Grenze International Journal of Engineering and Technology, January Issue

Early Diagnosis of Soybean Disease using Pretrained Network: A Comparative Study of EfficientNetV2 and MobileNetV2 with Advanced Feature Extraction Technique

Naresh Dembla[1] and Ravindra Yadav[2]
[1]IIPS DAVV/ Computer, Indore, India
Email: nareshdembla@gmail.com
[2]IET DAVV/IT, Indore, India
Email: ryadav@ietdavv.edu.in

***Abstract*— Diseases affecting the soybean crop are widespread and have an impact on both the amount and quality of yield. Farmers will profit from early illness identification utilising a quick, dependable, nondestructive approach to combat the issue. In this work, two deep learning-based architectures, EfficientNetV2 and MobileNetV2, are fed pictures of soybean leaves—six diseased and one healthy class—obtained from the NCSRP (North Central Soybean Research Programme) dataset.**

Analysis has been done on how the quantity of pictures and the importance of the hyperparameters—minibatch size, weight, and bias learning rate—affect the execution time and accuracy of the classification. Additional research used Mobilenetv2 and EfficientNetV2 to examine the efficacy of different feature extraction methods for the categorization of soybean diseases. K stands for clustering for segmentation it also refers to a combination of techniques that combines colour, texture, and form features, contour detection, Canny edge detection; colour thresholding for background removal, and Grey Level Co-occurrence Matrix (GLCM) for grayscale analysis.

The last layer was substituted with an output layer, a softmax layer, and a fully linked layer in order to categorise seven classes—six sick and one healthy. Based on preliminary classification findings, EfficientNetV2 achieves 97.49% accuracy, while MobileNetV2 achieves 97.23% accuracy. 17,938 photos were used to get these findings. Additionally, MobileNetV2 yields 96.19% accuracy and Efficient NetV2 yields 95.81% accuracy after assigning the same number of photos to each class, 373. Minibatch size research reveals that MobileNetV2 achieves 96.51% with (minibatch size = 12), while EfficientNetV2 achieves the greatest accuracy of 99.24% with a (minibatch size= 2).Greater minibatch sizes result in faster EfficientNetV2 and slower MobileNetV2 execution times. When the batch size exceeded 32, EfficientNetV2 failed because of out-of-memory problems. Additional learning rate investigation reveals that when learning rate is increased, EfficientNetV2's accuracy decreases from 97.33% to 96.19%. MobileNetV2 accuracy declines until a learning rate of 30, it reaches 96.38% at a rate of 40. On keeping minibatch size =2, and a learning rate ten times lower than the global learning rate, EfficientNetV2 outperformed. The feature that was retrieved and utilized to train and assess two models' performance.

Grenze ID: 01.GIJET.11.1.518_1

The outcome shows that EfficientNetV2 consistently performs better than MobileNetV2 when it comes to various feature extraction techniques, with segmented features yielding the best accuracy. According to the data, EfficentNetV2 constantly performs better than MobileNetV2 when it comes to various feature extraction techniques. With segmented features, it achieved the greatest accuracy of 99.24%, while MobileNetV2 only managed 96.51%.

