Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6, 793-813
2024
Publisher: Learning Gate
DOI: 10.55214/25768484.v8i6.2167

"A novel hybrid deep learning framework for enhanced segmentation of rice leaf diseases using attention-driven efficientnet models"

Pooja Sharma[1], Ajay Khunteta[2*]
[1,2]Poornima University, Jaipur; er9.pooja@gmail.com (P.S.) ajay.khunteta@poornima.edu.in (A.K.).

Abstract: Rice diseases such as bacterial leaf blight, brown spot, and blast pose significant threats to global food security by reducing crop yields, making early and accurate detection crucial. This study aims to improve automated segmentation of rice leaf diseases using advanced deep learning techniques, specifically U-Net with MobileNetV2, DeepLabV3+ with EfficientNetB4, and U-Net with EfficientNetB7 integrated with attention gates. The models were evaluated on a dataset of diseased rice leaves for segmentation accuracy, computational efficiency, and robustness in real-world conditions. Results show that the U-Net with EfficientNetB7 and attention gates outperforms the other models, particularly for complex leaf images, achieving superior accuracy and generalization. This research provides a practical, real-time solution for early disease detection in rice, contributing to precision agriculture by helping reduce crop losses and optimize the use of agrochemicals, ultimately promoting sustainable crop management.

Keywords: *Attention mechanisms, Computational efficiency, Crop management, Deep learning, DeepLabV3+, Disease detection, Image segmentation, MobileNetV2, Precision agriculture, Rice leaf diseases, U-Net, Efficient Net.*

1. Background

Rice is a staple food for a significant portion of the global population, especially in Asia. Rice diseases, such as bacterial leaf blight, brown spot, and blast, pose severe threats to crop yields, impacting food security and farmers' livelihoods. Early and accurate detection of these diseases is crucial for effective management and prevention of crop loss. While numerous methods have been developed for rice leaf disease detection, challenges such as segmentation accuracy, handling complex backgrounds, and the need for large, labelled datasets persist.

Image segmentation is a vital technique in the field of computer vision and image processing, particularly for agricultural applications. It involves partitioning an image into different segments or regions, each representing a specific part of the image, such as the background, healthy plant tissue, or diseased regions. In the context of leaf disease detection, segmentation is the process of isolating and identifying the diseased areas on a leaf from the healthy areas and the background.

1.1. Problems Addressed

Accurate Disease Localization: Image segmentation allows for precise localization of diseased regions on leaves, distinguishing between healthy and infected tissues. This precision is crucial for early detection and accurate diagnosis of plant diseases.

Handling Complex Backgrounds: In field conditions, leaf images often have complex backgrounds, such as soil, other plants, and varying lighting conditions. Segmentation techniques can effectively separate the leaf from the background, ensuring that the analysis focuses solely on the leaf's surface.

Quantification of Disease Severity: By segmenting the diseased areas, it is possible to quantify the extent of the infection, providing valuable information on the severity of the disease. This data can help

* Correspondence: ajay.khunteta@poornima.edu.in

in determining the appropriate level of intervention needed.

Automated Monitoring and Early Detection: Automated segmentation allows for continuous monitoring of crops using imaging devices, such as drones or cameras installed in fields. Early detection of diseases through such automated systems can prevent the spread of the disease and reduce crop losses.

Improving Decision-Making: Segmentation helps in providing precise data that can be used to make informed decisions regarding the timing and type of interventions, such as pesticide application, to manage the disease effectively.

Introduction to Techniques: Several advanced techniques have been developed for leaf disease image segmentation, leveraging the power of machine learning, deep learning, and image processing algorithms. Traditional methods, such as thresholding and edge detection, have been supplemented by more sophisticated approaches like:

- K-Means Clustering and Superpixel Segmentation: These methods group pixels with similar features, allowing for better differentiation between diseased and healthy regions.
- Deep Learning Models: Convolutional neural networks (CNNs) and other deep learning architectures have revolutionized image segmentation. These models can learn complex patterns and features from large datasets, improving the accuracy of disease detection even in challenging conditions.
- Hybrid Techniques: Combining different segmentation methods, such as using both color and texture analysis, can enhance the robustness of the segmentation process, particularly in diverse field conditions.

