https://doi.org/10.48047/AFJBS.6.Si4.2024.4892-4908

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

Seg Caps-Efficient Net Deep Learning Method for Accurate Plant Species Segmentation and Classification

Dr.S.Sujanthi,
Department of Computer Science and Engineering,
M.Kumarasamy College of Engineering,
Karur-639113.
s.sujanthi2021@gmail.com

Dr.K.Tamilarasi
Assistant professor
Department of ECE Excel Engineering College,
Namakkal, Tamilnadu - 637303
tamilarasi.k.eec@excelcolleges.com

Article History
Volume 6, Issue Si4, 2024
Received: 1 July 2024
Accepted: 20 July 2024
Doi:
10.48047/AFJBS.6.Si4.2024.4892-4908

***Abstract*—**Overall, agriculture's impact on a nation's economy is substantial, but in India it plays an especially vital role in improving the country's overall economic framework and standard of living. Crop yield in India is notoriously difficult to predict due to the country's erratic weather patterns and the prevalence of bacterial diseases. Researchers are focusing on this question in an effort to develop a method for identifying diseases in plants with small, green leaves at an early stage. Agriculture output relies on keeping plants healthy, and a single outbreak of a microbacterial infectious illness can wipe out an entire crop overnight. To this end, we set out to employ a deep-learning network-based technique for early detection of bacterial infections in green tiny leaves on green plants, before the onset of any outward signs. In this research, we evaluate recent advancements in the field of deep learning. To begin, we will investigate the deep neural network architecture, its data sources, and the various image processing methods that can be applied to the leaf images that have been recorded. In order to diagnose and categorize plant species, numerous DL architectures and data visualization's tools have emerged recently. To manage the functional automatic detection system for determining a particular plant illness in the field in which huge development of green small leaves on green organisms is mostly done, we had observed a problem that was unidentified in previous studies throughout our research survey and used their method to resolve that issue. CNN (convolution neural network) algorithms have recently undergone a number of improvements that result in more precise picture classification of any given object.To do this, we use the Agri-ImageNet collection in conjunction with realistically recorded photos of sunflower and

cauliflower leaves, bulbs, and flowers. The PlantVillage dataset has regulated settings and uniform backdrops; this dataset fixes that. Deep transfer learning, which involves reusing knowledge representations, may alleviate these challenges. The primary goal of this research is to examine and evaluate all available deep transfer learning approaches in order to determine which one is most appropriate for a dataset consisting of plant species. In our study, we use a deep learning network to monitor plant symptoms in the leaf and, using the plant categorization information provided by the plant dataset, to determine the particular kind of bacterial infection at work. Our study illustrates the use of a Deep Learning algorithm implemented in SegCaps and EfficientNet Model to detect leaf diseases in green plants at an early stage and under varying environmental conditions. When compared to other illness detection methods using the HybridNet model, our findings demonstrate that SegCaps has an accuracy of 96.90%.

Keywords —SegCaps, EfficientNet, Deep Learning, Plant Classificatio, and Agri-Image Net

I. INTRODUCTION

Aesthetically and functionally, plants contribute to Earth's natural ecology. For human existence, one plant in particular has been crucial. Some loss of vegetation is expected in proportion to the level of urgency. Unfortunately, these plants are in danger of becoming extinct, even though Ayurvedic medication made from their leaves has great potential. Building a database on plant defenses is, therefore, of the utmost importance. Leaves of many plants have proven elusive to more than one expert. A simple and quick way to identify plant species would be useful for amateur botanists and experts alike. The feature extraction technique benefits from deep learning because of its ability to provide more detailed information about pictures. Researching plant variety is essential because plants are useful in many fields, including medicine and agriculture. Plants play a crucial part in mitigating environmental impacts. Ansimple and fast way to recognize plants is to correctly name and classify them. By classifying plant species into useful groups, we may get a better understanding of their shared traits, evolutionary relationships, and pasts. A shortage of resources in many regions makes early diagnosis for plant diseases more problematic, despite the fact that they pose a significant danger to the world's food supply. Regular agricultural practices won't cut it when it comes to solving the world's impending food crisis. For this reason, it is essential to find innovative ways to boost agricultural output so that we can build farming systems that are sustainable in the long run and can satisfy both supply and demand. India continues to face food shortages while being one of the most agriculturally dependent countries. As an example of this kind of shortcoming, consider the botched diagnosis and treatment of plant diseases. Here we take a look at the potential for developing a computer- and AI-based automated system to detect and diagnose plant illnesses. An advanced approach for specie detection in apple plant species local to India is developed using deep learning. The target market for this product is India. Our research has shown that the EfficientNetB0 model is the smallest and most accurate one currently available; hence, the whole system for assisting with plant care—from the front-end mobile application to the main cloud database—is constructed using TensorFlow Lite. Using enhanced hyperparameters and a completely updated base network, we demonstrate that the EfficientNetB0 model achieves excellent results. As a last step, our approach may aggregate user classifications to provide spatiotemporal insights into regional and periodic illness trends. These insights are accessible to everyone using the system, with the goal of increasing knowledge of global agricultural trends. We provide the results of a research into the identification and classification of various leaf kinds in this paper. Just by looking at it, you can't determine what kind of leaf it is. Image analysis along with machine learning techniques may, therefore, prove to be quite useful in classifying leaves. All of the images included in the project were captured using digital cameras. Before moving on to the background removal phase, be sure you apply the background removal technique to the given leaf picture in

the pre-processing phase After the final output has been adjusted to eliminate the backdrop, the picture segmentation technique is applied. By analyzing a variety of segmented photos, important properties like texture may be extracted. In the last step, the characteristics that have been acquired are fed into the classifier. The objective of this study is to categorize various plant species by applying image processing to their leaves. A complex leaf image is used for training purposes. As a result, there's a chance to improve in these areas:

- Optimal hyperparameter selection;
- Reduced overfitting;
- Enhanced feature learning;
- Improved dataset quality

Classification models for leaf prediction that have been previously suggested mostly use convolutional neural network (CNN) architecture, where each CNN layer has to be explicitly specified. Regardless, their precision is weak. Consequently, a method is required that eliminates all the shortcomings of the most recent publications. Using transfer learning with the pre-trained classifiers EfficientNet and SegCapsNet, this work offers a more straightforward and practical method for plant species categorization and identification. The AgriImage dataset was used for our model experiments. Results from experiments demonstrate that the suggested model is superior to competing methods for plant leaf identification. This approach may greatly improve the accuracy of plant species detection and is easy to implement. Researchers may benefit from this method in identifying plant species, and botanists can save countless hours of work thanks to it. Even laypeople with no background knowledge of plants may benefit from this plant identification approach when it comes to recognizing and using plants.