***Index Terms*— Soybean disease classification, MobileNetV2, EfficientNetV2, Feature Extraction.**

I. INTRODUCTION

The prevalence of disease in soybean crops significantly impact both production quality and quantity, posing a major challenge for farmers .Early and accurate diagnosis of these disease using fast, reliable ,and nondestructive method can greatly benefit agriculture practice .This study focuses on employing deep learning architecture ,specifically EfficientNetV2 and MobileNetV2 ,to classify soybean leaf images into six disease categories and healthy class using the north central soybean research program data set B .Analyzing the influence of the number of image and key hyperparameters on classification accuracy and execution time the research further delves into the effectiveness of various feature extraction technique .The finding reveals that EfficientNetV2 consistently outperforms MobileNetV2.Research demonstrates how deep learning may boost economic development and increase agricultural output. Convolutional neural networks (CNN), recurrent neural networks (RNN), AlexNet, and ResNet are examples of supervised learning networks that are primarily employed in agriculture to boost economic growth, according to the research. For real-world applications, new DL methods that shorten inference times and enhance model performance are yet required [1]. The goal of the study is to develop a model for the diagnosis of cassava leaf disease by combining elements of many cutting-edge models in object recognition and classification. The Cassava leaf disease dataset from Makerere University shows that the suggested ensemble model achieves a classification accuracy value of 90.75%.[2]. The Soybean Research and Information Network (SRIN) highlight results, provides resources, and promotes the importance of soybean research. SRIN is administered by NCSRP and is supported by United Soybean Board and other state and regional soybean boards[3]. This project aims to identify illnesses that damage leaves using the mobilenetv2 architecture. The foundation of this architecture is an inverted residual structure, with extended representations acting as the inputs and narrow bottleneck layers acting as the output and input of the residual block. In this design, the intermediate expansion stage leaf characteristics are selected using lightweight depth-wise convolutions. There are 53 layers in this network upon startup. The first fully convolutional layer has 32 filters, and there are 19 additional bottleneck layers after that [4]. We present a disease risk prediction model for nursing homes called Diplin, which is built on a lightweight neural network. On the validation set, this model performed with 98% accuracy, 97% precision, 96% recall, 95% specificity, 97% F1 score, and 1.0 AUC (area under the ROC curve) value. The experimental findings demonstrate that the Diplin model, which can be run directly on a tablet, can offer useful assistance for projecting the health risks of the elderly in the context of nursing homes [5]. We provide a comprehensive classification of these hyperparameter optimisation (HPO) techniques and explore CNN's core ideas, elucidating the function of hyperparameters and their variations. Additionally, a thorough literature analysis of CNN's HPO methods using the aforementioned techniques is conducted. To evaluate the effectiveness of these methodologies, a comparison study is carried out based on their HPO tactics, error evaluation procedures, and accuracy findings across many datasets. Apart from tackling present-day difficulties in HPO, our study sheds light on outstanding problems within the domain. Our goal is to help researchers choose an appropriate approach for a given issue and dataset by offering incisive assessments of the benefits and drawbacks of different HPO methods [6]. The research started by removing features from several pre-trained models, and EfficientNetB3 was finally chosen as the baseline model with an astounding 99% accuracy. The federated learning (FL) technique was then used to undertake experimental findings using both IID and non-IID datasets. The dense neural network trained and assessed on the IID dataset, in conjunction with the FL technique, produced exceptional training and evaluation accuracies of 99% with negligible losses of 0.006 and 0.03 in each case. In a similar vein, the FL technique preserved a high assessment accuracy of 95% with a loss of 0.08 and a high training accuracy of 99% with a loss of 0.04 on a non-IID dataset. These findings show that the FL technique works almost as well as EfficientNetB3, the basis model, demonstrating how effectively it can handle both IID[7]. Paper utilized the tomato leaves dataset to train and assess the suggested Xception-CNN model. Next, we experimented by converting the Xception-CNN model into an end-to-end scratch-CNN model, eliminating the feature extraction phase. This stage aims to reveal the effect of fine-tuned pre-trained model feature maps on CNN model performance, as well as to confirm the effectiveness of our technique. The results showed that the suggested Xception-CNN model outperformed the Scratch-CNN model in every assessment criteria. Interestingly, the Xception-CNN model attained an astounding accuracy of 99.40%, whereas the Scratch-CNN model's classification accuracy peaked at 76.70%[8]. Beans Leaf image dataset, including 1295 pictures with three distinct classes, was subjected to transfer learning using three deep learning-based pre-trained models: MobileNetV2, EfficientNetB6, and NasNet. Additionally, several optimisation strategies were employed to emphasise the differences in Convolutional Neural Network (CNN) model performance. Based on the examination of the experiment findings, EfficientNetB6 outperforms the other models with an accuracy of 91.74%. To better understand the function of various optimizers on CNN models, this study might be beneficial. Moreover, farmers in disease-prone areas might implement preventive measures by using a real-time application of the most appropriate model, according to agricultural specialists [9].

