A Design Approach for Identifying, Diagnosing and Controlling Soybean Diseases using CNN Based Computer-Vision of the Leaves for Optimizing the Production

Raj Kamal¹, Sadhna Tiwari¹, Savita Kolhe² and Manojkumar Vilasrao Deshpande¹

¹Prestige Institute of Engineering Management and Research, Vijay Nagar, Indore, MP 452010, India.
²ICAR, Indian Institute of Soybean Research, Department of Agriculture Research and Education, Khandwa Road, Indore MP 452017, India

E-mail: dr_rajkamal@hotmail.com

Abstract. Leaves express the initial symptoms of many diseases in soybean crop. CNN based Computer-Vision (CV) of the leaf-images identifies, enables the diagnosing and controlling the soybean-crop diseases. All three actions need proactive maintenance and Operational Intelligence (OI) for the model used for the control measures. The OI optimizes production. The paper describes a new design approach. Proposed design deploys Computer-Vision (CV), Convolution Neural Network (CNN), Edge AI Computing and the IoT. The approach consists of four-stage framework. Four stages are (1) Cameras in a chain of two sets of four cameras, each at two plants. (2) Edge-AI computing unit for each pair of plants at the field. (3) Edge-Server, which serves multiple networked edge units, one each of the two, sets of cameras. The server also functions as gateway for Internet connectivity with Cloud services. (4) Cloud services for the OI, proactively control the updating CNN model feature metrics, CV functions and camera operations.

1. Introduction
Soybean is globally important crop. Diseases cause significant loss, sometimes up to 100% losses in the case they are not identified, diagnosed and controlled at early stages. Emerging technologies now a days in uses for images classifications are, Convolution Neural Network (CNN), IoT, Cloud, Analytics, and AI based Intelligence and operations. A CNN model computes in the series of successive steps.

Leaf-images identify, diagnose and control the crop disease deploying the CNN technology since 2016. The performance was tested on apple, grape, pear and cherry for 13 plant-diseases. The CNN model used the data augmentation as first step before the image-input applies to the model. OpenCV Python used for the image augmentation. An open-dataset was used that consisted of 39 different classes of plant leaves and background images [1].

The CNN model developed and suggested for 14 crop plant species and 26 diseases for the automatic identification of plant diseases [2] The methods deployed on a public dataset which consisted of 54,306 images of diseased and healthy plant leaves together, collected under controlled conditions and wet rains. It identified (or absence thereof) with over 99% accuracy.

The CNNs [3] used soybean leaf-images for plant disease-parts identification and classification the computations deployed the deep CNN model. The model used was LeNet. The model deployed data-
augmentation before the inputs to CNNs. The model also assigned reduced weights to specific image-segments. The model performed the plant disease classification with 12,673 samples which contained leaf images taken under uncontrolled environment. Their results gave very-good disease-classification performance up to 99.32% accuracy.

SoyNet for diseases classification used in 2020 for the leaves’. The SoyNet identifies and classifies the plant disease effected parts [4] The SoyNet deploys the CNN model with images augmentations using number of image processing actions. Then, the image-data pre-processes for extraction of leaves part by background subtraction. The images were the ones taken in the natural environment. SoyNet-model used deep CNN model, and that gave very-good disease-classification performance, 97-98%. The model development deployed 12,068 training and 5172 validation samples containing leaf images.

A deep CNN and six types of data augmentation methods were deployed that showed improved performance [5]. The augmentation methods used were flipping, rotation, principal component analysis (PCA), colour enhancement, correction of gamma and noise injection.

A classification using CNN method deployed incept structure to fuse the extracted high-level features in deep CNN for disease identification, namely in apple, cedar and cherry and corn leaf images [6].

An improved deep CNN deployed in tea leaves images for disease identification [7]. An improved multiscale feature-extraction model was suggested. It used deep depth-wise separable convolution. That required less number of parameters and lowered the computational needs. That also gave at the same time some improvement in the model performance.

The multi-scale convolution kernel, dilated convolution and global pooling layer give the enlarged local receptive field, and enhance the feature extraction ability of the convolution layer [8]. This approach for the model deployed the robust model steps on the cucumber leaf for diseases-identification. It gave improved disease recognition accuracy.

