Crop disease detection using deep learning

Kshitiz Kumar, Aman Raj, Aryan Singh,
Snahil Shubhra
School of Computer Engineering
KIIT University
Bhubaneswar, Odisha

Niranjan Kumar Ray
niranjan.rayfcs@kiit.ac.in
School of Computer Engineering
KIIT University
Bhubaneswar, Odisha

***Abstract*—The fields of computer vision and image processing have been completely transformed by Convolutional Neural Networks (CNN). They are perfect for tasks like picture classification, object detection, and facial recognition since these deep learning architectures are made with the intent to extract complex patterns and features from visual input. In India, among its diverse agricultural assets, sugarcane stands out as the most hopeful crop. Nevertheless, farmers encounter numerous challenges when cultivating sugarcane, such as crop diseases and leaf issues. Deep learning presents an intriguing solution to address these concerns. This research project is focused on implementing a deep learning model to accurately identify and detect diseases affecting sugarcane leaves using smart agriculture techniques.**

***Index Terms*—Smart agriculture, Image classification, Deep learning, Convolutional Neural Networks (CNN), Tensorflow**

I. INTRODUCTION

Problem statement : In modern agriculture, the need for efficient disease detection and management is critical to ensure food security and sustainable farming practices. Conventional methods often fall short in providing timely and accurate diagnoses, leading to crop losses and excessive resource use. Deep learning, a subset of artificial intelligence, has emerged as a promising solution to this problem. However, the widespread adoption of deep learning for crop disease detection faces challenges related to model deployment, data collection, and integration with existing agricultural systems. This study aims to address these challenges and demonstrates how smart agriculture, empowered by deep learning technology, can revolutionize disease detection, resource optimization, and decision-making in farming, ultimately contributing to a more resilient and productive agricultural sector.

Convolutional Neural Networks (CNNs) have made a significant advancement in the quickly developing field of deep learning, significantly altering how computers interpret and process visual data. Experts traditionally rely on visual observation with the naked eye to identify and detect sugarcane leaf diseases, which necessitates continuous monitoring for confirmation and validation of diagnoses. However, this approach is impractical for large farms, especially in developing countries like the Philippines, where seeking expert assistance is expensive and labor-intensive. Machine learning, a popular technology in some studies, is used for plant disease classification and detection. Conventional machine learning techniques like SIFT and SVM are often employed, but they involve complex and resource-intensive calculations, making them unsuitable for online applications and limiting their performance.

To enhance performance and accuracy in feature extraction, advanced tools such as electromagnetic radiation, IR spectrums, and plant genomics are required, but these options are prohibitively expensive for small-scale farmers seeking to extract disease-related features. Deep learning, on the other hand, utilizes artificial neural network architectures with many layers, as opposed to traditional neural networks. It has revolutionized various fields, including image detection, image classification, and acoustics, which demand substantial data processing.

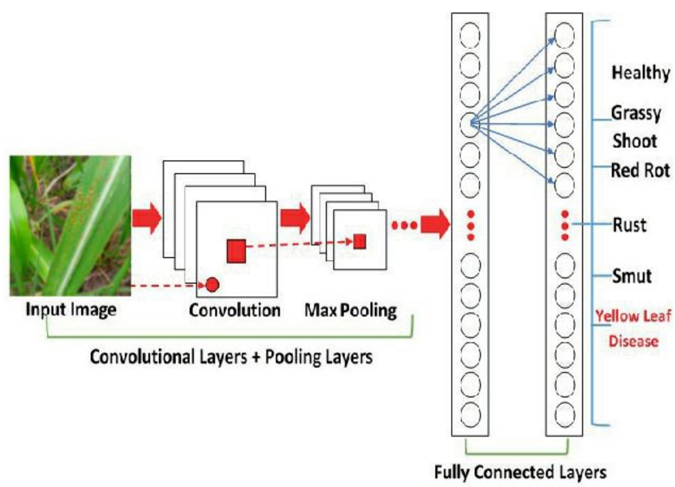

Fig. 1 Convolutional Neural Network Architecture

The architecture of CNNs is inspired by the human visual system, which is adept at recognizing patterns and objects in visual data. CNNs consist of multiple layers, each with a specific function:

- Input Layer: The input layer represents the raw data, such as an image. Each neuron in this layer corresponds to a pixel in the input image.

