Classification of Bean Leaf Lesions Using Modified EfficientNetV2 for Implementation in TensorFlow Lite

1st Muhammad Abiya Makruf
School of Computing
Telkom University
Bandung, Indonesia
abiyamf@student.telkomuniversity.ac.id

2nd Falah Asyraf Darmawan Putra
School of Computing
Telkom University
Bandung, Indonesia
falaaah@student.telkomuniversity.ac.id

3rd Akif Rachmat Hidayah
School of Computing
Telkom University
Bandung, Indonesia
akifrh@student.telkomuniversity.ac.id

4th Mahmud Dwi Sulistiyo
School of Computing
Telkom University
Bandung, Indonesia
mahmuddwis@telkomuniversity.ac.id

Abstract*—Bean leaf diseases such as Angular Leaf Spot and Bean Rust significantly threaten the productivity and quality of common beans (Phaseolus vulgaris L.), a critical crop for global food security. While existing approaches like ensemble methods demonstrate high classification accuracy, their reliance on multiple architectures increases computational complexity, making them unsuitable for resource-constrained devices. This research develops a lightweight yet high-performing model for classifying bean leaf diseases using the EfficientNetV2 architecture and its modified variant. Key metrics evaluated include classification accuracy, inference time, and TensorFlow Lite (TFLite) model size, emphasizing suitability for real-time deployment. EfficientNetV2 was fine-tuned on a dataset containing healthy leaves, Angular Leaf Spot, and Bean Rust. To reduce complexity, the architecture was modified by selectively removing specific blocks. The model was optimized for deployment through TFLite conversion and post-training quantization, significantly reducing size and inference time. Grad-CAM visualizations enhanced interpretability by highlighting regions influencing model predictions. Experiments evaluated trade-offs between accuracy and efficiency. The modified EfficientNetV2B0 achieved 97.76% test accuracy, a model size of 6.18 MB, and an inference time of 0.0594 seconds. These results demonstrate improved efficiency compared to larger models and ensemble approaches, while maintaining competitive accuracy. This research highlights the feasibility of deploying lightweight, accurate models for real-time agricultural applications. Future work could explore integrating ensemble techniques or federated learning to further enhance robustness and scalability in agricultural disease detection.

Keywords—bean leaf diseases, EfficientNetV2, modified EfficientNetV2B0, tensorflow lite, model optimization, grad-cam

I. INTRODUCTION

Common bean (Phaseolus vulgaris L.) is a globally significant crop, encompassing both dry beans and green snap beans. It is widely cultivated in regions with temperate to subtropical climates, such as Africa, as well as in various other continents [1]. Since 1998, Africa's agricultural sector has experienced increased demand for beans, leading to intensified production and improved crop management to reduce pest and disease losses. While some farmers benefit from bean export opportunities due to surpluses, disparities persist in production regions, highlighting varied challenges and opportunities. [2].

The common bean (Phaseolus vulgaris L.) is a critical legume for human consumption, with approximately 30% of its global production attributed to small-scale farmers in Latin America and Africa. It represents the second most significant source of dietary fiber for human nutrition and ranks as the third largest provider of calories among all agricultural products in Eastern and Southern Africa [1]. Despite this, Angular leaf spot has been reported in at least 78 countries, primarily in tropical and temperate regions of the world, including Africa, Americas, Europe, Asia, Indonesia, Australia, and Oceania [3].

ALS has been reported in 18 eastern and southern African countries and is regarded as the most significant dry bean disease in Malawi, as well as in central, eastern, and humid southern African regions [3]. Additionally, bean rust, another significant disease affecting common beans, has been widely reported across bean-growing regions. This fungal disease poses a substantial threat to yield and quality, necessitating efficient methods for its identification and management. Accurate classification of bean rust alongside other diseases is critical for developing targeted interventions.

In light of the challenges posed by diseases like Angular Leaf Spot and Bean Rust, advancements in technology have enabled more precise and efficient methods for crop classification and disease detection. Growing beans is important since they are a staple meal for so many people throughout the world. Bean rust and angular leaf spot are just two of the many diseases that threaten the well-being of bean crops and, in turn, cause considerable output losses. In recent research, an ensemble deep learning strategy named EnDeel was proposed to address the reliable identification of bean leaf lesions as healthy, angular leaf spots, or bean rust. While the findings demonstrated that EnDeeL outperformed individual classifiers, achieving notable accuracy, the reliance on multiple architectures and majority voting introduces complexity and computational overhead [4]. This highlights a gap in the need for simpler yet effective models that can achieve similar or improved accuracy without significant resource requirements.

