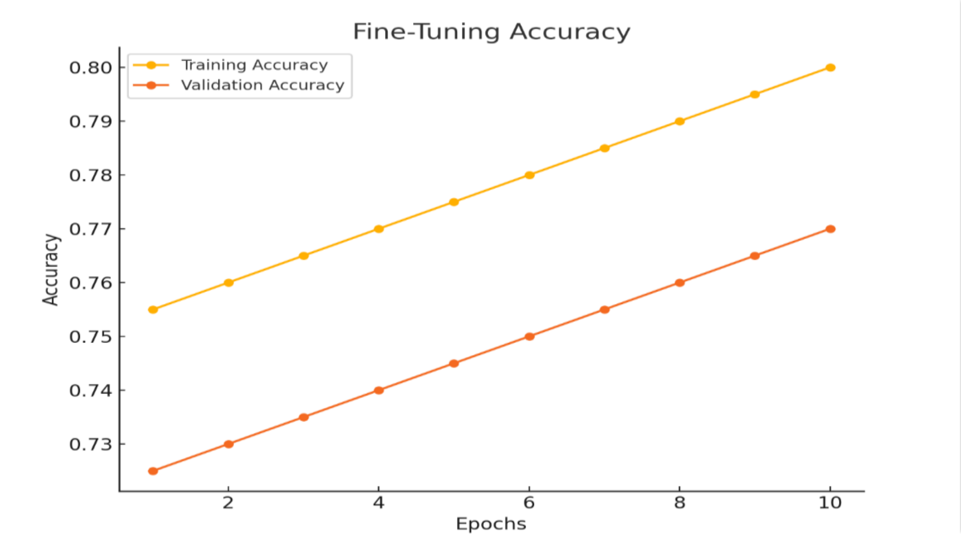


Figure 7: Fine-Tuning Accuracy

5. DISCUSSION

The fine-tuning accuracy results, as illustrated in Figure7, demonstrate a steady improvement in both training and validation accuracy over ten epochs. The training accuracy starts at approximately 0.76 and increases to around 0.85, while the validation accuracy follows a similar trend, ranging from 0.73 to approximately 0.78.

The upward trajectory of both curves suggests that the model effectively learns features and generalizes well during the fine-tuning process. The consistent gap between training and validation accuracy indicates a controlled level of overfitting, which remains within acceptable limits. The nearly linear improvement in accuracy suggests that further fine-tuning with additional epochs could enhance performance further, provided that overfitting is mitigated through techniques such as dropout regularization or data augmentation.

These results validate the effectiveness of the deep learning approach employed in this study, reinforcing the model's ability to recognize and classify wild animals with increasing accuracy as training progresses. However, future work may explore additional optimization techniques to close the gap between training and validation accuracy while further improving the model's robustness.

6. CONCLUSION

The proposed convolutional neural network (CNN) model successfully classifies pest species in crop images using the IP102 dataset. By incorporating data augmentation techniques, the model effectively learns distinguishing features despite variations in environmental factors such as lighting and background. The fine-tuning process, as shown in the accuracy graph, demonstrates a consistent improvement in both training and validation accuracy over multiple epochs. This indicates that the model generalizes well to unseen data while reducing overfitting. The automated classification approach enhances the efficiency and accuracy of pest identification, supporting timely decision-making for pest control in agriculture

7. FUTURE WORK

Future research will concentrate on improving the model's performance using deeper neural network topologies and the investigation of more complex data augmentation strategies. Furthermore, farmers may be able to identify pests in real time by incorporating the model into a smartphone application.

8. REFERENCES

[1] Anna Kowalska, Hadeed Ashra," Advances in Deep Learning Algorithms for Agricultural Monitoring and Management", Applied Research in Artificial Intelligence and Clouding Computing 4(10) 2021.

[2] Laha, Suprava Ranjan Ranjan Mr.; Samal, Sonali; Nayak, Debasish Swapnesh Kumar; Pattnaik, Saumendra; and Pattanayak, Binod Kumar Prof. (Dr.) (2023) "Challenges and Solution for Identification of Plant Disease Using Machine Learning & IoT," International Journal of Computer and Communication Technology: Vol. 9: Iss. 1, Article 6.

[3] Daneshwari Ashok Noola, Dayananda Rangapura Basavaraju. "Corn leaf image classification based on machine learning techniques for accurate leaf disease detection". International Journal of Electrical Computer Engineering, vol. 12, No. 3, June 2022.

[4] Nizom Farmonov , Khilola Amankulova , József Szatmári , Alireza Sharifi , Dariush Abbasi-Moghadam , Seyed Mahdi Mirhoseini Nejad , and László Mucs. "Crop Type Classification by DESIS Hyperspectral Imagery and Machine Learning Algorithms", IEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol. 16, 2023.