This is how the rest of the work is structured: We review several new studies in Section 2. Section 3 details the approach that is being suggested. In Section 4, we detail the findings of the identification process. A debate follows in Section 5, and a conclusion follows in Section 6.

II. Literature Review

Provide a high-level summary of the several approaches used to date to address the problem of leaf identification in [8]. The methodologies in question have been used in a great deal of prior research. The purpose of this work is to classify plant leaves using CNN. The enhanced CNN classifier is fed a combination of texture and color characteristics extracted from the given data. Over 94.26% accuracy was achieved by the system with the use of a tensor flow architecture. Automatically categorizing seventeen distinct plant species is within the system's capabilities.

A groundbreaking method that employs convolutional neural networks (CNN) is introduced in the paper [9] as D-Leaf. To pre-process the leaf photos and extract features, three Convolutional Neural Network (CNN) architectures were utilized: a pre-trained AlexNet, an updated AlexNet, and D-Leaf. Attribute rankings were determined using the following five machine learning algorithms: Naive Bayes (NB), k-Nearest Neighbor (k-NN), Support Vector Machine (SVM), and Artificial Neural Network (ANN). Machine learning's sixth tool was CNNs, or convolutional neural networks. In order to calculate the morphological data, traditional morphometric techniques were used, using Sobel segmented veins serving as the benchmark. The D-Leaf model outperformed AlexNet (95.54%), with a test accuracy of 94.88%, finishing in the same ballpark as AlexNet (93.26%). Additionally, CNN models outperformed the conventional morphometric data utilized for such studies by a wide margin (66.55 percent). The ANN classifier has shown to be effective when fed CNN-obtained properties. Results from tests indicate that D-Leaf has good potential as a computerized method for identifying plant species.

Research in [10] makes it much easier to identify plant species by their veining patterns in the leaves. The proposed method for capturing images of leaf veins consists of four main parts: sample, preprocessing (which involves converting RGB images to grayscale, detecting sobel edges, and skeletionizing), feature extraction, and identifying features using CapsNet. In what follows, we'll go further into each of these four processes individually. The book classifies plants according to the veins on their leaves. Plants with complex leaf structures benefit greatly from it.
The authors of [11] provide a novel method for plant categorization that uses a combination of leaf edge data, SIFT, and morphological modifications to isolate critical edge regions. The method is tri-pronged. Processing images, extracting features, and classifying images. We apply Morphological Transformations to remove noise from the images and Canny Edge Detection to locate the leaf edges when preparing the pictures. Plant leaf characteristics were extracted by CNN using the proposed method, which included using SIFT to identify the edge of the leaf. Next, a random forest is used to sort the plant leaves. Ten categories, five for healthy leaves and five for damaged leaves, make up the PlantVillage data set, which was used in the experiments. When compared to traits derived from leaf shape and texture, the findings demonstrated that the suggested technique considerably enhanced the accuracy of plant species categorization. The proposed method achieves a precision of 95.62%. Discuss the use of deep learning approaches to the job of plant species classification in [12]. A leaf's vein pattern is one of the most nuanced and significant characteristics utilized for recognizing and categorizing plant species. By looking at the plant as a whole and at the leaves in particular, you may identify distinct types of plants. By analyzing visual depictions of leaf morphology, a botanist might use the recovered attributes to get a better understanding of plant characteristics. The three primary components of the proposed approach are feature extraction, classification, and picture preprocessing. The leaf photos were preprocessed so that the deep learning system could understand them. Utilizing bottleneck features allows for data reduction, while a feature extraction method and the clever edge detection technique reveal the leaf vein patterns. For training and classification on this dataset, the VGG16 CNN has been used. The research made use of the Flavia dataset, which was created via the kaggle website. This data collection included fifteen distinct photographic styles. It was shown that the success rate of the model was 95%.
To identify possible issues with apple trees, these researchers employ ResNet-34, a design of convolutional neural networks, as described in [13]. Applying the suggested model to a dataset that was split 70-30% and 80-20% yielded 97.5% and 98.4% accuracy, respectively. This study's results demonstrate that the suggested model has been extensively tested and may be used to future research of a similar kind. This is accomplished via the integration of several sophisticated models for machine learning and components that provide guidance on prevention and treatment for health issues that have been detected.
The goal of this piece of writing is to educate readers how to recognize the early symptoms of plant species so that they may improve their gardening skills and have a better understanding of horticulture. Our project includes the development of a mobile app for plant care, which we name "AgroAId." It uses AI and computer vision to identify one of 39 plant diseases or species based on an input picture of the leaves. We are doing comparative study to find the optimal convolutional neural network (CNN) architecture, transfer learning method, and hyperparameter settings for the purpose of developing the best classification with multiple labels model. Our goal was to assess four mobile-optimal thin CNNs by analyzing different learning transfer situations. The following were the possible outcomes: (1) convolution layer freezing; (2) layer freezing of 80%; (3) layer thawing of 50%; and (4) layer upgrading. The four CNNs that were evaluated did

an excellent job. Using common measures like accuracy, F1-score, as well as confusion matrices, 32 distinct model versions are being tested.

In [15], the threshold function is used to hide green pixels and a method like isotropic diffusion is used to eliminate picture noise. The employment of a classifier based on deep learning ensures accurate picture segmentation and categorization. The ability to distinguish between different leaf species and the illnesses to which they are vulnerable depends on an accurate identification system.