However, previous studies utilizing MobileNet architectures for bean leaf disease classification have highlighted the need for further exploration to enhance classification performance [5]. While recent research introduced ensemble methods like EnDeeL, which demonstrated improved accuracy, the reliance on multiple architectures introduces complexity and computational overhead. This underscores the necessity of developing simpler and more efficient approaches to address the classification of bean leaf diseases effectively.

To address this gap, this study proposes the use of EfficientNetV2 [6] for bean leaf disease classification. EfficientNetV2 is selected due to its ability to balance computational efficiency and accuracy by scaling the network width, depth, and resolution effectively. This method involves training the EfficientNetV2 architecture on a labeled dataset of bean leaf images containing healthy leaves, Angular Leaf Spot, and Bean Rust. By leveraging transfer learning, pre-trained weights will be fine-tuned to optimize classification performance on the specific dataset.

In addition to classification accuracy, this study evaluates key practical aspects, including inference time, the size of the TensorFlow Lite (TFLite) model, and the interpretability of the model through Grad-CAM visualization to highlight the areas of the leaf influencing the model's decisions. Moreover, a modified version of EfficientNetV2 is proposed to further enhance performance by modifying the architecture to the specific requirements of bean leaf classification. This modification aims to improve model efficiency and maintain high accuracy while ensuring the feasibility of deployment on resource-constrained devices.

II. RELATED WORK

Singh et al. [7] utilized MobileNetV2, EfficientNetB6, and NasNet for bean leaf disease classification, achieving the highest validation accuracy of 91.74% with EfficientNetB6. Their study highlights the effectiveness of pre-trained CNN models and optimization techniques in improving classification accuracy. Tiwari and Kumar [4] proposed an ensemble approach (EnDeeL) combining MobileNetV2, ResNet50, EfficientNetB2, DenseNet121, and VGG16 for robust classification of bean leaf lesions, including healthy leaves, Angular Leaf Spot, and Bean Rust. By using majority voting from the top three architectures, EnDeeL achieved a test accuracy of 92.12%, outperforming individual classifiers.

Elfatimi et al. [5] leveraged MobileNet for the classification of bean leaf diseases using a public dataset that includes images of healthy leaves, Angular Leaf Spot, and Bean Rust. Their MobileNet-based model achieved a test accuracy of 92.97%, showcasing its effectiveness in disease classification while maintaining computational efficiency suitable for deployment. Serttaş and Deniz [8] implemented a transfer learning approach using pre-trained CNN models, including VGG16, ResNet50, and MobileNetV2, for bean leaf disease detection. Their study utilized a dataset of 1,295 images divided into three classes: healthy leaves, Angular Leaf Spot, and Bean Rust. The ResNet50 model achieved the highest accuracy of 98.33%, demonstrating the effectiveness of transfer learning in agricultural applications.

A robust multiclass detection approach combining GoogleNet and CNN was proposed for bean leaf disease classification [9]. Leveraging CNN's hierarchical feature learning and GoogleNet's inception modules, the ensemble achieved a classification accuracy of 98.93% and demonstrated superior performance compared to existing models. This study highlights the potential of ensemble techniques in precision agriculture.

This research builds on prior works by specifically focusing on the classification of bean leaf lesions using EfficientNetV2 and its modified variant. Unlike previous studies, this research investigates additional metrics, including inference time and TensorFlow Lite (TFLite) model size, to assess its suitability for deployment on edge devices. Furthermore, it uniquely incorporates Grad-CAM visualizations to improve the interpretability of model predictions, emphasizing the areas of the leaf that influenced the model's decision. This comprehensive evaluation aims to optimize the trade-off between computational efficiency and classification accuracy, laying the groundwork for future development of lightweight, high-performance models for real-time agricultural applications.

