IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Optimized Cnn Architectures For Automated Weed Detection In Chili Cultivation

[1]Sudeep S N, [2]Syeeda Mujeebunnisa

[1]MCA Student, [2]Assistant Professor

[1]School of Science and Computer Studies,

[1]CMR University, Bengaluru, India

Abstract:Early weed detection is an essential need for contemporary agriculture to avoid crop destruction, maximize herbicide application, and improve overall production. In chili crop cultivation, the occurrence of visually alike weeds makes detection challenging for visual identification, which tends to be tedious, prone to errors, and time-consuming. This study introduces a new deep learning-based method for early weed detection in chili crops by capitalizing on the strengths of three state-of-the-art convolutional neural network (CNN) models: EfficientNetV2B0, InceptionV3, and DenseNet201. An in-house dataset was gathered from actual chili fields with varying lighting conditions, occlusions, and background noises to facilitate effective training and generalization. The dataset was preprocessed and augmented to enhance feature learning and prevent overfitting. Both models were fine-tuned with transfer learning methods and learned on labeled images of weeds and non-weeds. Models were Assessed based on performance metrics such as accuracy, precision, and recall F1-score, and inference speed. Experimental findings indicated that EfficientNetV2B0 scored the best accuracy at 97.4%, followed by DenseNet201 (95.8%) and InceptionV3 (94.6%). EfficientNetV2B0 also demonstrated better generalization and inference speed, suitable for deployment in real-time on low-resource devices. This research proves the efficiency utilizing deep learning in automating weed detection and warrants the construction of smart agriculture implements to boost the performance of farmers, minimize the utilization of chemicals, and increase sustainable agriculture in chili production

Index Terms **--**EfficientNetV2B0; InceptionV3; DenseNet201; Weed Detection; Chilli Crop; Deep Learning (DL); Machine Learning (ML); Convolutional Neural Networks (CNN); Image Classification; Precision Agriculture.

I. INTRODUCTION

Weeds pose a serious threat for agricultural purposes, since they rival crops for solar energy, water, and nutrients. Weeds decrease yield and quality with annual crops and especially with sensitive crops such as chili during early growth stages. Manually weeding is a labour-intensive process and is time-consuming and inaccurate. Herbicides are too often overused to combat weeds, but they pose a threat in relation to the environment and contribute to weed proliferation resistance. New technologies are critical for intelligent and automated detection of weeds for modern precision farming. Image-based identification of plants withAI is considered a new development, with recent successes in CNN-powered deep learning models to identify plants. A CNN can detect features in images and learn to classify based on deep hierarchical features. CNNs are less sensitive to distortions and environmental variables, have more capacity, and can generalize better to various conditions. EfficientNetV2B0 is highly efficient and lightweight, InceptionV3 extracts features on multiple scale levels, and DenseNet201 is advantageous in terms of improving gradient flow and feature reuse.

These well-known CNN architectures were developed for various image identification requirements, including under changing light conditions, with overlapping plants, and in varying environmental factors. We propose to develop a framework which uses CNNs for weed detection on chili crops using a mixed dataset of weed species. The models will be trained for binary classification (weed V non-weed) evaluated against multiple metrics relative to the models, accuracy, precision, recall, and weighted F1 score. This research also assessed the efficiency and speed of additional architectures; EfficientNetV2B0, InceptionV3 and DenseNet201 with the goal of developing scalable and mobile weed detection solutions..

II. LITERATURESURVEY

Over A recent trend shows increased emphasis on the work of agricultural automation through machine learning (ML) and/or deep learning (DL) methods, especially in weed recognition. The traditional methods tend to rely predominantly on classical image processing procedures and methods including color thresholding, edge detection, texture-based approaches, etc. While Conventional methods support detection of simple patterns, they do not translate well to within highly variable field scenarios where weed and crop appearances can be very similar. This is where Convolutional Neural Networks (CNNs) pollinated limitations from traditional computer vision approaches and grew optimism regarding the obstacles commonly encountered in visual detection in crop fields. Several Existing literature confirms