- Convolutional Layers: Convolutional layers are the fundamentals of CNNs. They apply convolution operations to extract features from the input data. Convolution involves sliding a small filter over the input data and computing dot products. These filters are learned during the training process and capture various patterns, such as edges, textures, and shapes.

- Activation Layers: After convolution, an activation function is applied element-wise to introduce non-linearity.

- Pooling Layers: Pooling layers reduce the spatial dimensions of the data, making it computationally efficient. Common pooling operations include max-pooling, which retains the maximum value in a region, and average pooling, which computes the average.

- Fully Connected Layers: Fully connected layers connect all neurons from the previous layer to the current layer. These layers learn high-level representations and are typically used in the final stages of the network.

- Output Layer: The output layer provides the final predictions or classifications. The activation function of this layer depends on the specific task, such as softmax for multiclass classification or sigmoid for binary classification.

II. Literature survey

With applications ranging from facial recognition and object identification to medical image analysis and autonomous cars, picture classification is a key topic in computer vision. The capacity of convolutional neural networks (CNNs) to automatically acquire hierarchical features from raw pixel input has led to their emergence as the dominant architecture for image classification tasks. In this detailed analysis of the literature, we will examine the main publications and turning points in the evolution of CNNs for image classification, demonstrating how these developments have influenced the discipline throughout time. Researchers from all over the world are looking at many elements to address problems with the older models, leading to many advances in the earlier models for CNN and its use in generative AI.

Vishali Wadhe et al. "Sugarcane Disease Detection using Deep Learning" [1] ,emphasizes on the importance of disease prediction in respect to sugar cane. The study uses Mobile Net v2 model, a deep learning model, to predict the infestation. Additionally, the paper considers the changing weather to give the model a more reliable and accurate touch.

Sammy V. et al. "Sugarcane Disease Recognition using Deep Learning" [2], uses the traditional deep learning algorithm for image detection. The model was trained using 13,842 images that comprised healthy and infested leaves. The model turned out to be a success as the model featured an respectable accuracy of 95%. The aim of the study was to help farmers identify the diseases using technology. However, the study fails to consider various real life aspects such as temperature and humidity.

Hyeon Park et al. "Image-based disease diagnosing and predicting of the crops through the deep learning mechanism" [3],is a brief study that employees deep learning image classification to predict the infestation in crops. The key aspect of the paper is identification of the disease and the subsequent action.

N.K. Hemalatha et al. "Sugarcane leaf disease detection through deep learning" [4], is a through study based on the detection of sugarcane disease. The study proposes an efficient deep learning neural architecture to detect the presence of the following five types of sugercane disease- rust spots, yellow leaf disease, Helmanthospura spot, Cercospora leaf spot, and red rot. The study uses a convolution neural network based image classifier, which uses 3000 images to train the model. The trained model is tested with 1000 test images and the accuracy of 96%. The study goes a step further to design a mobile phone application so that the farmer can easily, using the mobile's camera, capture the image and get the results after the backend processing.

Alexander A. Hernandez et al. "Classification of Sugarcane Leaf Disease using Deep Learning Algorithms" [5], uses variety of deep learning technologies to showcase the effectiveness of deep learning models over classifiers as far as the image prediction is concerned. The model uses 16800 images for testing, 4800 for validation and 2400 for testing. In the end, inception v4, a deep learning algorithm is found to outperform various classifiers with a outstanding accuracy of 99.61%.

Juncheng Ma et al. "A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network" [6], is a neural based detection method for four types of cucumber crops mainly, anthracnose, downy mildew, powdery mildew, and target leaf spots. The model uses data argumentation to address over fitting and 14208 were trained for the model, resulting an accuracy of 93.4%. Finally the study compares the deep learning learning model to machine learning models such as random forest and support vector machine(SVM).

TABLE I
SUMMARY TABLE.