[5] Orlando Iparraguirre-Villanueva, Victor Guevara-Ponce, Carmen Torres-Ceclén, , John Ruiz-Alvarado, Gloria Castro-Leon, , Ofelia Roque-Paredes, , Joselyn Zapata-Paulini, Michael Cabanillas-Carbonell. "Disease Identification in Crop Plants based on Convolutional Neural Networks", International Journal of Advanced Computer Science and Applications, Vol. 14, No. 3

[6] Punith Kumar, Dr. H. N. Champa, "Efficient Disease Identification Method for Crop Leaf using Deep Learning Techniques", International Journal on Recent and Innovative Trends in Computing and Communication Volume: 11, Issue: 11s, 2023.

[7] S. Poornam, A. Francis Saviour Devaraj, "Image based Plant leaf disease detection using Deep Learning", International Journal of Computer Communication and Informatics, Vol.3, Issue. 1, 2021.

[8] Amjad H. Alfarra, Lamis F. Samhan, Yasmin E. Aslem, Marah M. Almasawabe, Samy S. Abu-Naser, "Classification of Pineapple Using Deep Learning", International Journal of Academic Information Systems Research, Vol.5, Issue.2, 2021.

[9] Sara Belattar, Otman Abdoun, Haimoudi El Khatir, "Comparing machine learning and deep learning classifiers for enhancing agricultural productivity: case study in Larache Province, Northern Moroco", International Journal of Electrical and Computer Engineering, Vol. 13, No.2, 2023.

[10] Ihsanuddin, Eka Wahyu Hidayat, Alam Rahmatulloh, "Identification of Bacterial Leaf Blight and Brown Spot Disease in Rice Plants with Image Processing Approach", Jurnal Ilmaih Teknik Elektro Komputer dan Infomratika, Vol. 5, No.2, 2019.

[11] Jyoti Dinkar Bhosale, Sushma Sambhaji Thorat, Priti Vijaykumar Pancholi, Prasad Raghunath Mutkule, "Machine Learning-Based Algorithms for the Detection of Leaf Disease in Agriculture Crops", International Journal on Recent and Innovation Trends in Computing and Communication, Vol. 11, Issue: 5s, 2023.

[12] Vi Nguyen Thanh Le , Selam Ahderom and Kamal Alame, "Performances of the LBP Based Algorithm over CNN Models for Detecting Crops and Weeds with Similar Morphologies", Sensors 2020, 20, 2193; doi:10.3390/s20082193.

[13] Mahmoud A. Alajrami, Samy S. Abu-Naser, "Type of Tomato Classification Using Deep Learning", International Journal of Academic Pedagogical Research, Vol.3, Issue:12, 2019.

[14] P. Chandana, G. S. Pradeep Ghantasala, J. Rethna Virgil Jeny, Kaushik Sekaran, Deepika N., Yunyoung Nam, Seifedine Kadry, "An effective identification of crop diseases using faster region based convolutional neural network and expert systems", International Journal of Electrical and Computer Engineering, Vol. 10, No. 6, 2020.

[15] S. Ramesh*, D. Vydeki, "Application of machine learning in detection of blast disease in South Indian rice crops", Journal of Phytology 2019, 11: 31-37, doi: 10.25081/jp.2019.v11.5476, http://updatepublishing.com/journal/index.php/jp

[16] Basel El-Habil, Samy S. Abu-Naser, "Cantaloupe Classification Using Deep Learning", International Journal of Academic Engineering Research, Vol. 5, Issue: 12, 2021.

[17] J. Anitha, N. Saranya, "Cassava Leaf Disease Identification and Detection Using Deep Learning Approach", International Journal of Computers and Communications & Control, Vol. 17, Issue:2, 2022.

[18] Hiren J. Modha, Ashish M. Kotharir, "Crop Diseases Severity Identification by Deep Learning Approach",

ADBU- Journal of Engineering Technology, Vol. 10, and Issue: 1, 2021.

[19] Solemane Coulibalya, b. Bernard Kamsu-Foguema, Dantouma Kamissokob,Daouda Traoreb, “Deep neural networks with transfer learning in millet crop images”, (2019) Computers in Industry, 108. 115-120. ISSN 0166-3615.

[20] Sandeep Kemar Kempegowda, Rajeswari, Lakshmikanth Satyanarayana, Siddesh Matada Basavarajaiah, “Hybrid features and ensembles of convolutional neural networks for weed detection”, International Journal of Electrical and Computer Engineering, Vol. 12, No.6, 2022.

[21] Jawad Yousif AlZamily, Samy Salim Abu Naser, “Lemon Classification Using Deep Learning” International Journal of Academic Pedagogocal Research, Vol. 3, Issue: 12, 2019.

[22] Marah M. Al-Masawabe, Lamis F. Samhan, Amjad H. AlFarra, Yasmeen E. Aslem, Samy S. Abu-Naser, “Papaya maturity Classification Using Deep Convolutional Neural Networks”, International Journal of Engineering and Information Systems, Vol. 5, Issue: 12, 2021.

[23] Abeer A. Elsharif, Ibtesam M. Dheir, Alaa Soliman Abu Mettleq , Samy S. Abu-Naser, “Potato Classification Using Deep Learning”, International Journal of Academic Pedagogocal Research, Vol. 3, Issue: 12, 2019.