2. Literature Survey on Rice Diseases Image Segmentation

The identification and segmentation of rice diseases using image processing and deep learning techniques have been a significant focus in agricultural research over the past decade. This survey summarizes key studies, focusing on methods used, findings, and their contributions to the field.

2.1. Key Findings from Major Papers

Automatic Rice Leaf Disease Segmentation: A study proposed an algorithm for automatic detection and segmentation of bacterial leaf blight and brown spot in rice leaves under varying environmental conditions. The method demonstrated robust performance in diverse conditions [1].

Image Processing Techniques Survey: A comprehensive survey reviewed various image segmentation and classification techniques for diagnosing rice plant diseases. It highlighted achievements and limitations in the field, suggesting future research directions [2].

Deep Learning for Rice Disease Detection: An approach combining deep learning with level set methods was proposed to improve the detection and classification of rice diseases. The method achieved high accuracy even with limited labeled data [3].

Bi-Level Thresholding for Disease Diagnosis: A portable tool using bi-level thresholding was developed to predict rice diseases. The method achieved better accuracy than traditional methods by effectively segmenting regions of interest [4].

Lesion Segmentation with Random Forest: This study presented a method combining superpixel segmentation and random forest classifiers to segment lesions from rice leaf blast images, achieving good segmentation performance [5].

Comparison of Segmentation Methods: Various image segmentation algorithms were compared to identify the most effective method for segmenting diseased rice leaves, emphasizing the need for a unique segmentation method [6].

Machine Vision for Disease Spots Extraction: A method based on machine vision was developed to segment and extract disease spots in rice leaves. The technique involved gray-level transformation followed by segmentation [7].

Detection Using SVM and Random Forest: An automated method for detecting rice diseases using

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 793-813, 2024
DOI: 10.55214/25768484.v8i6.2167

SVM and Random Forest classifiers was proposed. The method showed superior performance in identifying diseases compared to traditional approaches [8].

Copy Paste and Semantic Segmentation: A lightweight network using copy paste and semantic segmentation was introduced for accurate disease region segmentation and severity assessment in rice leaves [9].

Texture and Deep Features for Disease Detection: A deep learning model using SegNet for segmentation and deep recurrent neural networks (Deep RNN) for detection was proposed. The model achieved high accuracy and specificity [10].

Transfer Learning with CNN: A CNN-based model using transfer learning was developed for predicting rice leaf diseases, achieving 95.67% accuracy [11].

Otsu's Method for Segmentation: A method using Otsu's thresholding for segmentation combined with Local Binary Patterns (LBP) and Histogram of Oriented Gradients (HOG) achieved 94.6% accuracy for rice disease classification [12].

Fusion of FCM-KM and Faster R-CNN: A fusion method using FCM-KM clustering and Faster R-CNN for rapid detection of rice diseases demonstrated high accuracy and reduced detection time [13].

Deep CNN for Disease Identification: A CNN-based method achieved 95.48% accuracy in identifying 10 common rice diseases, significantly outperforming traditional machine learning models [14].

Attention Networks for Rice Disease Detection: Lightweight attention networks based on MobileNet-V2 achieved 99.67% accuracy in identifying rice plant diseases [15].

Backpropagation Neural Network: A method using backpropagation neural networks for detecting three common rice diseases achieved 100% accuracy [16].

Survey on Rice Disease Detection: A survey reviewed various image processing and machine learning techniques for detecting rice diseases, highlighting strengths and limitations [17].

Rice-Fusion Framework: A multimodality data fusion framework combining agro-meteorological data and image features achieved 95.31% accuracy for rice disease diagnosis [18].

2.2. Image Analysis for Disease Detection

An image processing method for classifying rice diseases based on leaf and lesion attributes was evaluated using a publicly available database [19].

Automated System for Disease Monitoring: An automated system using image processing for identifying four common rice diseases achieved high classification accuracy with k-NN and Minimum Distance Classifier (MDC) [(Joshi & Jadhav, 2016)] [20].