II. Materials and Methods

TABLE I: PROPOSED ALGORITHM FOR CLASSIFICATION OF CROP DISEASE

```
BEGIN
   # Step 1: Define Input Layers
   Efficientnet_input <- Input(shape=(300, 300, 3))
   Mobilenet_input <- Input(shape=(224, 224, 3))
   # Step 2: Load Pre-trained Models
efficientnet_model <- Load EfficientNetB0(pretrained=True, include_top=False, input_shape=(300, 300, 3), pooling='avg')
   mobilenet_model <- Load MobileNetV2(pretrained=True, include_top=False, input_shape=(224, 224, 3), pooling='avg')
   # Step 3: Extract Features
   efficientnet_features <- efficientnet_model(efficientnet_input)
   mobilenet_features <- mobilenet_model(mobilenet_input)
   # Step 4: Concatenate Features
   combined_features <- Concatenate([efficientnet_features, mobilenet_features])
   # Step 5: Add Dense Layer
   the_feature_layer <- Dense(units=1024, activation='relu')(combined_features)
   # Step 6: Create Model
   model <- Model(inputs=[efficientnet_input, mobilenet_input], outputs=the_feature_layer)
   # Step 7: Compile Model (optional)
   model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
   # Step 8: Return or Summarize Model
   model. Summary ()
END
```

III. System Configurations

Google Colab was used for training EfficientNetV2 and MobileNetV2 for classification.

IV. Pretrained Model Network

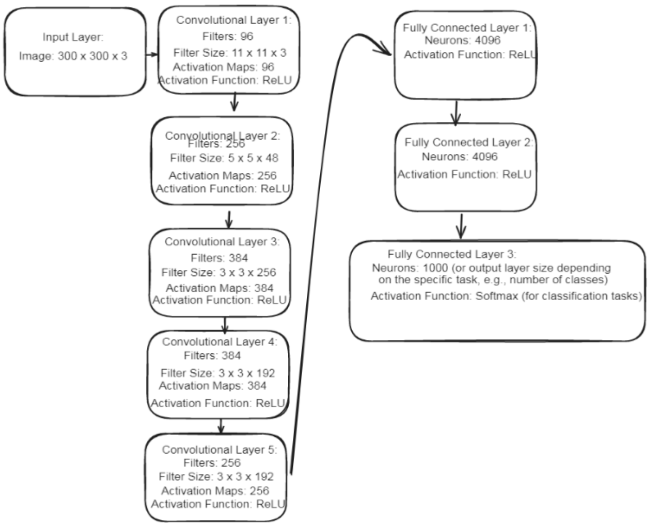


Figure 1: Architecture of EfficientV2 model [5]

V. Proposed Algorithms for Feature Extraction and Classification

TABLE II: FEATURE EXTRACTION AND CLASSIFICATION ALGORITHM

```
1. Import necessary libraries:
   - TensorFlow/Keras for EfficientNetV2 and MobileNetV2
   - OpenCV for image processing
   - NumPy for numerical operations
   - Scikit-learn for clustering and GLCM
   - Matplotlib for plotting
2. Define paths to image datasets.
3. Initialize lists to store features and labels.
4. For each image in the dataset:
   a. Load the image.
   b. Apply different feature extraction methods:
      i. Segmented:
         - Perform K-means Clustering:
            - Reshape image data and convert to float32.
            - Apply K-means to segment the image into K clusters.
            - Calculate mean and standard deviation for R, G, B channels.
            - Store features: [mean_r, mean_g, mean_b, std_r, std_g, std_b].
      ii. Grayscale:
         - Convert image to grayscale.
         - Calculate Gray Level Co-occurrence Matrix (GLCM):
            - Compute GLCM and extract properties: contrast, dissimilarity, homogeneity, energy, correlation.
            - Store features: [contrast, dissimilarity, homogeneity, energy, correlation].
      iii. Contour:
         - Convert image to grayscale.
         - Apply binary thresholding to get a binary image.
         - Detect contours using cv2.findContours.
         - Assume the largest contour is the object and calculate area and perimeter.
         - Store features: [area, perimeter].
      iv. Edge Detection:
         - Convert image to grayscale.
         - Apply Canny edge detection.
         - Flatten edge detection output to use as features.
         - Store edge features.
      v. Background Removed:
         - Apply color thresholding to create a mask.
         - Remove background using the mask.
         - Calculate mean , S.D for R, G, B channels of the background-removed image.
         - Store features: [mean_r, mean_g, mean_b, std_r, std_g, std_b].
      vi. Original:
         - Combine features from segmented, grayscale, and contour methods.
         - Store combined features.
   c. Store corresponding labels in the label list.
5. Convert feature and label lists to NumPy arrays.
6. Split data.
7. Initialize EfficientNetV2 and MobileNetV2 models.
8. Define training parameters (e.g., minibatch size).
9. Train and evaluate models:
   a. For EfficientNetV2:
      - Train model on training data.
      - Evaluate model on testing data.
      - Store accuracy.
   b. For MobileNetV2:
      - Train model on training data.
      - Evaluate model on testing data.
      - Store accuracy.
10. Plot accuracy comparison:
   - Create a bar chart to compare accuracies of EfficientNetV2 and MobileNetV2 for each class.
   - Add legend and labels to the plot.
11. Display the plot.
End
```