III. PROPOSED MODEL

Using image processing methods, plant species may be identified using plant categorization. Applications in botanical research and other plant-based enterprises may benefit from this system's ability to distinguish between various plant species. Systems have been constructed in earlier research using data sets with a small number of classes. There is need for improvement in the current plant taxonomy, nevertheless, so that a wider range of species may be included. Figure 1 depicts the process for plant species recognition. Features are extracted using DL-based SegCaps models.

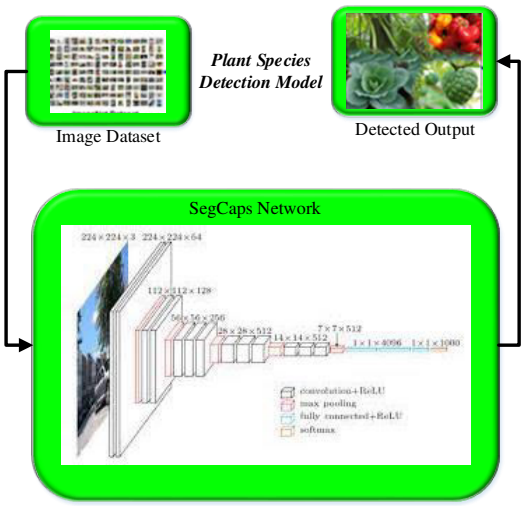


Fig.1 Proposed Model

A. *System Flow*

- **Image acquisition**—In order to conduct analysis towards categorization, this phase entails obtaining a picture of the whole plant or one of its organs.
- **Preprocessing**—Improving picture data by reducing unwanted distortions and highlighting important elements for further processing is what image preprocessing is all about. To prepare an image for feature extraction, the preprocessing subprocess takes an input image and produces a modified version of the original picture. Image denoising, content augmentation, and segmentation are common preprocessing processes. It is possible to apply them in sequence or one by one, and to repeat the process as needed to get the desired picture quality.
- **Feature extraction and description**—In image processing, feature extraction is the process of extracting relevant areas from a picture by measuring them geometrically or otherwise. The features of a plant or its organs seen in photographs are defined by a collection of numerical characteristics, often called descriptors.
- **Classification**—The classification process begins with merging all of the retrieved features into a feature vector.

B. Color Constance

When compared to more traditional methods like shades of grey, grey world, and max RGB, the image is significantly enhanced by the use of the second order grey edge method for colour constancy.

The input image I has its colour effects subtracted before its chromaticity and illumination are calculated. Instead of using the RGB colour values, the chromaticity of the colours is adjusted. Thus, chromaticity is determined by,

$r = R/R + G + B$ (1)

$g = G/R + G + B$ (2)

$b = B/R + G + B$ (3)

Where $R, G, andB$ are the camera RED, GREEN and BLUE channel values and r, g and b are the chromaticity values.

The colour constancy procedure fixes the plant lesion image's problems with contrast, brightness, and sharpness. This effectively improves segmentation performance.

The color pixels image $I(i,j)$ where $i \in (r,g,b)$ is defined by illumination values as $C(i,j)$ and reflectance of surface $SR\ (x,y)$. It is expressed as,

$I(i,j) = C(i,j) * SR(x,y), i \in (r,g,b)$ (4)

The final colour image is created by applying histogram stretching. The below equation depicts how the histogram of a coloured image is stretched, $g(x,y) = \frac{f(x,y)-f(\min(x,y))}{f(\max(x,y))-f(\min(x,y))} * 2^{bpp}$ (5)

Each pixel's intensity is denoted by f(x,y), where bpp is the number of bits used to store that value. The colour invariant type forms the basis of this colour constancy concept. The resultant image is a combination of the R, G, and B channels. Remember that the intensity range of the final image's pixels is not exactly 0-255. The final image produced after plant lesion segmentation undergoes a linear transformation to bring it into the range of values from 0 to 255. Next, we use the input image with the changed image g (f,c) to determine the illuminant colour. To calculate the white illuminant, use the following formula,

$p(c|f) = P(w|g\ (f,c))$ (6)

The value of the illuminant colour, represented by $c_{max} = argmax_{c \epsilon C} \log(P(w|g(f,c))$ (7)

The likelihood of obtaining the color-corrected image is maximized with the illuminant colour value shown in the preceding equation. Probability-based colour constancy relies on picking the most plausible illuminant provided in the image, with pixel values standing in for the image's semantic meaning. Low contrast images are stretched using a histogram derivation from the image's illumination and chromaticity.

Noise Filtering

The ISBF is an adjustable and modified filter that analyzes each pixel to see if it has been tainted by noise. For the purpose of identifying noisy pixels in dermoscopy images, SQMV is used to calculate a reference median value. The input window wI is divided into four (N+1)(N+1)-sized subwindows, and the mean of each is then used to get the reference median

value. The reference median is determined by averaging the medians of the individual subwindows.

When the median value used as a reference is suboptimal, noise detection errors result. Noisy, low-quality pixels always have values that are negative, and this filtering obscures the finer points. Impulse noise is defined as a current pixel that deviates considerably from its reference median value. Similar to how it becomes Gaussian noise if the level is not excessive. Therefore, noise detection relies heavily on the use of a reference median value. That's why the windows need to be able to sense the situation and adjust accordingly. Identifying sources of noise requires the following,

If $|x(i,j) - r(m)| \geq T(n1)$
$x(i,j) = Impulse\ Noise$
Else if $|x(i,j) - r(m) \leq |T(n2)$
$x(i,j) = Gaussian\ Noise$
Else
$x(i,j) = noise\ free$