3. Research Problem & Objectives

Despite advances in image segmentation and deep learning, existing methods for rice leaf disease detection face several limitations:

- **Segmentation Accuracy:** Accurately segmenting diseased regions in rice leaves remains challenging due to the complex and varying backgrounds in field images, which often resemble the color and texture of the leaves.
- **Data Scarcity:** Deep learning models typically require large, annotated datasets for training. However, such datasets are often limited, particularly for rice diseases.
- **Model Generalization:** Models trained on controlled environments may not generalize well to real-world field conditions, where lighting, shadows, and occlusions vary significantly.
- **Computation Efficiency**: Real-time application of these models in the field requires computationally efficient algorithms that can process images quickly and accurately on resource-limited devices.

3.1. Objective

The primary objective of this research is to develop a robust, efficient, and accurate method for the

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 793-813, 2024
DOI: 10.55214/25768484.v8i6.2167

segmentation and classification of rice leaf diseases that can overcome the limitations of current approaches. This includes:

1. Enhancing Segmentation Accuracy: Developing advanced segmentation techniques that effectively differentiate between diseased and healthy regions in complex field images.
2. Data Augmentation and Synthetic Data Generation: Utilizing data augmentation techniques and synthetic data generation (e.g., copy-paste techniques) to expand the training dataset and improve model robustness.
3. Model Generalization: Implementing and testing models in diverse real-world conditions to ensure generalization across different environmental settings.
4. Computation Efficiency: Optimizing the model to run efficiently on portable devices for real-time disease detection in the field.

3.2. Expected Outcomes

- A novel segmentation algorithm that significantly improves the accuracy of disease region detection in rice leaf images.
- A deep learning model that generalizes well across different field conditions, ensuring reliable performance in real-world applications.
- An efficient, portable system for real-time rice disease detection, aiding farmers in timely and accurate disease management.

Impact: This research will contribute to the field of precision agriculture by providing a reliable and practical tool for early disease detection in rice crops. The outcomes could lead to improved crop management practices, reduced losses, and enhanced food security.

4. Materials and Methods

The dataset Rice Leaf Disease with Segmentation Labels [21], is designed for the task of image segmentation, specifically targeting the identification and segmentation of rice leaf diseases. The dataset is a curated collection derived from three publicly available datasets, specifically filtered and annotated to focus on three types of rice leaf diseases: rice bacterial blight, rice blast, and brown spot. For each disease category, the dataset includes 150 original images of diseased rice leaves, as well as their corresponding segmentation masks that provide semantic information essentially labelling the diseased areas in the images.
The dataset consists of two main components:

4.1. Contents

Images: The dataset contains RGB images of rice leaves captured in different conditions. These images showcase leaves affected by various diseases, including but not limited to bacterial blight, brown spot, and blast. The images typically have different resolutions, though they may be normalized or resized for consistent input into machine learning models.

4.2. Segmentation Masks

Each image has an associated segmentation mask that labels the pixels in the image as either part of a diseased area or a healthy area. The masks are usually in grayscale, where one pixel value (e.g., 0) might represent healthy leaf tissue, and another pixel value (e.g., 1) represents diseased tissue.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 793-813, 2024
DOI: 10.55214/25768484.v8i6.2167