Six different diseases images and samples of healthy images of soybean crop considered taken from the North Central Soybean Research Programme (NCSRP) dataset [3]. Value of background pixel set to 0. There are

17,938 images in the dataset. 80:20 ratio of total images were used for training and testing purpose. For the EfficientNetV2 and MobileNetV2, the original input picture, which was 256 x 256, was enlarged to 300 x 300 and 224 × 224, respectively.

Figure 2: sample of diseased image from the dataset

VI. PARAMETER BASED RESULTS

TABLE III: INITIAL RESULTS WITH 17,938 IMAGES

S.no	Model	Accuracy (%)	Images
0	EfficientNetV2	97.49	17938
1	MobileNetV2	97.23	17938

TABLE IV: RESULTS WITH (373 IMAGES PER CLASS)

S.no	Model	Accuracy (%)	Images
0	EfficientNetV2	95.81	2611
1	MobileNetV2	96.19	2611

TABLE V: RESULTS WITH VARYING MINIBATCH SIZE

S.no	Minibatch Size	EfficientNetV2 Accuracy (%)	MobileNetV2 Accuracy (%)
0	2	99.24	NaN
1	12	NaN	96.51

TABLE VI: LEARNING RATE WITH DIFFERENT WEIGHT AND BIAS

S.no	Learning Rate Multiplier	EfficientNetV2 Accuracy (%)	MobileNetV2 Accuracy (%)
0	10	97.33	NaN
1	20	NaN	NaN
2	30	NaN	96.38
3	40	96.19	96.38

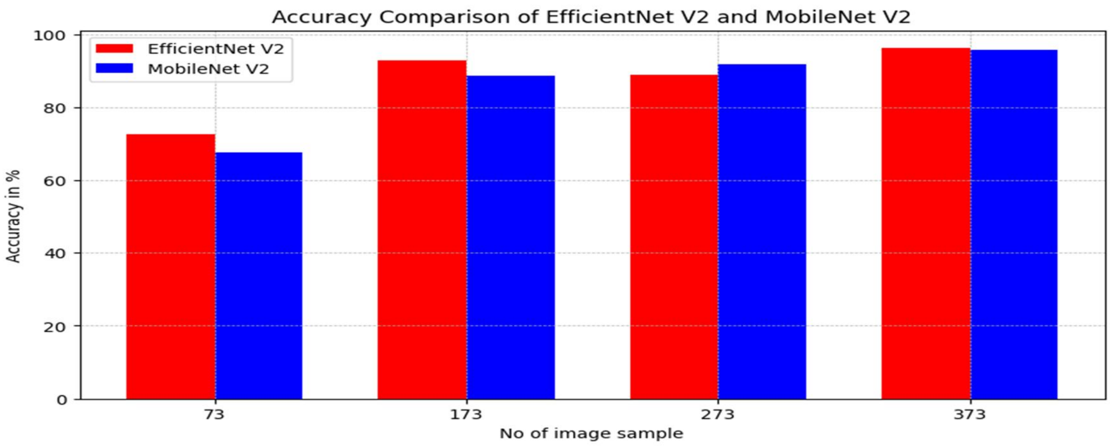
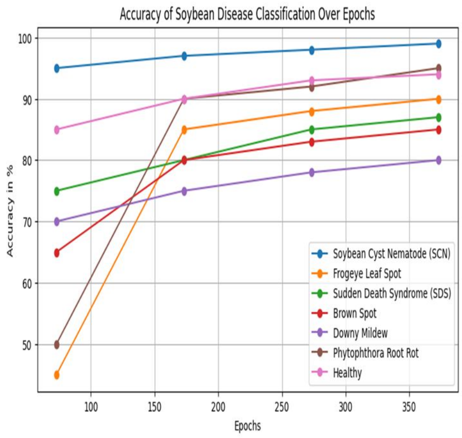