Study	Summary
Vishali Wadhe et al. "Sugarcane Disease Detection using Deep Learning"	Used CNN based transfer learning approach to quickly and accurately upto 90% accuracy.
Sammy V. et al. "Sugarcane Disease Recognition using Deep Learning"	The highest recorded validation accuracy during the training was 95% with 60 epochs.
Hyeon Park et al. "Image-based disease diagnosing"	Built a two conv and 3 fully connected network and got an accuracy of 89.7%
N.K. Hemalatha et al. "Sugarcane leaf disease detection through deep learning"	Used CNN trained as an image classifier with around 3000 leaf images. The model is tested for about 1000 images. The proposed model has achieved 96% accuracy.
Alexander A. Hernandez et al. "Classification of Sugarcane Leaf Disease using Deep Learning Algorithms"	Evaluated residual nets with a depth of up to 152 layers—8× deeper than VGG nets, obtained a 28% relative improvement on the COCO object detection dataset.
Juncheng Ma et al. "A recognition method for cucumber diseases"	The DCNN's accuracy on the imbalanced and balanced datasets was 93.4% and 92.2%, respectively. AlexNet beat the DCNN in comparison testing because to its comprehensive feature display.

1

III. PROPOSED METHODOLOGY

A. DenseNet CNN Architecture

DenseNet [7] is a unique CNN design that placed an emphasis on dense connectivity between layers. In conventional CNNs, feature maps were processed by earlier layers before being concatenated or summed. By physically connecting each layer to every next layer, DenseNet used a new strategy. Due to the network's extensive connection, feature reuse was encouraged and gradient flow was enhanced. Additionally, it decreased the amount of parameters, which improved network efficiency while preserving high performance. On a number of image classification datasets, DenseNet produced cutting-edge results. The groundbreaking design of DenseNet demonstrated the benefits of dense connections, paving the way for more effective and parameter-efficient CNN systems. It underlined the significance of increasing feature reuse and information flow between layers. This study uses DenseNet 201 for the research.

B. Libraries and Modules

Tensorflow is one of the most popular deep learning libraries worldwide, it is. A versatile and effective platform for creating, honing, and deploying deep learning and machine learning models is offered by TensorFlow. CNNs, RNNs, reinforcement learning, and other machine learning models are supported by TensorFlow. High-level APIs like Keras are included in TensorFlow, which makes it easier to create and train neural networks. Models can be deployed seamlessly on various areas, including mobile devices and the cloud. The Python-based, open-source Keras deep learning package offers a high-level neural networks API. It is intended to be expandable, modular, and user-friendly.

C. IOT Implementation

IoT-based sugarcane plant disease detection with deep learning integrates gathering images through sensor and analysis to monitor environmental conditions and detect diseases in sugarcane crops, enabling timely intervention and enhancing agricultural yield and sustainability.

D. Dataset [8]

This refers to a dataset of images that contains 100 images of healthy sugarcane leaves, 100 images of red rot disease infected leaves and 100 images of bacterial blight disease infected leaves, totaling 300 images. The dataset is designed for sugarcane plant disease classification, where the goal is to build a ML model that can accurately classify an input image as either healthy or infected and detect the infected disease type. The dataset has been split into a training set and a test set in a 80:20 ratio. The purpose of this split is to train the model on a large portion of the data and then assess its accuracy and generalization performance on a separate, unseen portion of the data.

E. Dataset Cleaning and Preprocessing

Data pre-processing and cleaning is the the fundamental procedure done on the raw data set to remove inconsistencies, anomalies and inessential data. The data-preprocessing helps to make the raw data set in the machine readable form before the data is fed into the algorithm. The major operations done in the data pre-processing and cleaning stage are:

1) Data Collection: For our CNN classification, we gather a well defined data set from Kaggle that is diverse and and contains the class.

2) Data inspection: Next, we manually examine the dataset thoroughly. Any inessential, duplicate or mislabeled

data is noted and is taken care of.

3) Data splitting: The data is then split. The split is 80% for the training data and 20% for the test data.

4) Resizing and Normalization : The images are all set to an univocal, consistent size fo the CNN model.The most common choices include 224x224 pixels. Then the pixel values are normalized or calibrated to range [0-1] or [-1 to 1].

5) Data augmentation: This is used to increase the training set by duplicate sampling of the images. Its one of the techniques to avoid overfitting.

F. Feature Selection

Through a sequence of convolutional and pooling layers, CNNs automatically extract features from the original pixel values of images. Beginning with straightforward edge detectors and progressing to increasingly intricate patterns and structures, these acquired features are hierarchical and progressively abstract. However, there are various methods and things to think about while choosing features and engineering CNNs. Feature selection can be viewed as a type of regularisation approach, such as dropout or L2 regularisation. They assist the network in narrowing its attention to the most crucial traits while lessening reliance on distracting or unimportant ones.