The features learned using a CNN model at various processing layers in the hierarchy and deployed the attention-mechanism with validation for the datasets. The model computations used the 5-fold cross-validation [9]. The model deployed for tomato-leaf diseases’ identification and gave improved overall accuracy 98%.

The transfer learning approach for deep CNN model creation used the image pre-processing and augmentation. A pre-trained model learns from deploying the typical massive datasets and identifies the leaf-diseases [10]. The approach uses VGGNet, which was pre-trained on ImageNet along with an inception module. It gives accuracy near to 92% for rice and maize. It needed lower computing power with lower complexity.

A review of over 20 researches in many plant species on CNNs for the automatic identification of plant diseases is given at [11].

Do identifying and diagnosing really need deep CNN models? They need large computing power and involve computing complexities. Concepts of shallow CNN with support vector machine (SVM) and with Random Forest are proposed for vegetable crop leaf images for disease identification [12]. The concept of multi-scale feature extraction module along with deep depth-wise separable convolution suggested in [7] also requires less number of parameters and has thus lower computational needs.

This paper presents a design approach using the concepts of AI, IoT and Computer-Vision (CV) based identification, diagnosing and control measures. The steps before CNN model computations use image augmentation and pre-processing.

The design thus deploys four stages: (1) the images acquisition in the natural bright light environment and uses local computing units (LOCs) (2) Edge AI Computing for inferencing disease, severity and control measure, and communication with stages 1 and 3, (3) Gateway Edge server, and (4) Cloud data storage, model creation and continuous updating, and developing and improving OI.

Section 2 describes the proposed design of stage 1. Section 3 describes edge-computing unit of stage 2. The unit connects, configures, CV data acquisition, CNN model, and controls the camera functions and operations using sensing and display interfaces. Section 4 describes design for the edge server functions of stage 3. Section 5 gives design functions which uses the Cloud Services of Stage 4. Section 6 gives the implementation steps, results and discussion results.
2. Design of Stage 1 for the Cameras and Hardware

CV hardware uses fixed view cameras, stepper and servo-motors controlled cameras for height and rotation adjustments for visualizing the leaves for disease classification. The images are taken at regular predefined weekly, 4 days’ intervals or as configured. Stage 2 Edge also configures the height and angle, data acquisition and communication. The following subsections give the details of module:

2.1. Camera Module Hardware for Soybean plants.

Assume that module design deploys two sets of vertical columns of cameras, four cameras, \( C_i(A) \), \( C_i(D) \) and four cameras \( C_i(E), \ldots, C_i(H) \) at two growing plants. They cover in all eight-camera stages in view. Figure 1 shows the block diagram of the field with two sets of cameras, assembly in a camera sets.

![Figure 1. Block Diagram of Two Sets of Cameras Focused on Soybean Leaves of Soybean Plant for a CY System with Two LOCs](image)

The module includes \( LOC_{i1} \) and \( LOC_{i2} \), and the height and angle adjusting assemblies \( A_{i1} \) and \( A_{i2} \). A chain of pair of plants is used to cover data from more plants. \( LOC_{i1} \) and \( LOC_{i2} \) communicate with the Edge Computing Unit \( E_i \). (Section 3) The communication uses wireless protocol for ZigBee, Bluetooth, Sigfox or LoRaWAN network for communication and motor-assembly for camera-positioning control. The LOCs communicate the data-acquired at the module. Controlling software at Edge Computing unit (Section 3), activates cameras and associated adjusting assemblies, \( A_{i1} \) and \( A_{i2} \).

The LOCs implements with RPIs and camera interfaces. The Edge Computing unit computations implement on RPI4 with camera and OLED display interfaces. It is small size single board computer for edge for each assembly of the camera sets. Alternatively, Arduino board ARM with ML processing hardware can be used. [Arduino Nano 33 BLE Sense, which enables AI in the pocket size Arduino’s tiniest form factor].