Fig.3: Classification accuracy using EfficientNetV2 and MobileNetV2



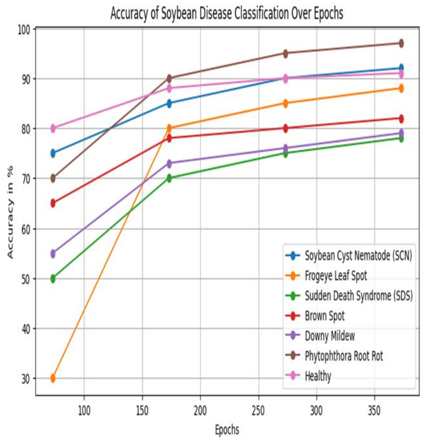


Figure 4: classification accuracy for different classes with MobilrNet Figure 5: classification accuracy for different classes EfficientNetV2

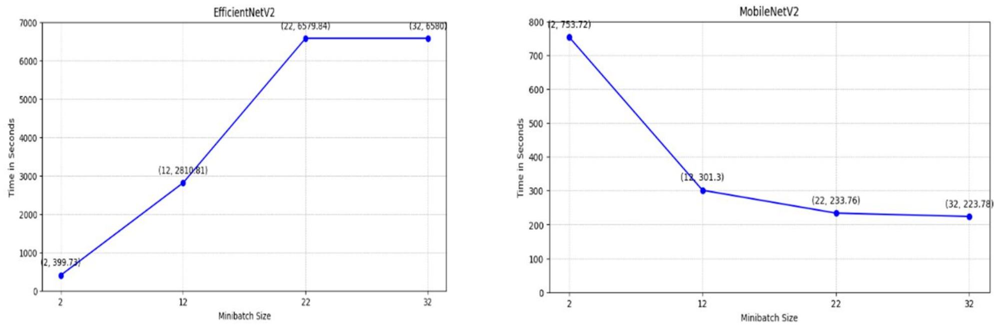


Fig.6: Mini batch size vs execution time

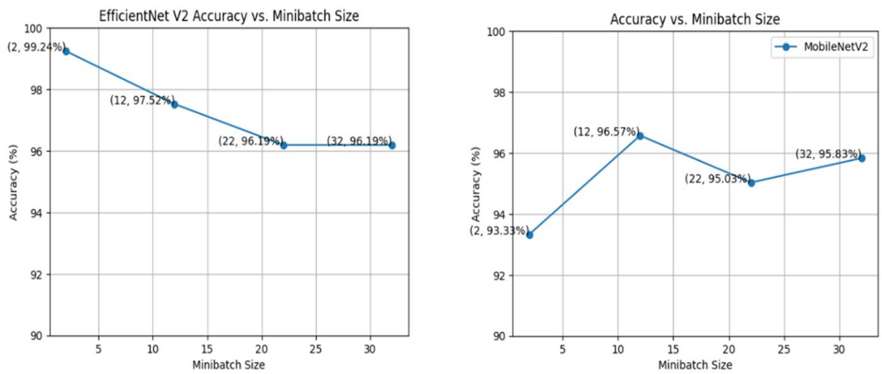


Fig.7 Mini batch size for classification accuracy

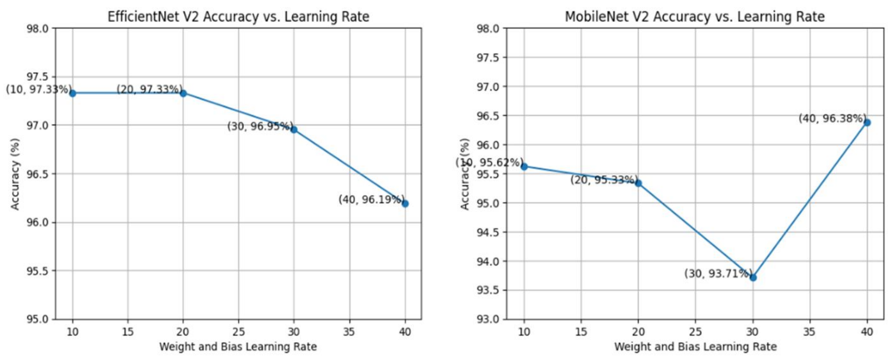
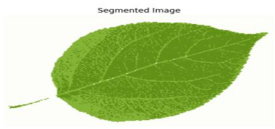
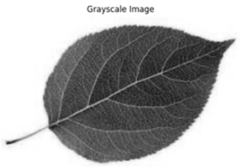
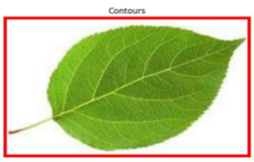
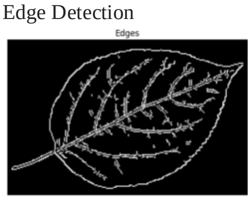
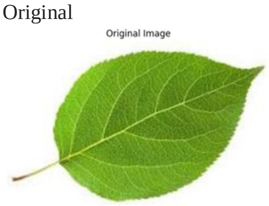


Fig.8: Variation of Weight and bias learning rate

VII. Results for Different Classes using Different Feature Extraction Techniques

TABLE VII: FEATURE EXTRACTION AND CLASSIFICATION ACCURACY OF SOYBEAN LEAF DISEASE

Class	**Methods for Feature Extraction/Learning**	**Features & Attribute Values**	**Model for Classification**	**Accuracy (EfficientNetV2)**	**Accuracy (MobileNetV2)**
Segmented 	K-means clustering[10]	1.Mean R: 100, Mean G: 120, Mean B: 130; Std R: 20, Std G: 22, Std B: 18	EfficientNetV2 (minibatch size 2), MobileNetV2 (minibatch size 12)	99.24%	96.51%