G. Training Scenerios

This is the final step of modelling. In this we use various techniques to train the model. After the training, we check the accuracy of the model using a test case. If the accuracy is not satisfactory, we retrain the model. The following are the techniques used:

a) Design the architecture:We configure the basic architecture of the CNN such as convolution layer, pooling size. Etc Moreover, we decide on the kernel size, filters and the other hyper parameters.
b) Initiation of the model: We initiate the training with a relative predefined weightsusing a sequential keras model.

c) Forward Propagation: A batch of the dataset is fed through the model. We use convolutions, pooling, activation functions, and connected layers to find the accuracy. The deviation from the actual accuracy is noted.

d) Back Propagation: We calculate the gradient loss with respect to the weights and biases in the CNN network. Then the weights are updated using optimization techniques like gradient decent. This is repeated with various training data.

e) Evaluation : The model is fed with test data and the accuracy is noted. If the model does not achieve a good accuracy, retraining is done through fine tuning and optimization.

H. Model Design & Working

a) First a base model using the DenseNet201 architecture is created without the top classification layer. It's pre-trained on ImageNet data, meaning it has learned to recognize a wide range of features from images. The model's output is averaged ("pooling='avg'") to create feature vectors for input images of size 224x224 pixels with 3 color channels (RGB).

b) Then there comes a fully connected layer with 256 neurons and ReLU activation function. A dropout layer with a dropout rate of 0.2 is applied to the output of the previous layer. Dropout helps prevent overfitting by randomly deactivating a fraction of neurons during training.

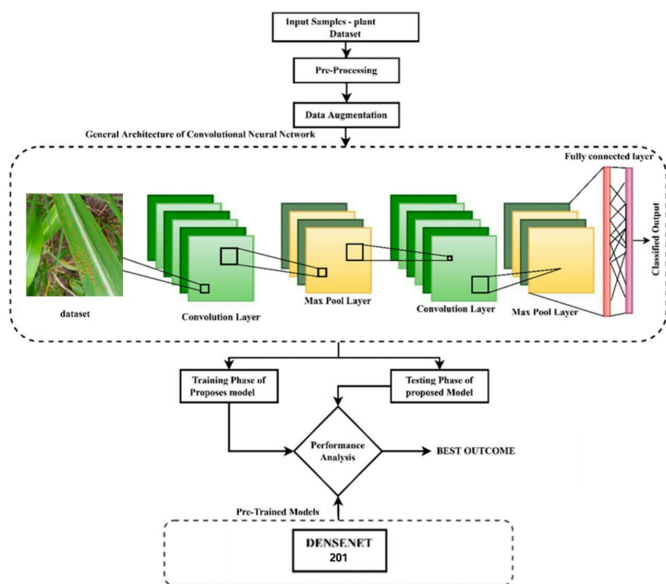

Fig. 1. Model working

c) Then the output layer with 3 neurons, representing the three classes, using a softmax activation function. It also includes L2 regularization (kernel regularizer) with a strength of 0.01 to help prevent overfitting.

d) Following Batch Normalization, a MaxPooling layer with a 2x2 pool size and a stride of 2 is added. This reduces the spatial dimensions of the feature maps by a factor of 2, reducing computation and promoting translation invariance.

e) The model is compiled with the Adam optimizer and the squared hinge loss function. Accuracy is chosen as the evaluation metric.

f) The model is trained with a batch size of 32 and for 30 epochs. It uses data augmentation to generate augmented training samples, and it tracks training and validation accuracy and loss over the specified number of epochs. The training process is based on the specified batch size and number of steps per epoch.

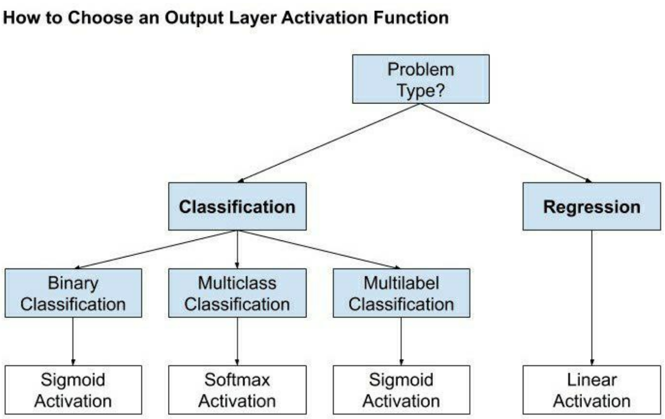

Fig. 2. Process to select Activation function

g) The output layer has 3 neurons with a softmax activation function. This is suitable for multiclass classification tasks, where the network will output a value between the available classes 0-2, representing the probability of the input image belonging to one of the three classes 0 or 1 or 2.