Segmentation

After the preprocessing is complete, the unique SegCaps method is used to accomplish the segmentation. The SegCaps algorithm efficiently identifies the area of dermoscopicplant damage. In addition, it takes into account the location, direction, context, and intensity of a plant lesion image to segment the lesion region accurately. In comparison to conventional algorithms like U-net and CNN, it is able to make use of large images while simultaneously decreasing the number of parameters required for the plant lesion division process. In order to partition the area of plant damage, it collects edge-enhanced input. Convolutional, primary cap, and segcap are the three main building blocks. The segcaps block in this case applies a mask to the provided lesion image by use of the reconstruction convolutional layer. At last, it gives you the segmented image of the plant lesion you fed into it. The SegCaps method entails the steps below**.**

SegCaps takes a 512x512 pixel image as input, which is then processed by a 2D conv_layer. In this layer, you'll find 16 parallel sets of feature maps. This is the first group of capsules where each capsule has a single type with a size of 512 by 512 pixels and a vector size of 16 dimensions. A layer L consists of $\mathfrak{M}$ set of capsule types such as $T^L = \{t_1^L, t_2^L, \dots, t_n^L\}$. For every $t_i^L \in T_i^L$, $\mathfrak{M}$ an $H_L \times W_L$ grid of Z_L dimensional child capsules,

$C = \{c_{11}, \dots c^l{}_{1W} \dots c_H L_1 \dots \dots C_H L_W L\}$ (8)

Where $H_L \times W_L$is the output layer L-1's spatial dimension. The subsequent layer of the network, l+1, has a variety of capsules defined as

$T^{L+1} = \{t_1^{L+1}, t_2^{L+1} \dots t_m^{L+1} | m \in N\}$ (9)

For every $t_j^{L+1} \in T_j^{L+1}$, $\mathfrak{M}$ an $H_{L+1} \times W_{L+1}$ grid of Z^{L+1} dimensional parent capsules $P_{xy} \in P$ obtains a set of vectors called as prediction vectors. It is denoted as,

$U = \{\bar{u}_{xy}|t_1^L, \bar{u}_{xy}|t_2^L \dots \bar{u}_{xy}|t_n^L\}$ (10)

Each one for capsule types as T^L. This capsule types set is computed between two information as conversion matrix $M_{t_1^L}$ and the sub-grid child cases$U_{t_1^L}$. Therefore, $M_{t_1^L}$ and $U_{t_1^L}$ are multiplied together to obtain each layer information, which is expressed as,

$\bar{u}_{xy}|t_n^L = M_{t_1^L} * \ U_{t_1^L}$ (11)

For all capsule types $t_j^{L+1} \in T_j^{L+1}$, $\bar{u}_{xy}|t_n^L$ has a shape of $k_H \times k_W \times Z_L$ where $k_H \times k_W$ are the User Defined Kernel Types. Each $M_{t_1^L}$ has shape $k_H \times k_W \times Z_L \times |T_{L+1}| \times Z_{L+1}$, where $|T_{L+1}|$This represents the total amount of parent capsule types in layer L+1. Importantly note that all dimensions of the $\bar{u}_{xy}|t_n^L$ are not depend in the physical space the transformation matrix describes. The back propagation method determines the spatial information values for each capsule type.

A parent capsule's ultimate input, denoted as Pxy∈ P, is determined by computing the total values of prediction vectors, which are written as,

$$P_{xy} = \sum_n R_{t_i^L|xy}\, \bar{u}_{xy}|t_n^L \qquad (12)$$

Where $R_{t_i^L|xy}$ is the dynamic routing scheme's calculated routing coefficients. The softmax function is used to calculate these values. To calculate it, use the formulas below,

$$R_{t_i^L|xy} = \frac{e^{\left(b_{t_n^L|xy}\right)}}{\sum_k e^{\left(b_{t_n^L|xy}\right)}} \qquad (13)$$

Where $b_{t_n^L|xy}$ is the log prior likelihoods that forecast vector $\bar{u}_{xy}|t_n^L$ is routed via parent capsule p_{xy}. The proposed technique uses different routing algorithm. The modifications are follows,

As an alternative to routing by each child capsule for each parent capsule, the prediction vectors calculation is constrained locally, and child capsules inside the User Defined Kernel are routed to the Parent. The output capsule is computed by "Non-Linear Squashing Function"

$$V_{xy} = \left\|p_{xy}\right\|^2 \Big/ 1 + \left\|p_{xy}\right\|^2 \frac{p_{xy}}{\|p_{xy}\|} \qquad (14)$$

Where V_{xy} represents the capsule vector output at spatial site(x, y) and p_{xy} is the final input. The capsule outputs are then used to create the final segmented mask.

Transfer Learning Models for Classification

(i). ResNet-50

One solution to the deterioration issue with deep neural networks was the idea of residual connections, which ResNet-50 put forth. Shortcut connections, fully connected layers, pooling layers, convolutional layers, and a total of fifty layers make up the ResNet-50 architecture. Instead of directly fitting the intended underlying mapping, ResNet-50 learns residual functions with the help of residual blocks, which is the main novelty of the network.

Two or three convolutional layers using batch normalization with ReLU activation algorithms make up each residual block in ResNet-50. After these layers, a skip connection is formed, which links the input of the leftover block to its output immediately. To lessen the impact of disappearing gradients, this skip link makes gradient flow easier during training.

The mathematical representation of the skip link is as follows:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \mathbf{W}_i) + \mathbf{x} \qquad (15)$$

that is, x is the input to the residual block, $\mathcal{F}(\mathbf{x}, \mathbf{W}_i)$ is the residual function that the convolutional layers with weights W1 apply, and y is the output of the residual block. With the addition of these residual links, ResNet-50 successfully addresses the issue of information deterioration in deep networks, enabling the model to go deeper while keeping or exceeding its performance. In order to learn deeper and more complicated characteristics, the skip connections improve the gradient's propagation across the network.

(ii). DenseNet-201

The information flows efficiently and there are a lot of layers in DenseNet-201. A number of dense blocks, each with several convolutional layers that are tightly coupled, make up the DenseNet-201 architecture. New to DenseNet-201 are dense connections, which enable direct connections among layers of varying depths, as opposed to the sequential flow of information in conventional CNN layers. The effective flow of information across the network is made possible by this dense connection, which also encourages feature reuse.