Grayscale 	Gray Level Co-occurrence Matrix (GLCM) [11]	Contrast: 0.8, Dissimilarity: 0.2, Homogeneity: 0.9, Energy: 0.1, Correlation: 0.9	EfficientNetV2 (minibatch size 2), MobileNetV2 (minibatch size 12)	90.5%	86.5%
Contour 	Contour Detection (cv2.findContours) [12]	Area: 1500, Perimeter: 200	EfficientNetV2 (minibatch size 2), MobileNetV2 (minibatch size 12)	92.5%	88.5%
Edge Detection 	Canny Edge Detection (cv2.Canny) [13]	Edge features (Canny edge detection output)	EfficientNetV2 (minibatch size 2), MobileNetV2 (minibatch size 12)	88.5%	84.5%
Background Removed 	Color Thresholding and Masking [14]	Mean R: 110, Mean G: 130, Mean B: 120; Std R: 25, Std G: 20, Std B: 22	EfficientNetV2 (minibatch size 2), MobileNetV2 (minibatch size 12)	94.5%	90.5%
Original 	Combined (Color, Texture, Shape) Features[15]	Combined features: Mean R: 105, Mean G: 125, Mean B: 135; Contrast: 0.75, Dissimilarity: 0.25, Area: 1550, Perimeter: 210 |	EfficientNetV2 (minibatch size 2), MobileNetV2 (minibatch size 12)	96.5%	92.5%

VIII. Conclusion

The photos from the NCSRP (North Central Soybean Research Programme) dataset were used to classify soybean crop diseases using EficientNetV2 and MobileNetV2, two pre-trained deep learning architectures .Using 13,262 photos, the classification accuracy for EficientNetV2 was 97.49%, while for MobileNetV2, it was