III. PROPOSED METHOD

Proposed method in this paper for classifying bean leaf diseases using a modified version of the EfficientNetV2 architecture. The objective is to develop a model that balances high accuracy with reduced model size and faster inference times. The model is trained using this architecture to ensure efficient performance. Once the model is trained, it is converted to TensorFlow Lite (TFLite) format, making it ready for deployment on mobile devices. This conversion enables real-time leaf disease classification while maintaining both efficiency and accuracy. The method includes dataset preparation, training with the modified EfficientNetV2, and conversion to TFLite, resulting in a model that is ready for deployment in mobile disease detection systems.

A. Dataset

The dataset used in this paper, the Bean Leaf Lesions Classification dataset from kaggle consists of 1,167 images of bean leaves categorized into several classes based on the type of lesions they exhibit. The classes in this dataset include Angular Leaf Spot, Bean Rust, and Healthy. Each class represents a distinct condition of the bean leaves, ranging from healthy leaves to those affected by various diseases. These categories make the dataset ideal for training machine learning models to classify and diagnose different leaf diseases accurately. Sample images of each class in the dataset are shown in Fig 1. The images in the dataset are 500x500 pixels in size and are in JPEG format.

(a) Healthy (b) Angular Leaf Spot (c) Bean Rust

Fig 1. Sample images of bean leaf conditions from the Bean Leaf Lesions Classification dataset. (a) Healthy, (b) Angular Leaf Spot, (c) Bean Rust.

The dataset is split into three main subsets: training, validation, and testing. The training set consists of 1,034 images, which are used to train the model. To assess the model's performance during training, 10% of the training images, amounting to 102 images, are set aside for validation. The testing set, consisting of 133 images, is used to evaluate the final model's performance after training and validation. These 133 images are sourced directly from the dataset “Val” directory and provide an unbiased evaluation of how well the model performs on unseen data. TABLE I shows the detailed distribution of the dataset.

TABLE I. DISTRIBUTION OF IMAGE ACROSS TRAINING, VALIDATION, AND TESTING SETS

Class	Training	Validation	Testing
Healthy	307	34	44
Angular Leaf Spot	311	34	44
Bean Rust	314	34	45
Total Images	**932**	**102**	**133**

B. *Experimental Schemes*

Several experimental schemes will be conducted to compares the performance of the original EfficientNetV2 and a modified EfficienNetV2. Each scheme tests different configurations of model architecture.

1) Original Architecture:

a) Freeze All Layers with Pre-Trained Weights: The original architecture with no trainable layers, relying solely on pre-trained weights from ImageNet.

b) All Layesrs Trainable with No Pre-Trained Weights: The original model with all layers trainable, using no pre-trained weights. Training from scratch evaluates how well the model could learn directly from the dataset without any prior knowledge.

2) Modified EfficientNetV2 Architecture: Finally, this scheme uses a modified EfficientNetV2 architecture where certain blocks are removed. This configuration tests if a lighter model could still maintain high classification accuracy compares to the original architecture model.

C. *Modified EfficientNetV2*

The model used in this paper is based on the EfficientNetV2 architecture, which was modified by removing blocks 4 and 6, as shown in Fig 2, to reduce the number of parameters and cut down computation time. Through experimentation, it was observed that eliminating these blocks resulted in a significant reduction in model size and inference time, while maintaining comparable classification accuracy. The removal of these blocks helped to reduce the overall computational burden, making the model more efficient and suitable for deployment on mobile devices.

After these modifications, the model's size decreased by approximately 67%, and inference time was reduced by approximately 97%. Despite these reductions, the accuracy of the model in classifying bean leaf diseases remained high, ensuring that the model could still effectively detect conditions such as Angular Leaf Spot, Bean Rust, and Healthy leaves. These results demonstrate that the modifications successfully optimized the EfficientNetV2 architecture for ready for deployment, balancing both efficiency and performance.

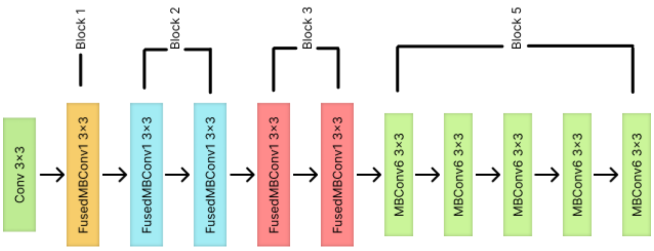

Fig 2. Modified EfficientNetV2 Architecture with the removal of Block 4 and 6 to reduce model size and inference time.