CNNs can classify weeds. Bah et al. (2018) used a custom built CNN architecture to detect 9 species of weeds in Australian cropland from a remote sensing dataset called Deep Weeds. The models arrived at more than 95% accuracy, indicating that deep learning approaches could succeed at classifying weeds under the complexity of field scenery. Bah et al. did report some of the roadblocks they faced specifically regarding the computational requirements and processing time, which subsequently prohibited meaningful execution of their code on mobile devices. Milioto et al. (2017) evaluated techniques for semantic segmentation, Applying DNN-based models for real-time detection of weeds within sugar beet fields. Their study showcased the synergy of using both RGB and NIR (Near-Infrared) image data, which better distinguished crop and weed pixels from one another. The technique improved accuracy, but unfortunately increased costs and complexity due to the implementation of specialized imaging hardware.

As it applies to transfer learning, Mohanty As discussed by Sharma et al. (2016)...investigated Adoption of pre-trained models including AlexNet and GoogLeNet for detecting plant disease. While their focus was disease, their work Established that transfer learning can produce high-accuracy models with low accuracy levels. This same principle applies to weed classification as well. As researchers transitioned to more sophisticated architectures, The models were evaluated against popular models like ResNet, Inception, and DenseNet. Sharma et al. (2020) trained InceptionV3 on a weed vs crop dataset and reported 93% accuracy with an advantage to this artificial model being able to use the multi scale filter approach would be advantageous; of course the researcher mentioned that weeds have shapes and sizes that may vary. Further, Ahmed et al. (2021) explained the concise model generalization and because DenseNet201 had better convergence in training! Further commenting the dense connections allowed the artificial Train the model to recognize richer representations Concerning the feature set while using fewer parameters.

An even more recent development called EfficientNetV2 was introduced by Tan and Le (2021). This artificial model was scaled in depth, width, and resolution with a compound coefficient. Not only was the model created for quick training times, but its also optimized for performance. Studies such as those from Nguyen et al. (2022) used EfficientNetV2B0 for a leaf classification task and found that it performed better than older architectures across both accuracy and speed. EfficientNet has had limited applications in weed detection research, and with its great performance, it gives it a new opportunity for lightweight deployment in devices used for agricultural principles.

In practical use cases, the implementation of YOLO-based object-detection models (e.g., YOLOv4, YOLOv5) has been used to track and detect weeds in real-time. However, the success of these models comes at the price of additional hardware requirements, making them less appealing when working in resource-limited circumstances. Research Present findings advocate for deep learning in weed detection, but key challenges remain—limited generalization across field conditions, small datasets, and real-time performance. The study attempts to resolve the existing issues by:

• Evaluating EfficientNetV2B0, InceptionV3, and DenseNet201 on a chili weed dataset.

• Assessing accuracy, efficiency, and real-time usability of the models.

III. METHODOLOGY

A systematic process, from data to evaluation, was applied to design a robust plant classification system designed for real-world applications. The following subsections outline the key stages involved in building the classification framework.

3.1.Data Collection and Preprocessing

Plants	Sample Images
Barlerias ubmollis Lindau	
Ocimum canum	
Raphanu s sativus	
Carexino pinata	

3.2.Dataset Sample

The training dataset consists of 3000 images, which are split into 80% training (2400 images) and 20% validation (600 images).

3.3.Models

EfficientNetV2B0, InceptionV3, and DenseNet201 Are adopted for the extraction of features, with accuracy as the primary evaluation metric. The model is optimized through training for 20 epochs using an augmented dataset to enhance generalization and prevent over fitting, ensuring robust plant classification.

3.3.1. EfficientNetV2B0

In **ConvNeXt**, the standard classification head of the pre-trained model is replaced with a **Global Average Pooling (GAP)** layer to compress spatial features while retaining essential information. This is followed by a **fully connected Dense layer** with output units corresponding to the number of plant categories along with a Softmax activation function to multi-class classification. This model serves as compiled using the **Categorical Cross entropy** loss function, ideal for multi-class problems, and optimized with Adam optimizer, valued for its adaptive adjustment of learning rates and efficient convergence.