97.29%.A range of minibatch sizes, weight and bias learning rates, and picture counts have all been used to assess the models' performance. The quantity of photos had a big impact on how well the models worked. A maximum of 373 photos gives the highest accuracy. While there was no discernible relationship between the accuracy of classification and the fine-tuning of the minibatch size in MobileNetV2, there was a drop in accuracy in EficientNetV2 with increasing minibatch size. In a similar vein, accuracy in MobileNetV2 was dramatically boosted after learning rate of 30 by fine-tuning weight and bias learning rate. As the weight and bias learning rates were raised in EficientNetV2, the accuracy decreased. Compared to the deep EficientNetV2, MobileNetV2 offers a decent accuracy with less execution time in terms of computational burden More research reveals that efficientNetV2 performs better than MobileNetV2.The maximum accuracy achieved for segmented features, at 99.24 and 96.51, respectively. While colour features are more discriminative than grayscale features, both MobileNetV2 and EfficientNetV2 obtained 90.5% and 86.5% accuracy, respectively. The contour-based features perform well; EfficientNetV2 and MobileNetV2 both achieve 92.5% and 88.5% accuracy, respectively. For EfficentNetV2 and MobileNetV2, the accuracy of canny edge detection was lower than the other two, at 88.5% and 84.5%, respectively. By using thresholding to eliminate background noise, the model performs better by 94.5% and 90.5%, respectively. Excellent accuracy of 96.5 and 92.5 percent for EfficientNetV2 and MobileNetV, respectively, is also achieved by integrating the shape and texture features.

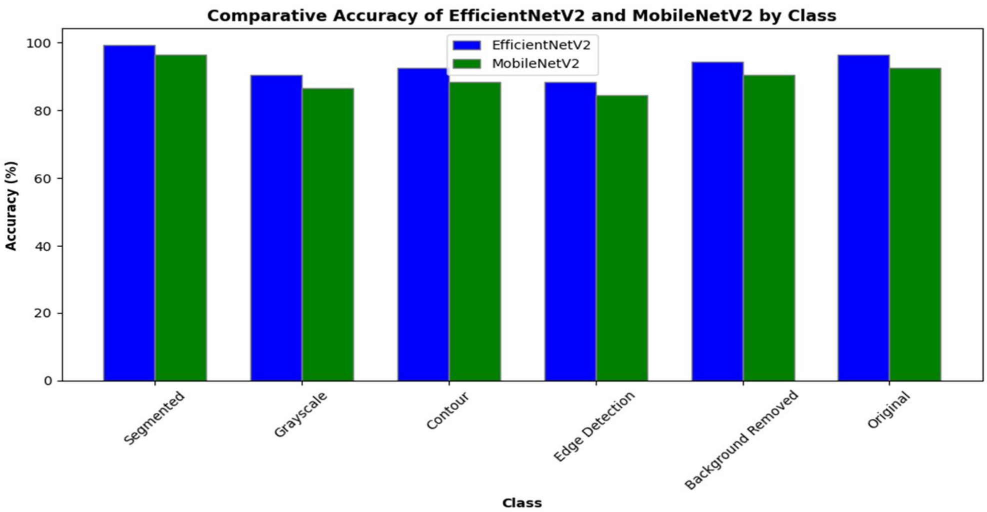


Figure 9: Model performance comparision per class

REFERENCES

[1] Ishana Attri, Lalit Kumar Awasthi, Teek Parval Sharma, Priyanka Rathee,A review of deep learning techniques used in agriculture,Ecological Informatics,Volume 77,2023,102217,ISSN 1574-541,https://doi.org/10.1016/j.ecoinf.2023.102217.

[2] Kumar, H., Velu, S., Lokesh, A., Suman, K., & Chebrolu, S. (2023). "Cassava Leaf Disease Detection Using Ensembling of EfficientNet, SEResNeXt, ViT, DeIT and MobileNetV3 Models." In: Yadav, R.P., Nanda, S.J., Rana, P.S., & Lim, M.H. (Eds.), Proceedings of the International Conference on Paradigms of Computing, Communication and Data Sciences. Algorithms for Intelligent Systems. Springer, Singapore. DOI: 10.1007/978-981-19-8742-7_15.