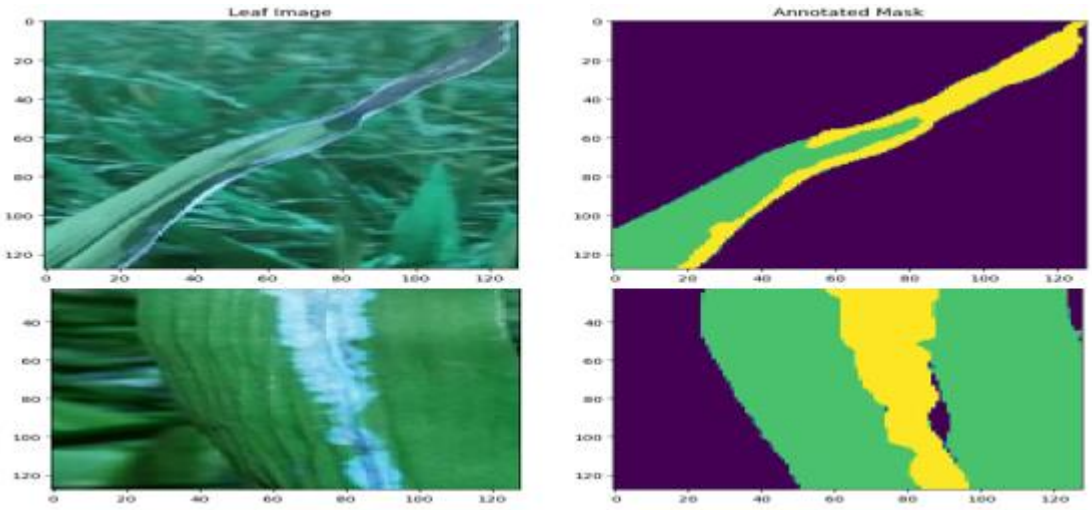

Figure.1.
(A)Bacterial blight.

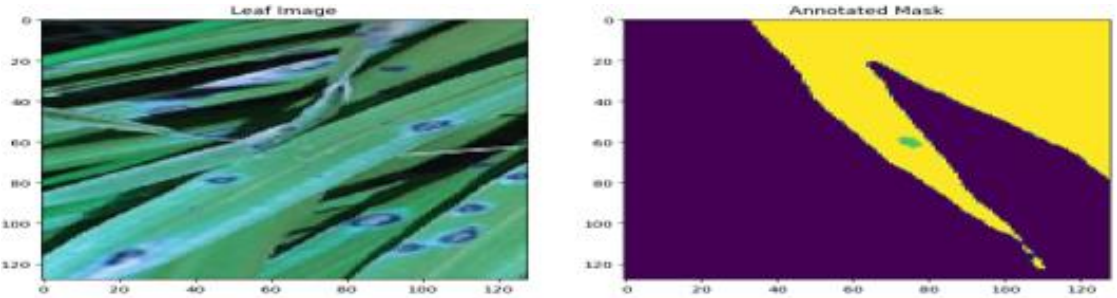

Figure 1.
(B) Brown spot.

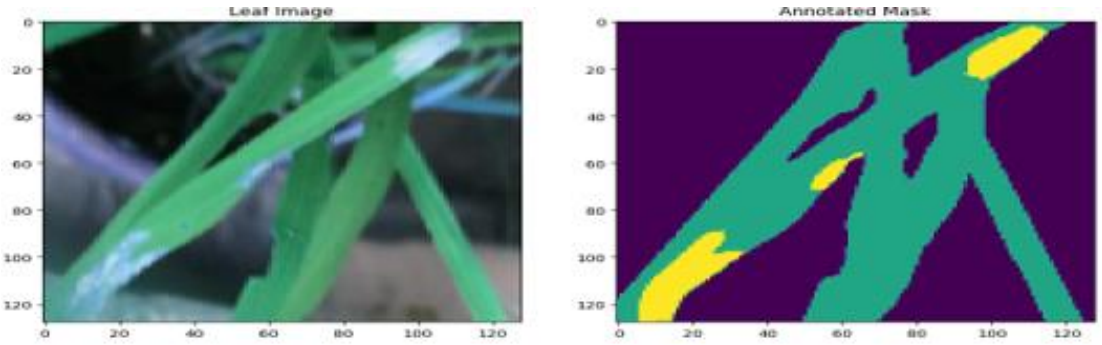

Figure 1.
(C) Blast.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 793-813, 2024
DOI: 10.55214/25768484.v8i6.2167

5. Data Preparation

In this data preparation process, the goal was to effectively prepare a small dataset of rice blast images and their corresponding masks for training a deep learning model in image segmentation. The steps involved were crucial for ensuring that the dataset was ready for model training, especially considering the limited number of images available.

5.1. Loading and Preprocessing Images and Masks

- Image and Mask Loading: Images and masks were loaded from specified directories. Each image was resized to a target size of 128x128 pixels to standardize the input dimensions for the model. Masks were converted to grayscale and reshaped to match the image dimensions.
- Splitting Data into Training and Validation Set: The dataset was split into training and validation sets, with 20% of the data allocated to validation. This split helps in monitoring the model's performance on unseen data during training, which is essential for assessing its generalization ability.