I. Result Analysis

Figure 4: Comparing the training loss and validation loss is a crucial aspect of monitoring the model's performance during training. These two metrics provide valuable insights into how well the model is learning and whether it is overfitting or underfitting. A minimal overfitting was observed.

Figure 5: Comparing the training accuracy and validation accuracy is a common practice to assess the performance and behavior of your model during the training process. These two metrics provide valuable insights into how well your model is learning and whether it is overfitting or underfitting. Interpreting the

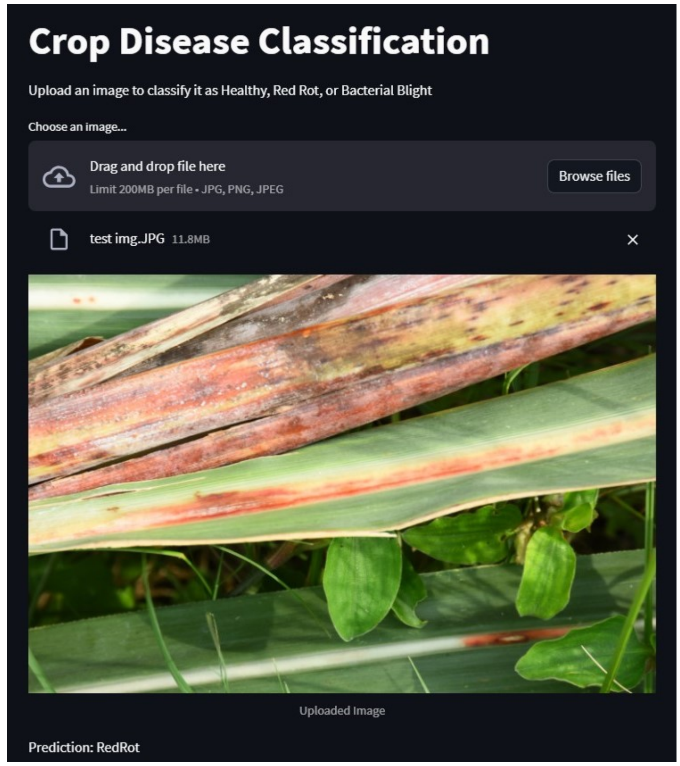

Fig. 3. Working Application

relationship between training accuracy and validation accuracy in a CNN training process: In the beginning, both training and validation accuracy may increase as the model learns to recognize patterns in the data. Over time, training accuracy may continue to increase, while validation accuracy did decrease a bit. This indicates that the model is bit overfitting the training data and not generalizing well. It's often useful to monitor this behavior and consider early stopping or other regularization techniques when validation accuracy starts to degrade.

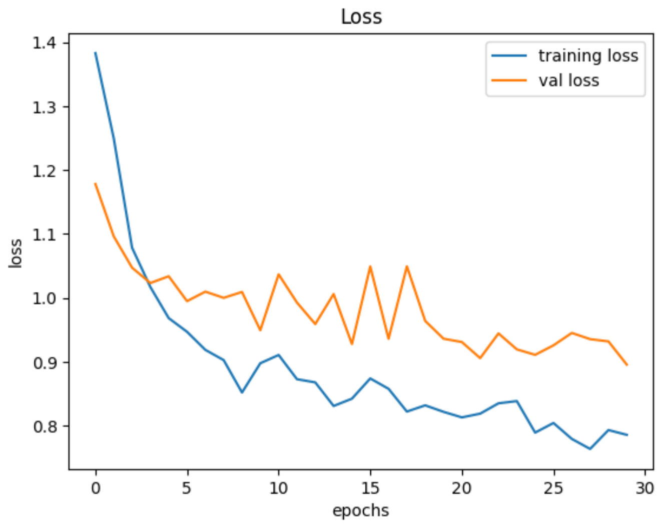

Fig. 4. Train Loss vs Validation Loss by epochs

Fig. 5. Train Accuracy vs Validation Accuracy by epochs

Figure 6: A confusion matrix is a evaluation metric that assesses the effectiveness of classification models when applied to a specific test data set which relies on having access to the actual true values of the test data to generate meaningful results. The values in clockwise order are True Negative , False Positive , True Positive and False Negative. Train accuracy: 93%, Test accuracy: 88.06%