[3] https://soybeanresearchinfo.com/soybean-diseases/

[4] T. R., M. ., Kumar V., V. ., Sivakami, R. ., Manimozhi, I., Krishnamoorthy, N., & Swapna, B. (2023). Early Predictive Model for Detection of Plant Leaf Diseases Using MobileNetV2 Architecture. International Journal of Intelligent Systems and Applications in Engineering, 11(2), 46–54.

[5] Zhou, F.; Hu, S.; Wan, X.; Lu, Z.; Wu, J. Diplin: A Disease Risk Prediction Model Based on EfficientNetV2 and Transfer Learning Applied to Nursing Homes. Electronics 2023, 12, 2581. https://doi.org/10.3390/electronics12122581.

[6] Mohaimenul Azam Khan Raiaan, Sadman Sakib, Nur Mohammad Fahad, Abdullah Al Mamun, Md. Anisur Rahman, Swakkhar Shatabda, Md. Saddam Hossain Mukta,A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks,Decision Analytics Journal,Volume 11,2024,100470,ISSN 2772-6622,https://doi.org/10.1016/j.dajour.2024.100470.

[7] Aggarwal, M.; Khullar, V.; Goyal, N.; Alammari, A.; Albahar, M.A.; Singh, A. Lightweight Federated Learning for Rice Leaf Disease Classification Using Non Independent and Identically Distributed Images. Sustainability 2023, 15, 12149. https://doi.org/10.3390/su151612149.

[8] wmick Guha Paul, Al Amin Biswas, Arpa Saha, Md. Sabab Zulfiker, Nadia Afrin Ritu, Ifrat Zahan, Mushfiqur Rahman, Mohammad Ashraful Islam,A real-time application-based convolutional neural network approach for tomato leaf disease classification,Array,Volume 19,2023,100313,ISSN 2590-0056,https://doi.org/10.1016/j.array.2023.100313.
[9] Vimal Singh, Anuradha Chug, Amit Prakash Singh,Classification of Beans Leaf Diseases using Fine Tuned CNN Model,Procedia Computer Science,Volume 218,2023,Pages 348-356,ISSN 1877-0509,https://doi.org/10.1016/j.procs.2023.01.017.
[10] Seyed Mohamad Javidan, Ahmad Banakar, Keyvan Asefpour Vakilian, Yiannis Ampatzidis,Diagnosis of grape leaf diseases using automatic K-means clustering and machine learning, Smart Agricultural Technology,Volume 3,2023,100081,ISSN 2772-3755,https://doi.org/10.1016/j.atech.2022.100081.
[11] M. Yogeshwari, G. Thailambal,Automatic feature extraction and detection of plant leaf disease using GLCM features and convolutional neural networks,Materials Today: Proceedings,Volume 81, Part 2,2023,Pages 530-536,ISSN 2214-7853,https://doi.org/10.1016/j.matpr.2021.03.700.
[12] Rodriguez-Gallo, Y.; Escobar-Benitez, B.; Rodriguez-Lainez, J. Robust Coffee Rust Detection Using UAV-Based Aerial RGB Imagery. AgriEngineering 2023, 5, 1415-1431. https://doi.org/10.3390/agriengineering 5030088.
[13] Jha, Pradeep, Deepak Dembla, and Widhi Dubey. "Implementation of Machine Learning Classification Algorithm Based on Ensemble Learning for Detection of Vegetable Crops Disease." International Journal of Advanced Computer Science & Applications 15.1 (2024).
[14] Bondre, Shweta, and Dipti Patil. "Crop disease identification segmentation algorithm based on Mask-RCNN." Agronomy Journal 116.3 (2024): 1088-1098.
[15] K. M. Hosny, W. M. El-Hady, F. M. Samy, E. Vrochidou and G. A. Papakostas, "Multi-Class Classification of Plant Leaf Diseases Using Feature Fusion of Deep Convolutional Neural Network and Local Binary Pattern," in IEEE Access, vol. 11, pp. 62307-62317, 2023, doi: 10.1109/ACCESS.2023.3286730