Multiple densely connected layers, usually including convolutional layers, batches of normalization, and activation functions, make up each dense block in DenseNet-201. All the characteristic maps from the layers before it are combined with each layer's output in a single dense block. In mathematics, this action of joining is expressed as

$$\mathbf{x}_l = [\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{l-1}] \quad (16)$$

where$\mathbf{x}_l$is the dense block's i-th previous layer's feature maps, and h_l is the l-th layer's output.

To promote feature reuse and provide the network access to a rich collection of characteristics at each of them, DenseNet-201 tightly connects the layers inside each dense block. As a result, the model becomes smaller and the vanishing gradient issue is reduced, making deep network training more efficient. For even more control over the feature map count and spatial dimension reduction, DenseNet-201 uses transition layers between dense blocks. In order to reduce data size and increase computing performance, these transition layers usually include a convolutional layer and a pooling layer.

(iii). **Xception**

Aiming to collect more fine-grained spatial data to increase feature representation, Xception is famous for its novel approach to convolutional layer architecture. Separating the spatial and channel-wise aspects of the convolution process, the idea of depth-wise separable convolutions forms the basis of the Xception architecture. Xception offers distinct operations for every dimension, resulting in greater efficiency and efficacy, in contrast to typical convolutional layers that conduct convolutions across both the spatial and channels dimensions concurrently.

Applying spatial convolutions separately for every channel and then combining the results via pointwise convolutions is the main notion underlying depth-wise separable convolutions. It is possible to express the separable convolution depth-wise procedure mathematically as

$$\mathbf{Y} = vul(\mathbf{X}) * \rho ul(\mathbf{Y}') \quad (17)$$

In this context, X stands for the input feature maps, DUl for the depth-wise convolution functioning, DUl for the pointwise convolution functioning, Y^' for the intermediate value maps, and Y for the output feature maps.

Xception drastically cuts down on computation and parameter requirements as compared to conventional convolutional layers by splitting both spatial and channel-wise processes. Better learning and representation capabilities are made possible by deeper network topologies with fewer parameters. Following in the footsteps of the ResNet design, Xception includes depth-wise separable convolutions as well as residual connections. By establishing these links, the network is able to train extremely deep networks efficiently while learning residual functions and avoiding the vanishing gradient issue. The Xception architecture is also modular, with convolutional blocks and pooling layers appearing in repeating sequences. This architecture makes it easy to extract hierarchical features at various abstraction levels, which means the network can pick up on both simple and complex visual patterns.

An essential part of any computational recognition system are the properties of leaves, including their size, color, and form. Both contour-based and region-based extraction methods are

available for feature extraction. The dimensions used to describe the contour-based extraction are the leaf diameter, aspect ratio, breadth, and length. The length description is measured along the leaf's primary vein, from which it extends all the way to its tip. When looking at a leaf from the side, the width descriptor is the distance between the two extremes, measured from the top to the bottom. You may get the aspect ratio with dividing the length of the leaf by its breadth. The diameter of a leaf measures the greatest distance between any two locations inside its covered area, while the perimeter measures the greatest distance beyond the leaf's covered region. Finding the total amount of pixels that include the leaf border is another way to get the leaf perimeter.

In contrast, the region-based method employs form, rectangularity, rigidity, area, and eccentricity as descriptors for leaf characteristics extraction. The shape, also known as the convex hull, is defined as the set of coordinates at which the whole leaf area may be easily calculated. If you draw an image outside of the picture and plug it into Equation (18), you'll get a rectangle with dimensions R.

$$R = LpWp/A \quad (18)$$

where the length, breadth, and area of the leaf are represented by Lp, Wp, and A, respectively.

Equation (19) shows that compactness, often called roundness, is defined as the proportion of the leaf's surface area to the squared of its perimeter. In this context, "A" refers to the leaf's surface area and "P" to its perimeter.

$$\text{Compactness} = 4\pi A/P^2 \quad (19)$$

The outcome of the thresholding procedure is the definition of area. According to Equation (), the area of a leaf is defined by the pixel that has the value 1. Any part of a leaf that is conic in shape will have eccentricity. In Equation (20), we may get this characteristic by substituting the minimal elliptical axial length ('a') and the shortest axial length ('b').

$$\text{Eccentricity} = \sqrt{\ }(\&1 - [\quad[(b/a)]\quad]^2) \quad (20)$$

To provide state-of-the-art performance on picture identification tasks, EfficientNet employs a deep neural network design that combines model scaling with neural architecture search (NAS). An architecture for convolutional neural networks (CNNs) called EfficientNet aims to maximize the network's depth, breadth, and resolution all at once. The design does this by using both NAS and model scaling. When compared to pre-existing transfer learning networks, this one performed much better because to the researchers' adoption of a compound scaling strategy to consistently and firmly grow the network's dimensions.

Neural Architecture Search (NAS)

Finding the optimal neural network design for a certain job may be automated using NAS. Building structures that maximize a given objective function is the goal of using machine learning techniques. EfficientNet's design is generated using a mobile version of NAS dubbed "MBConv" (Mobile Inverted Residual Bottleneck Convolution).

Model Scaling

Principled network expansion in depth, breadth, and resolution is what model scaling is all about. EfficientNet scales the network's dimensions evenly using a compound scaling approach. During NAS optimization, a single coefficient "phi" controls the scale factor. By avoiding an increase in computing cost, this method significantly outperforms state-of-the-art networks.

It is equally vital to have a solid baseline network since scale of models does not alter the layer operators F i in the baseline. We will test our scaling approach with preexisting ConvNets, but we've also built an innovative mobile-size baseline, EfficientNet, to show how well it works.