D. *Model Optimization for Deployment*

To make the model suitable for deployment on mobile devices, several optimization techniques were applied to ensure both efficiency and performance. The primary goal was to reduce the model size and inference time without sacrificing accuracy, as mobile devices typically have limited processing power and memory.

First, after training the model using the modified EfficientNetV2 architecture, the model was converted into TensorFlow Lite (TFLite) format. This conversion significantly reduced the model's size, making it more efficient for mobile devices. TFLite is specifically designed for running machine learning models on mobile and embedded devices, ensuring that the model can be executed with minimal latency and resource usage.

E. *Evaluation Metrics*

To evaluate the performance of the model, several metrics were used, including the standard accuracy, precision, recall, and F1-score. These metrics provided insight into the model's ability to classify bean leaf diseases accurately. In addition to these, two critical metrics for mobile deployment were also considered: inference time and model size. Inference time measures how quickly the model can make predictions, which is essential for real-time applications on mobile devices. Model size, on the other hand, is crucial for ensuring the model can be efficiently stored and executed on devices with limited resources. These metrics, alongside the traditional performance measures, give a comprehensive understanding of the model's effectiveness and suitability for mobile deployment.

IV. RESULT AND ANALYSIS

In this section, the results of the experiments are presented and analyzed to assess the performance and effectiveness of the proposed method. The analysis includes a detailed comparison of the experimental setups, impact of the modified EfficientNetV2 architecture, and Grad-CAM analysis is also utilized to visualize the regions of the input images that the model focuses on when making predictions. The section concludes with a discussion of the optimization results for deployment, analyzing how the proposed modification can make a more suitable model for real-world applications.

A. *Original Architecture Performance*

The result for the first experimental scheme, as shown in TABLE II, highlight the performance of the original EfficientNetV2 architecture with all layers frozen and pre-trained weights. The models result in high training, validation, and testing accuracy across all configurations. EfficientNetV2B0, with the smallest number of parameters (6,280,531), achieves the lowest testing accuracy (0.9549), while EfficientNetV2B3, with the largest number of parameters (13,357,377), achieves the highest testing accuracy (0.9849). These results demonstrate a clear relationship between model complexity and performance, where more complex models with larger parameter sizes tend to generalize better to unseen data. However, this comes with the trade-off of higher computational requirements, as seen in the increase in total parameters from EfficientNetV2B0 to EfficientNetV2B3.

TABLE II. PERFORMANCE METRICS FOR ORIGINAL EFFICIENTNETV2 WITH ALL LAYERS FROZEN AND PRE-TRAINED WEIGHTS

Model	Train Acc	Val Acc	Test Acc	Total Parameter
EfficientNetV2B0	0.9861	**0.9949**	0.9549	**6,280,531**
EfficientNetV2B1	**0.9903**	0.9925	0.9625	7,292,343
EfficientNetV2B2	0.9764	0.9902	**0.9849**	9,163,361
EfficientNetV2B3	0.9775	**0.9949**	**0.9849**	13,357,377

TABLE III presents the performance metrics of the second experimental scheme. Original EfficientNetV2 models with all layers trainable and no pre-trained weights, evaluating their ability to learn directly from the dataset. This scheme is particularly relevant because the modified EfficientNetV2 architecture, cannot use pre-trained ImageNet weights due to its structural changes, making it crucial to understand how models perform when trained from scratch.

Among the models, EfficientNetV2B2 achieves the highest training accuracy (0.8219), validation accuracy (0.8824), and testing accuracy (0.8872). EfficientNetV2B1 also performs well, with slightly lower validation (0.8725) and testing (0.8571) accuracies. In contrast, EfficientNetV2B3, despite having the largest parameter count (13,357,377), struggles to generalize, showing the lowest validation accuracy (0.7549) and testing accuracy (0.8271). This proves that larger models may not always produce better result, especially without pre-trained weights. EfficientNetV2B0, the smallest model, achieves a relatively decent performance given its size, though its testing accuracy (0.8120) is the lowest overall. These results highlight the importance of striking a balance between model complexity and performance, aligning with the objective of the modified architecture.