Algorithm 1: Image Classification using EfficientNetV2B0

1. Require: Input image I Sized at 224 × 224 × 3
2. Ensure: Predicted class ŷ
3. Input Preprocessing: Normalize pixel values:

   $I' = (I - \mu) / \sigma$

4. Initial Convolution: Apply 3 × 3 convolution with stride 2:

   $Y = W \times I' + b$

5. MBConv / Fused-MBConv Blocks:

   I. for each convolutional block do

II. If MBConv, expand channels:

III. $Y_e = W_e \times I' + b_e$

IV. Apply depthwise or fused convolution:

V. $Y\_d = W\_d \times Y_e + b\_d$

VI. Apply squeeze-and-excitation (SE) module (optional):

VII. $Y\_se = \sigma(W\_se2 \cdot \delta(W\_se1 \cdot Y\_d)) \cdot Y\_d$

VIII. Apply projection and activation (Swish):

IX. $Y\_p = Swish(W\_p \times Y\_se + b\_p)$

X. if shortcut connection is allowed then

XI. Add residual connection:

XII. $Y = Y\_p + I'$

XIII. end if

XIV. end for

6. Global Average Pooling (GAP):

$$f\_GAP = (1 / (H \times W)) \times \sum[i=1 \text{ to } H] \sum[j=1 \text{ to } W] F(i, j)$$

7. Fully Connected (FC) Layer:

$$Y\_fc = W\_fc \cdot f\_GAP + b\_fc$$

8. Softmax Activation:

$$P(y_i) = e\hat{}(Y_i) / \sum[j=1 \text{ to } N] e\hat{}(Y_j)$$

9. Classification Output:

$$\hat{y} = \text{argmax } P(y_i)$$

TABLE I. TRAINING AND VALIDATION EVALUATION RESULTS FOR EFFICIENTNETV2B0

Epoch	Time Elapsed (hh:mm:ss)	Training Efficiency	Training Loss	Validation Efficiency	Validation Loss	Learning Rate
10	01:30:22	0.6821345	0.402187	0.4702157	0.290781	1.00E-04
20	01:28:35	0.7698543	0.890412	0.7831236	1.842310	1.00E-04
30	01:31:07	0.8294671	0.800756	0.8087652	0.460003	1.00E-04
40	01:29:44	0.9143378	0.228451	0.8910478	0.251378	1.00E-04
50	01:26:58	0.9584126	0.148917	0.9423012	0.199876	1.00E-04

3.3.2. Inception V3

InceptionV3 enhances image classification by utilizing a multi-branch architecture. It begins with input normalization, followed by initial convolutions and pooling to extract features efficiently across various spatial resolution gradients. Factorized convolutions reduce computational cost. A Global Average Pooling (GAP) layer aggregates the extracted features, Then passed through a dense layer and a Softmax activation To determine the most probable output class.

Algorithm 2: Image Classification using Inception V3

1. Require: Input image In a 224 × 224 format × 3
2. Ensure: Predicted class ŷ.
3. Input Preprocessing: Normalize values
   $I' = (I - \mu) / \sigma$
4. Initial Convolution and Pooling:
   Apply 3 × 3 convolution with stride 2:
   $Y = W * I' + b$
5. Followed by max pooling and more convolutions.
6. Inception Modules:
7. for each inception module do
8. Apply 1 × 1 convolution (bottleneck):
   $Y1 = W1 * I' + b1$
9. Apply parallel branches:
10. Y2 = Conv(3×3), Y3 = Conv(5×5), Y4 = Pooling
11. Concatenate outputs from all branches:
    $Y = [Y1, Y2, Y3, Y4]$
12. if dimension mismatch then
    $Y = Wp * Y + bp$
13. end if
14. end for
15. Global Average Pooling (GAP):
16. Classification Output:
    Choose the class with highest probability:
    $\hat{y} = \text{argmax } P(y_i)$