A function is one way to define a ConvNet Layer i.: $Y_i = \mathcal{F}_i(X_i)$, X_i is the input tensor and Y_i is the output tensor, where F_i is the operator and X_i has a tensor shape $\langle H_i, W_i, C_i \rangle$[1], where H_i and W_i are space dimension, with C_i representing the channel dimension. The following is a list of the constructed layers that make up a ConvNet N: $\mathcal{N} = \mathcal{F}_k \odot ... \odot \mathcal{F}_2 \odot \mathcal{F}_1(X_1) = \odot_{j=1...k} \mathcal{F}_j(X_1)$. As an example, ResNet (He et al., 2016) comprises five stages, and each layer has identical convolutional type except for the first layer, which does down-sampling. In fact, ConvNet layers are often divided into many stages, with all levels sharing the same design. Hence, a ConvNet may be described as:

$$\mathcal{N} = \odot_{i=1...s} \mathcal{F}_i^{L_i}\left(X_{\langle H_i, W_i, C_i \rangle}\right) \quad (21)$$

where$\mathcal{F}_i^{L_i}$indicates that layer F_i is iterated through stage L_i times$i$, $\langle H_i, W_i, C_i \rangle$is the form that the input tensor X of the first layer takes. As shown in Figure 2(a), a typical ConvNet has its spatial dimension shrinking over time while its channel dimension expanding across layers; for instance, it goes from an initial input form of ⟨224,224,3⟩ to an output shape of ⟨7,7,512⟩.

Model scaling seeks to increase the network's length (L_i), width (C_i), and/or resolution, in contrast to standard ConvNet designs that primarily aim at determining the optimal layer architecture F_i. (H_i, W_i) without altering$\mathcal{F}_i$specified in the initial network configuration. Model scaling makes the design challenge for additional resource restrictions easier by addressing F_i, but there is still a lot of room to explore other L_i, C_i, H_i, W_iat all levels. Keeping all layers at a fixed ratio and equally sized helps us further condense the design area. Maximizing the model's accuracy within the restrictions of available resources is our goal; this challenge may be expressed as an optimization issue:

$$\begin{aligned} \max_{d,w,r} \quad & \text{Accuracy } (\mathcal{N}(d,w,r)) \\ \text{s.t.} \quad & \mathcal{N}(d,w,r) = \odot_{i=1...s} \hat{\mathcal{F}}_i^{d \cdot L_i}\left(X_{\langle r \cdot \hat{H}_i, r \cdot \hat{W}_i, w \cdot \hat{C}_i \rangle}\right) \\ & \text{Memory}(\mathcal{N}) \leq \text{ target\_memory} \\ & \text{FLOPS}(\mathcal{N}) \leq \text{ target\_flops} \end{aligned} \quad (22)$$

w, d, and r are the scaling coefficients for the width, depth, as well as resolution of the network, respectively; $\hat{\mathcal{F}}_i, \hat{L}_i, \hat{H}_i, \hat{W}_i, \hat{C}_i$parameters that are already set up in the baseline network.

IV. EXPERIMENTAL RESULTS

A machine with an Intel i7 core CPU to avoid GPU bottlenecking, a Samsung 1 TB SSD to put the dataset onto, and an NvidiaRTX 3070 configured for deep learning was used to execute this project. Along with Python and the CudNN libraries, it makes use of Nvidia'sCUDA (Compute Unified Devices Architecture).

4.1. Dataset Detail

One of the most popular benchmark datasets for computer vision object identification tasks is ImageNet, of which Agri-ImageNet is a subset. With a carefully selected collection of photos pertaining to different crops and agricultural products, Agri-ImageNet is built with the express purpose of agricultural image identification and categorization in mind. Acorn squash, zucchini, rapeseed, orange, pineapple, mushroom, lemon, jackfruit, fig, daisy, head cabbage, custard apple, cucumber, coral fungus, cardoon, butternut squash, broccoli, bell pepper, banana, artichoke, agaric, and many more classes are represented in the Agri-ImageNet dataset. Various fruits, vegetables, crops, and other agricultural products are reflected in these classes. Agriculturally relevant photos from the initial ImageNet collection are curated and classified to create Agri-ImageNet, a subset of ImageNet. For the purpose of training and testing agricultural

image classification models, this subset has been fine-tuned to concentrate on picture recognition tasks in the agriculture sector. The goal of AgriImageNet is to make it easier to study and build machine vision algorithms for use in farming, particularly in areas like crop tracking, disease identification, and yield prediction.

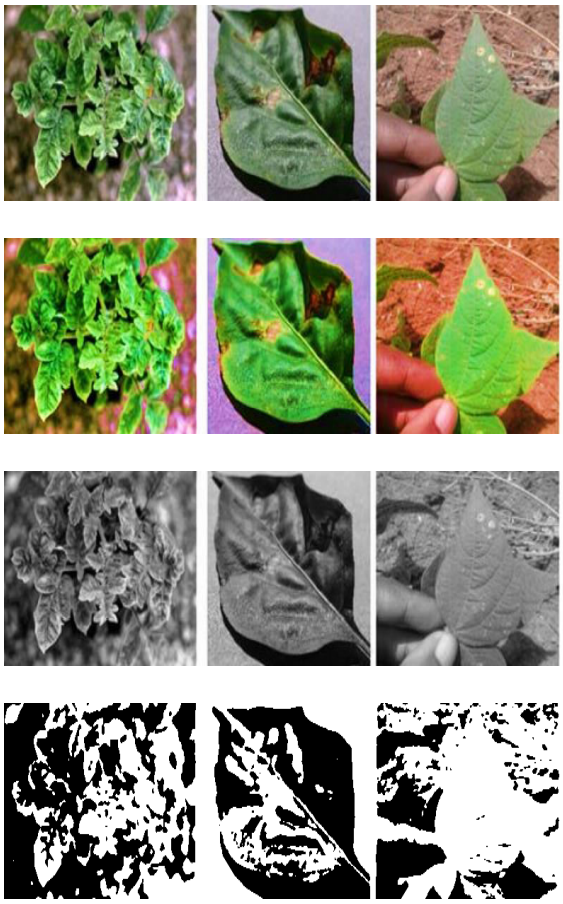


Fig.2. Sample Images and Classification Result

In order to create effective DL models, evaluating their performance is a crucial step. For the purpose of evaluating a model's efficacy on a specific dataset, many metrics are used. This gives us a chance to improve the model's efficiency by adjusting its hyper parameters. Making ensuring DL models can work with new, uncharted data is their main objective. A model's performance metrics may reveal how well it adapts to different datasets.