TABLE III. PERFORMANCE METRICS FOR ORIGINAL EFFICIENTNETV2 WITH ALL LAYERS TRAINABLE AND NO PRE-TRAINED WEIGHTS

Model	Train Acc	Val Acc	Test Acc	Total Parameter
EfficientNetV2B0	0.7682	0.8039	0.8120	**6,280,531**
EfficientNetV2B1	0.8101	0.8725	0.8571	7,292,343
EfficientNetV2B2	**0.8219**	**0.8824**	**0.8872**	9,163,361
EfficientNetV2B3	0.7393	0.7549	0.8271	13,357,377

B. Impact of Modified EfficientNet Architecture

The results of the modified EfficientNetV2 architecture with various blocks removed, as detailed in TABLE IV, show the impact of these modifications on model performance and complexity. Among the configurations, removing Block 4 & 6 provides the best balance between accuracy and model size. This setup achieves the highest testing accuracy (0.9776) and validation accuracy (0.9412) while reducing the total parameters to 1,639,019. Its decent training accuracy (0.8777) further demonstrates effective learning without overfitting. These results prove the effectiveness of selectively removing blocks to optimize the architecture for deployment in resource-constrained environments.

TABLE IV. PERFORMANCE METRICS OF THE MODIFIED EFFICIENTNETV2B0 ARCHITECTURE WITH DIFFERENT BLOCKS REMOVED

Block Removed	Train Acc	Val Acc	Test Acc	Total Parameter
Block 1	0.8294	0.8824	0.8881	6,275,363
Block 2	**0.8777**	0.9314	0.9627	6,208,851
Block 3	0.8648	0.9216	0.9257	6,140,627
Block 4	0.8380	0.9314	0.8582	6,017,815
Block 5	0.8519	0.8431	0.8507	5,308,555
Block 6	0.8261	0.9216	0.8651	1,901,735
Block 4&5	0.8326	0.8922	0.9328	5,041,231
Block 4&6	**0.8777**	**0.9412**	**0.9776**	1,639,019
Block 5&6	0.8487	0.9118	0.9179	**920,031**

As shown in TABLE V, among the EfficientNetV2 models, the modified EfficientNetV2B0 achieved the highest testing accuracy (0.9776) while maintaining a competitive training accuracy (0.8777) and a total parameter count of 1,639,019. This balance indicates that the model has learned from the training data without significant overfitting. EfficientNetV2B1 and EfficientNetV2B2 achieved slightly lower test accuracies (0.8797 and 0.8947, respectively) while showing comparable training accuracy, demonstrating their generalizability with moderate parameter counts.

MobileNetV3Small achieved high training accuracy (0.9839) but performed poorly on test accuracy (0.7594), indicating overfitting due to its limited parameter count of 1,120,115. On the other hand, MobileNetV3Large achieved the best testing accuracy (0.9925) and training accuracy (0.9979) but this came at the cost of the largest parameter size (3,275,651), making it less suitable for deployment on resource-constrained devices.

These results indicates that the modified EfficientNetV2B0 model provides a good balance between training accuracy, testing accuracy, and model efficiency. By selectively removing blocks, it achieves high classification performance while remaining lightweight and suitable for mobile deployment.

TABLE V. PERFORMANCE METRICS COMPARISON OF EFFICIENTNETV2 VARIANTS WITH REMOVED BLOCK 4 & 6 AND MOBILENETV3 MODELS

Model	Train Acc	Val Acc	Test Acc	Total Parameter
EfficientNetV2B0	0.8777	0.9412	0.9776	1,639,019
EfficientNetV2B1	0.8691	0.9118	0.8797	1,976,839
EfficientNetV2B2	0.8691	0.9216	0.8947	2,270,707
EfficientNetV2B3	0.8670	0.8824	0.9023	3,049,123
MobileNetV3Small	0.9839	0.8725	0.7594	**1,120,115**
MobileNetV3Large	**0.9979**	**0.9902**	**0.9925**	3,275,651

C. Grad-CAM Analysis

Grad-CAM visualization for the modified EfficientNetV2B0 model as shown in Fig 3, showcasing its ability to identify disease-specific patterns in bean leaf images. The top row shows the original images for three classes: healthy leaves, leaves affected by angular leaf spot, and leaves with bean rust. While the bottom row shows heatmap overlays that highlight the regions of the leaf that contributed most significantly to the model's predictions.