Figure 2 shows input of soybean leaf sample images from camera, one image is of healthy leaf and rest are of 9 leaves exhibiting symptoms: (1) Healthy, (2) Alternaria leaf spot, (3) Bacterial Blight, (4) Bacterial Pustule, (5) Frog Eye Leaf Spot, (6) Myrothecium leaf spot, (7) Yellow Mosaic Virus (YMV), (8) Cercospora Leaf Blight, (9) Powdery Mildew, and (10) Rust.
3. Design Stage 2 for the Edge AI unit for camera Module Hardware and Edge AI

Figure 2. Soybean Images of Healthy Leaf and 9 Leaves Exhibiting Symptoms

Edge is in-between two networks, (i) consisting of the data acquisition sources. Sources of data are the local/internal networked cameras and LOCs. (Refer figure 1), and (ii) network of Edge AI units with edge gateway-server (Stage 3) and/or cloud (Stage 4).

The edge connects to cameras nearer to the plant leaves, which means data sources from where data originate at one hand. Edge computing unit connects to gateway, server or cloud on other hand. Edge communicates to server/cloud using Internet protocols.

Edge computing is a type of computing. It modifies and assists the IoT computing and applications at Stage 4 cloud or web-servers in a centralized enterprise. (Section 5) The applications and services can execute at edge computing units itself. The computational latencies, and thus time delays in the actions are small at the edges.

3.1. Edge AI Computing Unit

Figure 3 shows function and operational views of Edge AI computing unit. The latencies are very small in the edge computations, compared to the designs using cloud services IoT/IoT, therefore, deploy edge units [13]. The AI algorithms enable real-time prediction/decision/knowledge discovery for the actions/autonomous control. Edge AI computing based design is design using images augmentation, datasets preprocessing, ML (machine-learning) and DL (deep-learning) model which are computationally intensive (depend on CNN layers depth). (Computer here means computing hardware and software producing results from the instructions.) [14].

New generation single board computer boards, such as Raspberry Pi4 with neural computing and Arduino ARM Cortex with Portenta H7 Vision Shield are fast. They run the AI algorithms on advanced processors such as i9 and i10 or ARM Cortex [15].

3.1.1 Edge AI Computing Unit CV module Software: Edge-AI computing unit Ei runs software using CV module. CV module has five components: (i) image augmentation, (ii) image preprocessing, (iii) CNN model, (iv) AI inference from the feature metrics and maps, and (v) display interface.

Step i image is subtracted by the background in order to create leaf only image as shown in figure 2. Step ii converts intensity and RGB (red, green and blue) into components as follows: (a) one
intensity lightness (from black 0 to white 100), (b) colour components green (-ve 127 to 0) to red (+ve 127 to 0) and (c) component blue (-ve 127 to 0) /yellow (+ve 127 to 0).

**Figure 3.** Stage 2 Edge Computing Unit Functional and Operational Views

Data-point for each image pixel thus consists of 5 elements in each data-point [two for horizontal pixel, vertical pixel positions and three components a, b and c. Each three-dimensional (3-D) tensor corresponds to 5 elements of position, intensity and colour in a pixel.

**Figure 4.** Camera Input, Image Augmentation Steps, Data-Points, Convolution and Pooling Layers from 6, 5. Down to 1, 0

Figure 4 shows the steps i to 5. Neuron is a connection of a datapoint in a segment of set of data-points or at a layer which connects and affects/activates another datapoint, and thus to neuron of next
layer. The effect on next layer neuron is after applying a function, such as pooling, convolution, activation function or a mathematical relationship.

Neural Net collects, analyses and predict in successive series of steps. The output of each layer is called activation map, as it activates the next layer. The steps in CNN model computations with processed image input are as follows: 3-D T tensors data-points are inputs to trained and validate CNN model. W and b are the weight and bias metrics. These metrics are used from intensive computations at stage 4 for the model development. The stage 2 uses these metrics as such.

The outputs using W and b are 25 x 1 Neural Net from which the Feature Map extract and the metrics F, D, S and C for the inference. Feature F, a vector metrics of p elements (1× p), D diagnosed disease probability of (Most likely, Likely and Suspected) vector metrics of q=4 elements (1x q), C suggested control measures metrics of r elements, and S degree of severity metrics of s elements (1× s).

The input layer to CNN is after the pre-processing step. The inputs are pre-processed image data-points. The projections of a data-point along the multiple axes (domains) can be represented by a 1-D, 2-D, 3-D tensor T or set of vector components. A tensor element may be a mathematical, statistical or feature variable. The computations in neural network are considered as tensor flow computations.

ML/DL model algorithms. They run on TPU (Tensor Processing Unit) and TensorFlow Lite library. The figure marks layers from 6, 5… 1, 0 in order, left to right.