Since the 38 models share the same hyper parameters—the optimizer, the learning rate, the number of epochs, and the patience in early stopping—the poor accuracy is not hyper parameter dependent. The "Adam" optimizer was used with a learning rate of 0.001, 30 epochs, and 6 patience. The models' constant and dependable performance is guaranteed by avoiding the issues of excessive variance and overfitting, which arise when the set of data is trained on them just once.

By using several processing methods, including random rotation, changing, shearing, flipping, as well as zooming, among others, image augmentation generates fake training images. Another way of looking at it is that image augmentation is a data augmentation technique that uses current training samples to generate new training data pictures.

For this dataset, we used a modified version of the popular EfficientNet convolutional network. To enhance the overall result's accuracy, transfer learning was implemented using a pre-trained weight package for the EfficientNet architecture. The model's pre-trained weights file originated from its training on plant disease classification. Using a variety of optimisers and loss functions, we evaluated the produced model and found it to be accurate. This was accomplished by first training the model on a sample dataset consisting of 135 classes selected at random from the original dataset. After 20 iterations of optimizing these N classes for optimal accuracy, we choose the best combination and applied it to the full dataset. The installation of the Cuda along with CudNN libraries allowed the testing and training steps to run smoothly on the GPU. We ran the dataset across 20 epochs after transferring it to a Solid State Drive.

4.2. Performance Evaluation

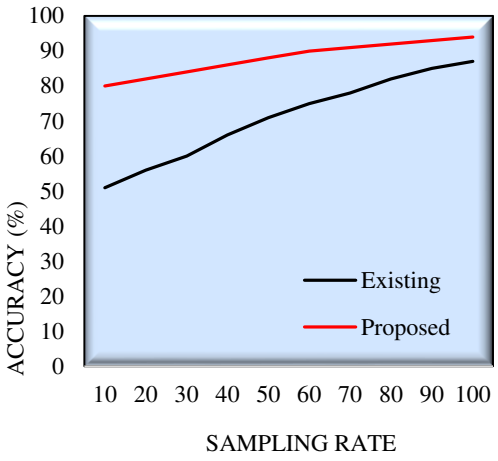
Although asymmetries in the input data might make accuracy too dependent on the accuracy of a single class, accuracy still assesses how right the classification is overall. While recall measures how well the model finds all relevant instances, precision measures how well the model accurately identifies the relevant items.

Their reliance on individual courses is reduced by averaging these indicators. The goal of the f1 score is to make sure that high recall or precision values don't make the accuracy exaggerated. This is the sweet spot when accuracy and memory harmonize.

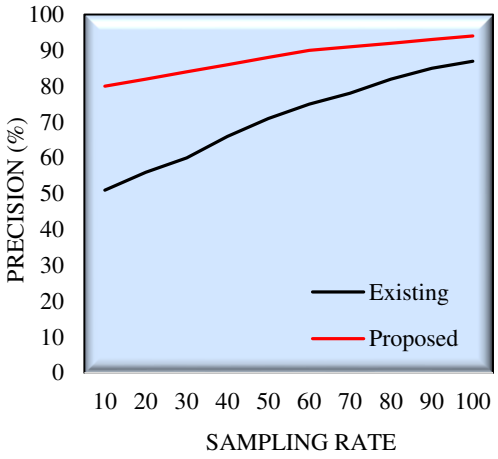
In order to measure the effectiveness of picture detection tasks for certain species, IoU computes the proportion of overlapping areas to total area. As measures for evaluating the model, we used precision, recall, accuracy, and F1 score. An F1 score indicates how well a classification model performs. Accumulated model Precision and Recall are what make up this average. Equations (23) provide the formulae for determining the F1 score, Accuracy, Recall, and Precision.

$$\begin{aligned} \text{Accuracy} &= \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \times 100\% \\ \text{Recall} &= \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \times 100\% \\ \text{Precision} &= \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \times 100\% \\ 11 &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100\% \end{aligned} \quad (23)$$

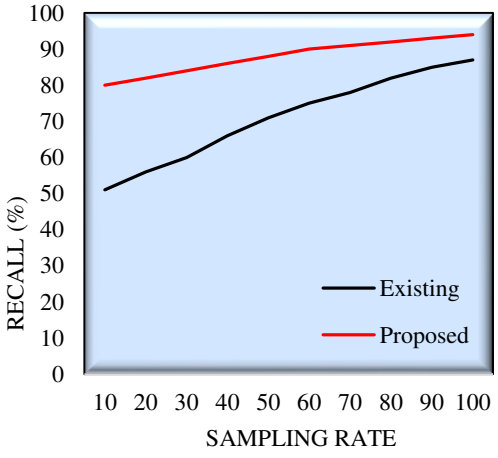
We offered many trials to back up our model selection and ensure experiment repeatability with other datasets in the same area. This allowed us to conduct an extensive evaluation and error comparison amongst each model in this dataset. The confusion matrix, ROC curve, error in validation per model, and Top-K table are some of the visualizations that we plot. Recall, accuracy, and precision are the metrics we use to evaluate the results.