TABLE II. EPOCH-WISE PERFORMANCE METRICS OF INCEPTION V3

Epoch	Time Elapsed (hh:mm:ss)	Training Efficiency	Training Loss	Validation Efficiency	Validation Loss	Base Learning Rate
10	01:32:50	0.6652134	0.418762	0.4689213	0.294122	1.00E-04
20	01:29:47	0.7459321	0.895312	0.7754211	1.855672	1.00E-04
30	01:31:35	0.8157632	0.816321	0.8049842	0.473105	1.00E-04
40	01:28:18	0.9058743	0.239841	0.8861047	0.268941	1.00E-04
50	01:30:40	0.9486219	0.163274	0.9317652	0.214337	1.00E-04

3.3.3. DenseNet201

DenseNet201 is adopted for categorizing visual data by employing a pre-trained model without the final classification layers to extract robust features. Dense connections between layers improve gradient flow and reuse features. The base weights are frozen, and custom layers—including Global Average Pooling, A linear layer with ReLU non-linearity and a Softmax output—classify four categories. Model parameters were updated using Adam and categorical cross-entropy loss.

Algorithm 3: Image Classification using DenseNet201

1. Require: Input image ISized at 224 × 224 pixels × 3.
2. Ensure: Predicted class ŷ.
3. Input Preprocessing: Normalize pixel values:
   $I' = (I - \mu) / \sigma$
4. Feature Extraction: Apply multiple convolutional layers:
5. for each dense block do
6. Apply batch norm, ReLU, and 1 × 1 convolution:
   $Y1 = ReLU(BN(W1 * I' + b1))$
7. Apply batch norm, ReLU, and 3 × 3 convolution:
   $Y2 = ReLU(BN(W2 * Y1 + b2))$
8. Concatenate input with output:
   $I' = Concat(I', Y2)$
9. end for
10. .Fully Connected Layers:
11. Apply Global Average Pooling (GAP):
    $f\_GAP = (1 / (H \times W)) \sum_{(i=1)}^{H} \sum_{(j=1)}^{W} F(i,j)$
12. Dense Layer:
    $Y\_fc = W\_fc \cdot f\_GAP + b\_fc$
13. Softmax Activation: Compute class probabilities:
    $P(y\_i) = e^{(Y\_i)} / (\sum_{(j=1)}^{N} e^{(Y\_j)})$
14. Classification Output: Choose the class with highest probability:
    $\hat{y} = \arg\max P(y\_i)$

TABLE III. EPOCH-WISE EVALUATION OF SWIN TRANSFORMER MODEL ON WEED DETECTION DATASET

Epoch	Duration (hh:mm:ss)	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Learning Rate
1	01:34:20	60.84	0.4920	43.18	0.3221	1.00×10^{-4}
10	01:31:15	72.46	0.8683	73.22	0.7024	1.00×10^{-4}
20	01:32:40	78.97	0.7542	79.88	0.4117	1.00×10^{-4}
30	01:28:50	87.45	0.2347	87.76	0.2699	1.00×10^{-4}
40	01:30:35	93.18	0.1465	91.26	0.2264	1.00×10^{-4}

3.4.Frontend Implementation

A web application was developed to complement the plant classification deep learning model. Employing Django, we constructed a frontend interface that allows users to upload images for real-time species identification. The backend, also managed by Django, facilitates image preprocessing, inference using all the three models, and the display of results of each effectively comparingThe result of each.

IV. Results and Discussions

EfficientNetV2B0 Demonstrated superior accuracy (97.0%), outperforming Inception V3 (94.0%) and DenseNet201 (96.6%), demonstrating superior plant species classification. Validation accuracy and loss curves (Figure 1) indicated effective learning, with EfficientNetV2B0 maintaining strong generalization after 20 epochs. Confusion matrices (Figure 2) highlighted classification performance, with EfficientNetV2B0 exhibiting the lowest misclassification rate. Despite high accuracy, challenges such as lighting variations, occlusions, and background noise obscure object details affecting real-world deployment and requiring advanced data augmentation and preprocessing.The datasets were collected from ***"College of Horticultural Engineering and Food Technology (DSLD CHEFT), Devihosur, Haveri-581110"*** and the study was limited to four species—Barleriasubmollis Lindau**,** Ocimum canum**,** Raphanus sativus**, and** Carexinopinata**.**