5.2. Data Augmentation

To artificially expand the training dataset and make the model more robust to variations, several data augmentation techniques were applied. The image augmentation process involved several transformations to enhance the dataset's variability and robustness. Images were rotated randomly within a 10-degree range to account for variations in orientation. Shifting transformations were applied to both the width and height, allowing for displacements of up to 10% in either direction. Additionally, a shearing transformation within a 10% range was introduced to simulate distortions in the images. Zooming was also performed, with images being randomly zoomed in or out by up to 10%. To further diversify the dataset, horizontal flipping was employed. Lastly, when these transformations caused pixels to move beyond the image boundaries, the missing areas were filled using the nearest fill mode, ensuring continuity in the augmented images.4. Batch Generation for Training.

- Batch Size: The images and masks were processed in batches of 8, which balances the memory requirements and the convergence speed during training.
- Generator Setup: Two separate ImageDataGenerator objects were used, one for images and one for masks, both applying the same augmentation transformations to ensure that the corresponding image and mask pairs remain aligned. The generators were then zipped together to create batches of augmented images and their corresponding masks for training.

Validation Data Preparation For validation, the images and masks were not augmented. This ensures that the validation set remains a reliable measure of how well the model is likely to perform on unseen, real-world data without artificial alterations.

6. Proposed Architectures

Algorithm: U-Net with MobileNetV2 Encoder and Pix2Pix Decoder [22] [23]

1. Input Initialization:
    - Define the input size for the model as (128, 128, 3).
2. Encoder (Feature Extraction):
    - Load the MobileNetV2 model pre-trained on ImageNet with include_top=False to exclude the final fully connected layer.
    - Set the input shape to match the specified input size.
    - Identify key layers from MobileNetV2 to use as skip connections:
        - Block_1_expand_relu (output size: 64x64)
        - Block_3_expand_relu (output size: 32x32)
        - Block_6_expand_relu (output size: 16x16)

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 793-813, 2024
DOI: 10.55214/25768484.v8i6.2167

        - Block_13_expand_relu (output size: 8x8)
        - block_16_project (output size: 4x4)
    - Extract the outputs of these layers to form the encoder's outputs.
3. Define the Upsampling Path (Decoder):
    - Create an upsampling path using the Pix2Pix upsampling technique:
        - Apply 4 upsampling layers:
            1. Upsample to transition from 4x4 to 8x8.
            2. Upsample to transition from 8x8 to 16x16.
            3. Upsample to transition from 16x16 to 32x32.
            4. Upsample to transition from 32x32 to 64x64.
    - Each upsampling layer increases the spatial resolution while reducing the number of feature channels.
4. Forward Pass through the Encoder:
    - Pass the input through the encoder to extract features at multiple scales.
    - Save the outputs from the specified layers for later use in skip connections.
5. Forward Pass through the Decoder:
    - Initialize the decoder with the last output from the encoder (smallest resolution).
    - Sequentially pass through each upsampling layer in the decoder:
        - Concatenate the upsampled output with the corresponding encoder output from the skip connections.
6. Final Output Layer:
    - Apply a Conv2DTranspose layer with a stride of 2 to double the spatial resolution from 64x64 to 128x128.
    - Use a sigmoid activation function on the final output to predict pixel-wise probabilities for binary segmentation.
7. Model Compilation:
    - Compile the model with the Adam optimizer.
    - Use binary_crossentropy as the loss function, appropriate for binary segmentation tasks.
    - Include accuracy as a metric to monitor during training.
8. Model Summary:
    - Print a summary of the model architecture, showing the layers and their output shapes.