For the healthy leaf image, the heatmap indicates uniform attention across the leaf surface, suggesting that the model recognizes the absence of disease-specific features. In contrast, for the angular leaf spot and bean rust classes, the heatmaps display concentrated attention on the diseased areas, particularly around lesions and discolorations, which are key indicators of the respective diseases.

These results demonstrate the interpretability of the modified EfficientNetV2B0 model, providing insights into how it arrives at its classification decisions. The Grad-CAM analysis proves that the model correctly focuses on biologically relevant regions, validating its suitability for practical applications in agricultural disease detection.

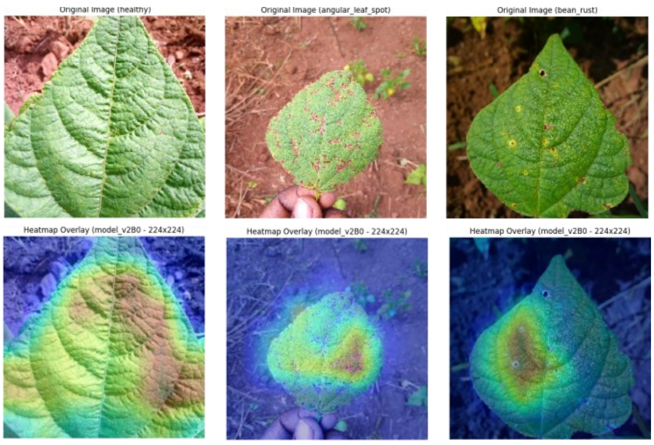

Fig 3. Grad-CAM Visualizations of Modified EfficientNetV2B0 Model Predictions

D. TensorFlow Lite Optimization for Deployment

TABLE VI show the impact of TFLite optimization on the model size and inference time for the modified EfficientNetV2 variants (B0, B1, B2, and B3). The results demonstrate a significant reduction in model size across all variants, ranging from approximately 67% (EfficientNetV2B0) to 71% (EfficientNetV2B3). For instance, the original EfficientNetV2B0 model size of 19.08 MB is reduced to 6.18 MB, highlighting the efficiency of TFLite in making these models deployment-ready for resource-constrained environments.

In terms of inference time, TFLite models result approximately 97% reductions compared to their original versions. The EfficientNetV2B0 model, for example, sees its inference time drop from 2.125 seconds to just 0.0594 seconds, while B1 achieves a similar improvement from 2.557 seconds to 0.0517 seconds. While B1 is slightly faster than B0, the difference is negligible, and both models maintain exceptional performance suitable for real-time applications. This demonstrates that TFLite optimization not only reduces model size but also significantly enhances inference speed. Such improvements not only make the models deployment-ready but also extend their usability across a wider range of devices, including older hardware and devices with constrained memory or processing power. This optimization process enables real-time applications, such as image recognition and video analysis, to operate seamlessly even on low-powered devices, broadening the potential applications for these EfficientNetV2 variants.

TABLE VI. COMPARISON OF MODEL SIZE AND INFERENCE TIME FOR ORIGINAL AND TENSORFLOW LITE VERSIONS OF MODIFIED EFFICIENTNETV2 VARIANTS

Model Name		**B0**	**B1**	**B2**	**B3**
Model Size (MB)	Original	19,08	23,05	26,4	35.32
	TFLite	6,18	7,46	8,57	11.51
Inference Time (s)	Original	2.125	2.557	2.9793	3.0687
	TFLite	0.0594	0.0517	0.0706	0.1218
Test Accuracy	Original	0.9776	0.8797	0.8947	0.9023
	TFLite	0.9776	0.8797	0.8947	0.9023

Combined with its strong classification performance as shown (0.9776 test accuracy), EfficientNetV2B0 emerges as the most suitable choice for deployment. Its compact size (6.18 MB), fast inference time (0.0594 seconds), and high accuracy showcase its ability to balance model complexity and performance effectively. This outcome directly aligns with the objective of developing a model that is lightweight, efficient, and ready for deployment on mobile or embedded systems. By leveraging architecture modifications and TFLite optimization, the modified EfficientNetV2B0 achieves a good balance between computational efficiency and classification accuracy, making it ideal for real-world applications.