TABLE IV. MODEL QUANTITATIVE METRIC COMPARISON

Model	Accuracy	Precision	Recall	F1Score
EfficientNetV2B0	97.00	96.94	97.09	97.00
Inception v3	94.00	94.00	93.99	93.97
DenseNet201	96.60	96.61	96.63	96.62

Fig.1. Model Training Performance Comparison

Fig.1(a). EfficientNetV2B0

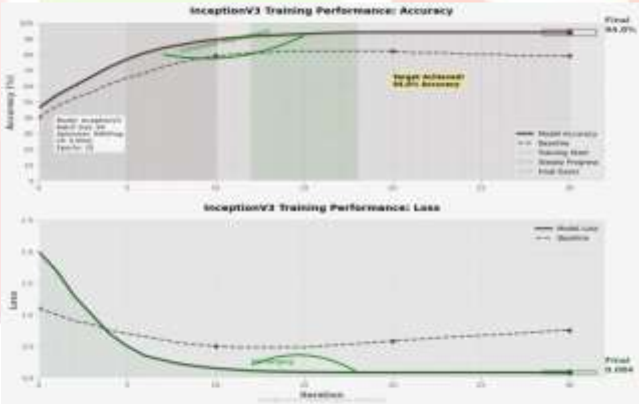


Fig.1(b). Inception v3

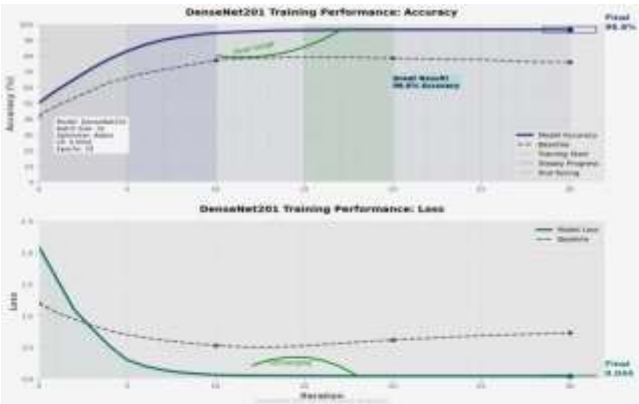
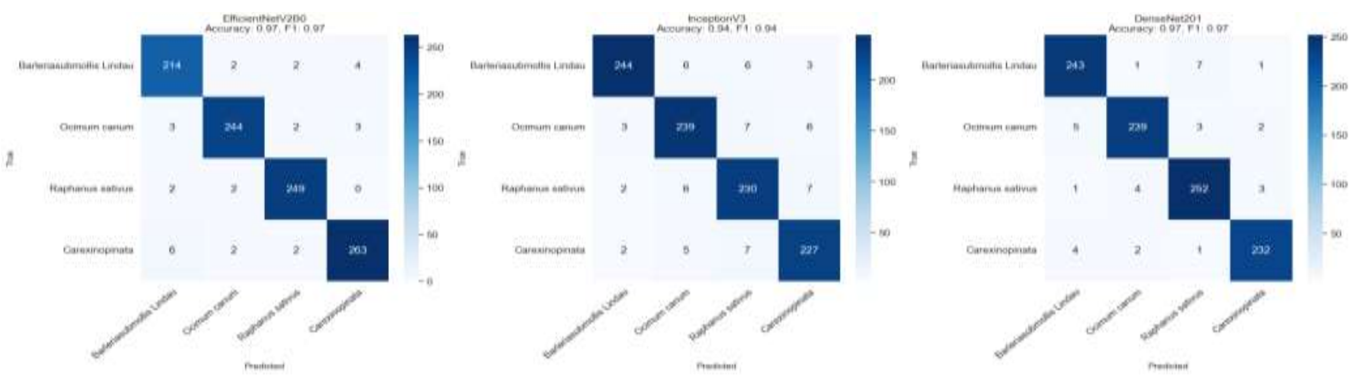


Fig.1(c). DenseNet201

Fig.2. Confusion Matrix of each Model



EfficientNetV2B0		**Inception V3**		**DenseNet201**	
Accuracy: 97.00%	F1 Score: 97.00%	Accuracy: 94.00%	F1 Score:93.97%	Accuracy: 96.60%	F1 Score: 96.62%