The CNN computes by tensor flow instructions from left to right side layers. The convolution layer convolutes using multiplications or dot products. The input of convolution layer includes the weights W and biases b from convolution trained and validated at the cloud stage 4. As the array elements of W and b in the model give the output close to the accurate features [4].

The output neuron layer is feature map from CV-CNN model. The feature map infers the metrics feature output. Feature map also infers the S, D and C from F.

The model is created at cloud after using training and validation datasets. A Cloud service does the computations of weights, biases metrics from training datasets, and finalises the model after validation datasets.

Consider the leaf images and corresponding diseases D in figure 2. Table 1 gives the CV-CNN output feature map for finding fully connected features F elements identified (Symptoms observed on leaves only). The inference F gives degree of Severity S [sever (80%-100%), moderate (40% to <60%), light below threshold (<40%)], and the diagnosed diseased D probability (Most likely, Likely and Suspected) and Control Measures as per recommendations by ICASR Scientists [16].

| Serial Number | Features F | Degree (S) | Diagnosed Disease elements (D) | Control Measures recommended |
|---------------|------------|------------|--------------------------------|------------------------------|
| 1             | Small, angular, translucent, water soaked yellow to light-brown spots | 80%         | Most likely Bacterial Blight | Foliar spray of copper oxychloride (0.2%)+ Streptocycline (0.2%) |
| 2             | Center of spot turn reddish brown to black | 60%         | Likely Bacterial Blight | Same as above |
| 3             | Minute, pale green elevated spots on interveinal areas on both sides of the leaves | 60%         | Likely Bacterial Pustule | Kasugamycin/validamycin(1.6kg)+ copper oxychloride(1.6kg) in 800 Liter of water/hector |
| 4             | Small raised gray to yellow pustules with pale | 80%         | Most Likely Bacterial | Spray of copper oxychloride (1.6kg)+ Streptocycline (160g) |
|   | Description                                      | Percentage | Diagnosis                              | Treatment                                                                                   |
|---|--------------------------------------------------|------------|----------------------------------------|--------------------------------------------------------------------------------------------|
| 5 | Grayish-green water soaked spots with green halo around them | 40%        | Suspected Frog Eye leaf spot            | No action                                                                                   |
| 6 | Small light brown circular to angular spots      | 40%        | Suspected Frog Eye leaf spot            | No action                                                                                   |
| 7 | Ashy gray spots with purple to dark brown margin | 60%        | Likely Frog Eye leaf spot               | Spray of Carbendazim or Thiophanate methyl @ 0.05% or Zineb or Ziram (0.2%)                 |
| 8 | Small round brown spots                          | 30%        | Suspected Myrothecium Leaf Spot         | No action                                                                                   |
| 9 | Dark brown spots surrounded by translucent areas in concentric rings | 60%        | Likely Myrothecium Leaf Spot            | Spray of Carbendazim or Thiophanate methyl (0.05% to 0.1% or 400 to 800 g/ha)               |
| 10| White erumpent structure on spots with black dots| 80%        | Most Likely Myrothecium Leaf Spot       | Spray of Carbendazim or Thiophanate methyl (0.05% to 0.1% or 400 to 800 g/ha) or Mancozeb (0.25% or 2 kg/ha) |
| 11| Yellow mosaic along veins                        | 60%        | Likely Yellow Mosaic Virus              | Thiomethoxam 25WG @100 g or methyl Dematon 25 EC @0.8 lit/ha or Ethofenprox 10 EC @1.0lit/ha |
| 12| Severe yellow mosaic and mottling of leaves      | 80%        | Most Likely Yellow Mosaic Virus         | Thiomethoxam 25WG @100 g or methyldematon 25 EC @0.8 lit/ha or Ethofenprox 10 EC @1.0lit/ha |
| 13| Purple patches on leaves                         | 40%        | Suspected Cercospora Leaf Blight        | No action                                                                                   |
| 14| dark purple colouration with bronze highlights   | 60%        | Likely Cercospora Leaf Blight           | Spray of copper oxychloride @0.2% or Carbendazim (0.05%)                                    |
| 15| White powdery patches                            | 60%        | Likely Powdery Mildew                   | Spray of Dinocap (Karathane) or Carbendazim @0.1%                                          |
| 16| Violet-White powder                              | 80%        | Most Likely Powdery Mildew              | Spray of Dinocap (Karathane) or Carbendazim @0.1%                                          |
| 17| Tan or reddish brown spots on underside          | 60%        | Likely Soybean Rust                     | Spray of Hexaconazole (Contaf) or Propiconazole (Tilt) or Triacimefon (Bayleton) or Oxycarboxin (Plantvax) @0.1% |
| 18| Blisters on Tan or reddish brown spots on underside | 80%    | Most Likely Soybean Rust                | Spray of Hexaconazole (Contaf) or Propiconazole (Tilt) or Triacimefon (Bayleton) or Oxycarboxin (Plantvax) @0.1% |
| 19| Yellow leaves                                    | 90%        | Most Likely Soybean Rust                | Spray of Hexaconazole (Contaf), Propiconazole (Tilt), Triacimefon (Bayleton) or Oxycarboxin (Plantvax) @0.1% |
4. Design Stage 3 between Edge and Gateway for Cloud Edge Server
The data communicate to Internet/Cloud for IoT for the database/analysis/Operational Intelligence. Edge-Server serves multiple networked edge units. IoT technology uses the Edge Server and web/Cloud services. Data are collected and analysed at Edge AI computing units and also communicate among themselves and take decisions/act autonomously (Section 3).