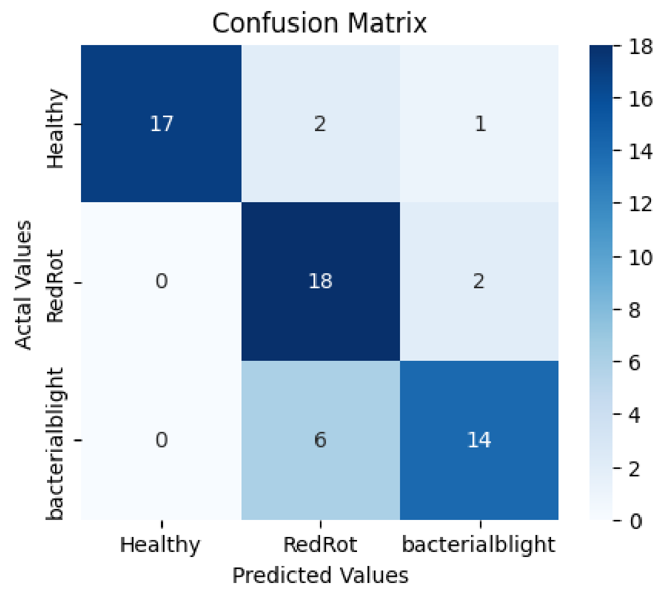

Fig. 6. Confusion Matrix

IV. Conclusion

Our model performs perfectly well considering real life scenario and able to classify between healthy: class 0, red rot: class 1 and bacterial blight: class 2 efficiently, the results have already been discussed in Result Analysis section. Further our model and study can be effectively used in implementing plant disease detection system at a large scale with the implementation of IOT based tools in a machine learning supported environment to tackle the smart agriculture problems easily. Smart agriculture, facilitated by deep learning-based disease detection, offers several benefits including early detection, increased yield, data-driven decision making, reduced human error, scalability, sustainability, data sharing and collaboration. In essence, our study paves the way for a transformative shift in agriculture, where technology and data-driven insights are harnessed to address the challenges of feeding a growing global population. By embracing deep learning for crop disease detection within a smart agriculture framework, we empower farmers with the tools they need to secure abundant and sustainable harvests in an increasingly complex and interconnected world.

References

[1] Vaishali Wadhe, Rashmi Dongre, Yash Kankriya, and Anish Kuckian. Sugarcane disease detection using deep learning. In *2022 5th International Conference on Advances in Science and Technology (ICAST)*, pages 40–44, 2022.

[2] Sammy V. Militante, Bobby D. Gerardo, and Ruji P. Medina. Sugarcane disease recognition using deep learning. In *2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, pages 575–578, 2019.

[3] Hyeon Park, Jee-Sook Eun, and Se-Han Kim. Image-based disease diagnosing and predicting of the crops through the deep learning mechanism. In *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 129–131, 2017.

[4] N.K. Hemalatha, R.N. Brunda, G.S. Prakruthi, B.V. Balaji Prabhu, Arpit Shukla, and Omkar Subbaram Jois Narasipura. Chapter 12 - sugarcane leaf disease detection through deep learning. In Ramesh Chandra Poonia, Vijander Singh, and Soumya Ranjan Nayak, editors, *Deep Learning for Sustainable Agriculture*, Cognitive Data Science in Sustainable Computing, pages 297–323. Academic Press, 2022.

[5] Alexander A. Hernandez, Joferson L. Bombasi, and Ace C. Lagman. Classification of sugarcane leaf disease using deep learning algorithms. In *2022 IEEE 13th Control and System Graduate Research Colloquium (ICSGRC)*, pages 47–50, 2022.

[6] Juncheng Ma, Keming Du, Feixiang Zheng, Lingxian Zhang, Zhihong Gong, and Zhongfu Sun. A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. *Computers and Electronics in Agriculture*, 154:18–24, 2018.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[8] Dataset. https://www.kaggle.com/datasets/prabhakaransoundar/sugarcane-disease-dataset/data.