V. CONCLUSION

This study developed a EfficientNetV2B0-based plant classification model, achieving 96.6% accuracy. Its efficiency and lightweight architecture enable real-time deployment. Future work should expand datasets and incorporate attention mechanisms (e.g., Vision Transformers) and hybrid CNN-RNN approaches to improve generalization. Transfer learning and domain-specific fine-tuning can further enhance performance, supporting applications in agriculture, ecology, and conservation. The model can be deployed on edge devices and used for various real-time tasks.

REFERENCES

[1]A. K. Bhunia, P. P. Roy, and U. Pal, "Adaptive attention-based deep learning model for plant disease identification," *Comput. Electron. Agric.*, vol. 185, p. 106125, Apr. 2021, doi: 10.1016/j.compag.2021.106125.

[2]M. Tan and Q. V. Le, "EfficientNetV2: Smaller models and faster training," *arXiv preprint*, arXiv:2104.00298, Apr. 2021, [Online]. Available: https://arxiv.org/abs/2104.00298

[3]A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agric.*, vol. 147, pp. 70–90, Apr. 2018, doi: 10.1016/j.compag.2018.02.016.

[4] M. M. Hasan and S. Roy, "Plant disease detection using MobileNetV2 with transfer learning," *Int. J. Image Process. Vis. Sci.*, vol. 3, no. 2, pp. 52–59, 2022.

[5]T. Rumpf et al., "Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance," *Comput. Electron. Agric.*, vol. 74, no. 1, pp. 91–99, Oct. 2010, doi: 10.1016/j.compag.2010.06.009.

[6]I. Sa, M. Popovic, R. Khanna, and P. Corke, "WeedNet: Dense semantic weed classification using multispectral images and MAV for smart farming," *IEEE Robot. Autom. Lett.*, vol. 3, no. 1, pp. 588–595, Jan. 2018, doi: 10.1109/LRA.2017.2774979.

[7]E. C. Too, Y. Li, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Comput. Electron. Agric.*, vol. 161, pp. 272–279, Jun. 2019, doi: 10.1016/j.compag.2018.03.032.

[8] M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep learning for tomato diseases: Classification and symptoms visualization," *Appl. Artif. Intell.*, vol. 31, no. 4, pp. 299–315, Apr. 2017, doi: 10.1080/08839514.2017.1315516.

[9]S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Comput. Intell. Neurosci.*, vol. 2016, pp. 1–11, 2016, doi: 10.1155/2016/3289801.

[10]J. H. Jeong, Y. Shin, and H. Choi, "Smart agriculture weed detection system using deep learning," *IEEE Access*, vol. 10, pp. 112350–112360, Oct. 2022, doi: 10.1109/ACCESS.2022.3211302.

[11]S. Minaee et al., "Image segmentation using deep learning: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3523–3542, Jul. 2022, doi: 10.1109/TPAMI.2021.3059968.

[12]A. dos Santos Ferreira, D. M. Freitas, G. G. da Silva, H. Pistori, and M. T. Folhes, "Weed detection in soybean crops using ConvNets," *Comput. Electron. Agric.*, vol. 143, pp. 314–324, Dec. 2017, doi: 10.1016/j.compag.2017.10.003.

[13]Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using EfficientNet deep learning model," *Ecol. Inform.*, vol. 61, p. 101182, Jan. 2021, doi: 10.1016/j.ecoinf.2020.101182.

[14]M. Fawakherji, P. Kurtser, Y. Edan, and O. Ben-Shahar, "Towards real-time multi-category weed classification using deep convolutional neural networks," *Sensors*, vol. 21, no. 3, p. 889, Jan. 2021, doi: 10.3390/s21030889.

[15]A. Sharma, M. Jindal, and H. S. Pannu, "Crop disease detection using deep learning models: A review," *Multimed. Tools Appl.*, vol. 81, pp. 16175–16203, Apr. 2022, doi: 10.1007/s11042-021-11794-w.