5. Design Stage 4 for the Cloud Services
The data objects and edge computing multiple units collaborate with the cloud platform for collection, analysis and predictions/decisions/OI guided control measures. Figure 5 shows edge computing multiple units, edge server and cloud deploying the databases, software for performance analytics, and OI for intelligent control measures and advance measures. OI is used optimise disease prediction accuracy and thus the production.

The W and b in the figure are the input feedback of neurons weights and biases in convolution layer from the output neuron layer (for the feature map and metrics). The sets contain input datasets along with the corresponding neural net output (figure 4) for each input. The outputs include observed or expected responses, feature maps and feature-metrics. Feature maps/feature-metrics enable AI inferences for D, S and C and take decisions.

Many training sets to the model steps followed by validation give the sound ability to extract features in new input unknown-dataset, and make operationally intelligent decisions using the model. The validation datasets are used on trained model. They validate the model steps for the features extraction and decisions. The input data may be first augmented before they are input to the CNN model. A deep CNN model is also used that performs multiple sets of convolution steps with in-between pooling and activation functions.

6. Implementation Steps, Results and Discussion
The design approach implementation steps are:

1. Cameras in two sets of four cameras each at two plants. The i-th camera sets Ci1 and Ci2 acquires and communicates the images in the field with growing soybean crop at regular 4 or 7-day intervals. (Figure 1).

2. Edge-AI computing unit for each pair of plants at the field. The unit runs software for CV using CNN using Raspberry Pi with neural computing. The edge AI computing unit (figure 3) uses Raspberry Pi 4 with neural computing stick. It connects and configures data acquisition and camera i-th assembly control Ai1 and Ai2 (figure 1). The Edge CV-CNN model computations (figure 4) use the CNN model created and updated at the cloud.

3. Edge-Server, which serves multiple networked edge units, one each for the two, sets of cameras. Edge-Server also function as gateway for the Internet connectivity with Cloud services. The Edge server (figure 5) assemble the results from number of Edge AI units for many pair of plants.
4. Cloud services consist of the databases, CNN model training and validation, predictive analytics for computing the OI for optimizing the control measures and production. The service proactively models the CV functions and camera operations. The OI uses ML with the analysed data of the CV- CNN model and operations. Cloud services provide data storage, perform analytics, create and update the model and Operation Intelligence for the Edge AI unit actions of configuring, camera-set assembly and control measures.

The features of the design are:
1. Less computational parameters and steps in the CV module at Edge AI unit. The unit implements easily on single compute board, such as Raspberry Pi 4 with neural computing module, display interface, and remote LOCs and camera system interface (Section 3).
2. The model creation by training and validation part uses cloud services. The model also improves from new observed datasets found successful in classifying. The data acquires after 4 days or weekly intervals of growth phases. The cloud stage also develops improved OI for the control measures.
3. The design implements with cameras assembly at multiple pair of plants in soybean field. The design implemented as per approach described above needs the analysis of the results and evaluation performance metrics. This work needs one crop cycle. This work will start from the incoming crop cycle.