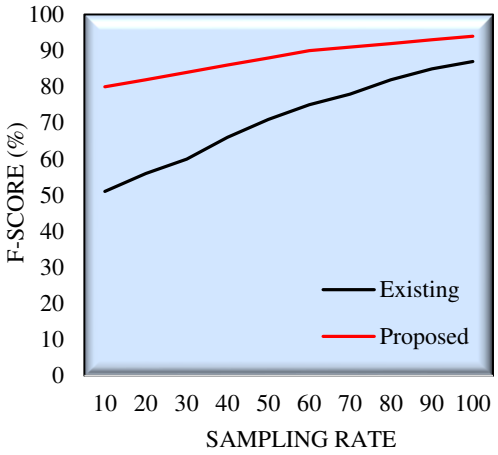
a. Accuracy



(b) Precision



(c) Recall



(d) F-score

Fig.3 Comparative Analysis

Table 1. Performance of different improvements dataset

Dataset	Added Modules	Accuracy	Recall	F1
Data Augmentation	ResNet-50	98.81%	98.16%	98.17%
	Xception	98.92%	98.11%	98.10%
	DenseNet201	**99.35%**	**98.51%**	**98.62%**
	Proposed	**99.55**	**99.85**	**99.89**
Without Data Augmentation	ResNet-50	95.29%	96.02%	95.43%
	Xception	95.17%	95.81%	95.1%
	DenseNet201	**95.61%**	**96.20%**	**95.52%**
	Proposed	**95.89**	**97.52**	**97.96**

With an order of magnitude reduction in parameter size as well as FLOPS (floating-point operations per second), EfficientNet outperformed conventional CNNs in terms of accuracy and efficiency. As an example, under the high-accuracy regime of Agri-ImageNet, the EfficientNet-B7 model attained a state-of-the-art 84.4% top-1 / 97.1% top-5 accuracy. Compared to earlier CNN models, it was 8.4 times smaller and 6.1 times quicker on CPU inference. Figure 3 clearly shows that the suggested SegCaps outperforms the other approaches in terms of accuracy (up to 94 percent), precision (up to 93 percent), and recall (up to 93 percent).

IV. CONCLUSION

This study introduces a transfer learning strategy for plant species identification, with a focus on the most prevalent ailments affecting sunflower and cauliflower plants. All of the current methods for transfer learning in keras are compared to one another using a variety of performance criteria so that the classification accuracy for a given dataset may be determined.

We test 38 learning transfer models on the Sunflower, Cauliflower, with Agri-Imagenet datasets, with a maximum of ten to sixteen rapid stops and six patience levels. Agri-Imagenet is a subset of the ImageNet dataset that includes all plant-related categories. Factors such as the Agri-ImageNet dataset's properties, the deep architecture, the innovative design, and the probable inadequate fine-tuning might explain the lackluster results achieved with VGG-16, VGG-19, Inception, a NasNet, and many ResNet models. For the Agri-ImageNet dataset, these models may perform less well than others because they have trouble capturing the unique characteristics and patterns associated with plant diseases. To pinpoint the reasons for their poor showing on the Agri-ImageNet dataset, more research and analysis is required. Results from an extensive evaluation of 38 transfer learning models on three different datasets (Sunflower, Cauliflower, and Agri-imagenet) demonstrate that model complexity, overfitting, data set, reliability, and overactive variables are the most important factors in determining a model's ability to detect plant diseases. Analysis of the model's dependence on these factors is necessary for determining the optimal transfer learning model. Since EfficientNetV2B2 and EfficientNetV2B3 are acceptable in all of the aforementioned qualities, they are the best models to be considered for use in throughput prediction.

REFERENCES

[1]. Omeer, A.A., &Deshmukh, R.R. (2021). Deep Learning-Based Models for Classification of Invasive Plant Species from Hyperspectral Remotely Sensed Data. Proceedings of the International Conference on Data Science, Machine Learning and Artificial Intelligence.

[2]. Hamuda, E., Parsi, A., Glavin, M., & Jones, E. (2022). Comparison of Support Vector Machines and Deep Learning for Plant Classification in Smart Agriculture Applications. Signal Processing and Vision.

[3]. Raj, A.P., &Vajravelu, S.K. (2019). DDLA: dual deep learning architecture for classification of plant species. IET Image Process., 13, 2176-2182.

[4]. Anantharaman, K., Divyasri, K., & Rani, S.V. (2022). Plant Species Identification using Probability Tree Approach of Deep Learning Models. Conference and Labs of the Evaluation Forum.

[5]. Barhate, D., Pathak, S., Dubey, A.K., &Nemade, V. (2022). Cohort study on recognition of plant species using Deep Learning methods. Journal of Physics: Conference Series, 2273.

[6]. Kanda, P.S., Xia, K., &Sanusi, O.H. (2021). A Deep Learning-Based Recognition Technique for Plant Leaf Classification. IEEE Access, PP, 1-1.

[7]. Showrav, T.T., Bain, S., Hossain, M., Ahmed, K.I., Fattah, S.A., & Ahmed, S. (2022). A Two-stage Approach for Plant Disease Classification Based on Deep Neural Networks and Transfer Learning. 2022 12th International Conference on Electrical and Computer Engineering (ICECE), 469-472.

[8]. S.Gawli, K., & Gaikwad, A.S. (2020). Deep Learning for Plant Species Classification. Journal of emerging technologies and innovative research.

[9]. Tan, J.W., Chang, S., Abdul-kareem, S., Yap, H.J., & Yong, K. (2020). Deep Learning for Plant Species Classification Using Leaf Vein Morphometric. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 17, 82-90.

[10]. Sunny, N., &Ktu (2020). A Review on Deep Learning for Plant Species Classification using Leaf Vein. International Journal of Engineering Research and, 9.

[11]. Thomkaew, J., &Intakosum, S. (2023). Plant Species Classification Using Leaf Edge Feature Combination with Morphological Transformations and SIFT Key Point. Journal of Image and Graphics.
[12]. B. R., P., & P, L. (2022). Deep Learning Model for Plant Species Classification Using Leaf Vein Features. 2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), 238-243.
[13]. Wani, S.F., Ashraf, A., &Sophian, A. (2022). Image-based disease detection and classification in Indian apple plant species by using deep learning. Applied Research and Smart Technology (ARSTech).
[14]. Reda, M., Suwwan, R., Alkafri, S., Rashed, Y., &Shanableh, T. (2022). AgroAId: A Mobile App System for Visual Classification of Plant Species and Diseases Using Deep Learning and TensorFlow Lite. Informatics, 9, 55.
[15]. Kumar, P., Danage, J., Desale, P., &Bhovi, B. (2020). Deep Learning for Plant Species & Disease Identification.