V. CONCLUSION

This study evaluated the performance of the EfficientNetV2 architecture, including a custom modification of its layers designed specifically for the classification of bean leaf diseases. The analysis focused on key metrics such as accuracy, inference time, and model size to assess suitability for deployment in resource-constrained environments. Initially, the original EfficientNetV2 architecture was evaluated in two configurations: with all layers frozen and pre-trained weights, and with all layers trainable and no pre-trained weights. The results demonstrated that while larger models like EfficientNetV2B3 achieved higher accuracy (0.9849), they also required significantly more computational resources. Smaller models, such as EfficientNetV2B0, provided reasonable performance (0.9549) with reduced complexity, making them better suited for deployment needs.

The custom modifications introduced in this study further enhanced the model's performance and efficiency. By restructuring the layers within EfficientNetV2, the modified EfficientNetV2B0 model achieved the best balance between accuracy and efficiency. It reached a high test accuracy of 0.9776% with a significantly reduced parameter count (1,639,019), demonstrating the effectiveness of the custom modifications. Grad-CAM analysis provided interpretability by highlighting biologically relevant regions of bean leaf images, validating the model's suitability for real-world agricultural applications. TensorFlow Lite optimization further improved deployment readiness by reducing the model size to 6.18 MB and inference time to 0.0594 seconds, making the modified EfficientNetV2B0 model an ideal candidate for real-time, mobile, and embedded applications. These results demonstrate that custom modifications and optimization strategies can create lightweight yet powerful models for practical use.

Future research could extend these findings by exploring additional custom modifications to further optimize the architecture for specific agricultural tasks and expanding the dataset to include more diverse environmental conditions and disease types. Real-time deployment on edge devices integrated with field sensors could provide valuable insights into practical challenges and opportunities. Additionally, applying ensemble methods or federated learning approaches could further enhance model robustness and scalability for large-scale agricultural applications.

References

[1] P. Pamela, D. Mawejje, and M. Ugen, "Severity of angular leaf spot and rust diseases on common beans in Central Uganda".

[2] A. Farrow and R. Muthoni-Andriatsitohaina, Atlas of Common Bean Production in Africa: Second Edition. Nairobi, Kenya: Pan-Africa Bean Research Alliance (PABRA); International Center for Tropical Agriculture (CIAT), 2020, p. 242.

[3] "A review of angular leaf spot of common bean (Phaseolus vulgaris L)," vol. 3, no. 2, 1997.

[4] R. G. Tiwari and A. Kumar, "Bean Leaf Lesions Image Classification: A Robust Ensemble Deep Learning Approach," 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS), Manama, Bahrain, 2024, pp. 986-993, doi: 10.1109/ICETSIS61505.2024.10459697.

[5] E. Elfatimi, R. Eryigit, and L. Elfatimi, "Beans Leaf Diseases Classification Using MobileNet Models," IEEE Access, vol. 10, pp. 9471–9482, 2022, doi: 10.1109/ACCESS.2022.3142817.

[6] M. Tan and Q. V. Le, "EfficientNetV2: Smaller Models and Faster Training," Apr. 2021.

[7] V. Singh, A. Chug, and A. P. Singh, "Classification of Beans Leaf Diseases using Fine Tuned CNN Model," *Procedia Computer Science*, vol. 218, pp. 348–356, 2023, doi: 10.1016/j.procs.2023.01.017.

[8] S. Serttaş and E. DeniZ, "Disease detection in bean leaves using deep learning," Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering, vol. 65, no. 2, pp. 115–129, Dec. 2023, doi: 10.33769/aupse.1247233

[9] A. Kaur, V. Kukreja, D. Upadhyay, M. Aeri, and R. Sharma, "An Integrated GoogleNet with Convolutional Neural Networks Model for Multiclass Bean Leaf Lesion Detection," 2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), Gwalior, India, 2024, pp. 1-6, doi: 10.1109/IATMSI60426.2024.10502575.