Suggestion for future work and directions are (i) in-site application and verifications of the proposed design approach in several crop cycles, and (ii) computations of precession, recall, accuracy and confusion matrix. [4].

The Edge using OLED display interface will enable offering innovative field-device to identify, diagnose and advise control measures for the diseases. Further, the design can also be used for other plant species and fruits and for the greater number of diseases.

7. References
[1] Sladojevic S, Arsenovic M, Anderla A, Culibrk D and Stefanovic D 2016 Deep neural networks based recognition of plant diseases by leaf image classification Computational Intelligence and Neuroscience 2016 Article ID http://dx.doi.org/10.1155/ (New York: Hindawi Publishing Corporation) 3289801.
[2] Sharada P M, Hughes D P, and Salathé M September 2016 Using deep learning for image-based plant disease detection— The Methods Frontiers in Science 10 (Rosemead, CA Scientific & Academic Publishing) 01419.
[3] Wallelign S, Polceanu M and Buche C 2018 Soybean plant disease identification using convolutional neural network Proc. The 31st Florida Artificial Intelligence Research Society Conf. (FLAIRS-31) p 146–51.
[4] Karlekara A and Seal A 2020 SoyNet: Soybean leaf diseases classification Computers and Electronics in Agriculture https://doi.org/10.1016/j.compag. 172 (Elsevier) 105342.
[5] Geetharamani G, Pandian J A, Agarwal M and Gupta S K 2019 Identification of plant leaf diseases using a nine-layer deep convolutional neural network Computers & Electrical Engineering 76 (Elsevier) p 323–328.
[6] Hang J, Zhang D, Chen P, Zhang J and Wang B 2019 Classification of plant leaf diseases based on improved convolutional neural network Sensors 19 (Basel, Switzerland) pp. 1-19.
[7] Gensheng H, Xiaowei Y, Yan Z and Mingzhu W 2019 Identification of tea leaf diseases by using an improved deep convolutional neural network Sustainable Computing: Informatics and Systems https://doi.org/10.1016/j.suscom. (Elsevier) 100353.
[8] Zhanga S, Zhangb S, Zhange C, Wangx X and Shia Y July 2019 Cucumber leaf disease identification with global pooling dilated convolutional neural network https://doi.org/10.1016/j.compag.03.012 Computers and Electronics in Agriculture 162 (Elsevier) p 422–430.
[9] Karthik R, Hariharan M, Anand S, Mathikshara P, Johnson A, and Menaka R 2019 Attention embedded residual CNN for disease detection in tomato leaves https://doi.org/10.1016/j.asoc.2019.105933 Applied Soft Computing 162 (Elsevier) 422–430.
[10] Junde C, Jinxiu C, Zhanga D, Sunb Y A and Nanekharana Y 2020 Using deep transfer learning for image-based plant disease identification https://doi.org/10.1016/j.compag. Computers and Electronics in Agriculture 173 (Elsevier) 105393.
[11] Poonia, Ramesh C., et al., eds. Smart Farming Technologies for Sustainable Agricultural Development. IGI Global, 2018.
[12] Li Y, Jing N and Chao X October 2020 Do we really need deep CNN for plan diseases identification? https://doi.org/10.1016/j.compag. Computers and Electronics in Agriculture 173 (Elsevier) 105803.
[13] Shi, W, Cao J., Zhang, Q, Li, Y and Xu L 2016 Edge Computing: Vision and Challenges, IEEE Internet Of Things J., 3(5) pp. 637-646.
[14] Monroe Y 2018 Chips for Artificial Intelligence https://cacm.acm.org/magazines/ 2018/ 4/ 226374- chips-for-artificial-intelligence/abstract Communications of the ACM, 61(4) pp. 15-17 (ACM) 10.1145/3185523.
[15] Kamal, R 2020 Embedded Systems SoC, IoT, AI and Real Time Systems (Chennai McGraw-Hill Education India) 4th edition chapters 5 and 6 pp. 168–256.
[16] Gupta G K and Chauhan G S 2005 Symptoms, identification and management of soybean diseases Technical bulletin, ICAR -Indian Institute of Soybean, Indore, MP, India.