This algorithm describes a U-Net architecture for image segmentation that efficiently combines the lightweight MobileNetV2 as an encoder with a Pix2Pix-inspired upsampling decoder. The model is optimized for binary segmentation tasks and is designed to handle inputs with a resolution of 128x128 pixels. The final output is a segmentation mask with the same spatial resolution as the input image. This model leverages the efficiency of MobileNetV2 for feature extraction while employing a U-Net architecture with skip connections to retain spatial details during upsampling. By combining the strengths of both architectures, it creates a powerful tool for segmenting images, particularly useful in agricultural applications like identifying diseased areas in rice leaves. This segmentation model can help in early disease detection, potentially saving crops from severe damage and aiding in precision agriculture.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 793-813, 2024
DOI: 10.55214/25768484.v8i6.2167

Figure 2.
(a) U-Net with MobileNetV2 Encoder and Pix2Pix Decoder (b) Simple DeepLabV3+ Architecture Using EfficientNetB4.

6.1. Algorithm: Simple DeepLabV3+ Architecture Using EfficientNetB4

The function simple_deeplabv3plus(input_shape=(128, 128, 3), classes=1) defines a segmentation model inspired by the DeepLabV3+ architecture. This model combines the EfficientNetB4 architecture as a feature extractor with an upsampling path designed to produce high-resolution segmentation maps. Below is a

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 793-813, 2024
DOI: 10.55214/25768484.v8i6.2167

detailed explanation of how this model works, along with the technical details and its potential applications.[24][25]

6.1.1. EfficientNetB4 as the Base Model (Encoder):

- EfficientNetB4 is part of the EfficientNet family of models, which are designed to achieve high accuracy while being computationally efficient. These models scale width, depth, and resolution systematically to optimize performance.
- In this function, EfficientNetB4 serves as the encoder, extracting features from the input image at different levels of abstraction. This model is pre-trained on the ImageNet dataset, providing a strong starting point for feature extraction.

base_model = EfficientNetB4(input_shape=input_shape, include_top=False, weights='imagenet')

- input_shape=input_shape: The input size is specified as 128x128 pixels with 3 color channels (RGB).
- include_top=False: This excludes the fully connected layers, as they are not needed for the segmentation task.
- weights='imagenet': Pre-trained weights are loaded to leverage the learned features from the ImageNet dataset.
- Output Shape: The output from EfficientNetB4 will be a feature map of size (None, 4, 4, 1792), where 1792 is the number of feature channels.

6.1.2. Upsampling Path (Decoder):

- The upsampling path is designed to progressively restore the spatial dimensions of the feature maps back to the original input size (128x128 pixels) while reducing the number of channels.
- Upsampling Layers: The model uses UpSampling2D layers to increase the spatial resolution of the feature maps. Each upsampling step doubles the resolution.
- 4x4 to 8x8 Upsampling:
    - UpSampling2D((2, 2)) increases the resolution from 4x4 to 8x8.
    - Conv2D(512, (3, 3), padding='same', activation='relu') applies a convolution with 512 filters, preserving the resolution but refining the features.
    - BatchNormalization() normalizes the outputs to stabilize and accelerate the training process.
- 8x8 to 16x16 Upsampling:
    - Resolution is increased to 16x16, and the number of filters is reduced to 256.
- 16x16 to 32x32 Upsampling:
    - Resolution is increased to 32x32, with 128 filters.
- 32x32 to 64x64 Upsampling:
- 64x64 to 128x128 Upsampling:
    - Finally, the resolution is restored to the original input size of 128x128, with 32 filters.

6.1.3. Final Convolutional Layer

- After upsampling, a final convolutional layer is used to produce the desired number of output channels. For segmentation tasks, this often corresponds to the number of classes.
- Conv2D(classes, (1, 1), padding='same') applies a 1x1 convolution to reduce the

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 793-813, 2024
DOI: 10.55214/25768484.v8i6.2167

number of channels to the number of classes. In this case, classes=1, so the output is a single-channel feature map.

- Sigmoid Activation:
  - The sigmoid activation function is used to produce output values between 0 and 1, which is suitable for binary segmentation tasks.

6.1.4. Model Compilation

- The model is compiled by connecting the input of the EfficientNetB4 encoder to the output of the upsampling decoder.
- This returns a Keras model that takes an input image of size 128x128 and produces a segmentation map of the same size.

6.2. Technical Summary and Use Cases

1. EfficientNetB4: EfficientNetB4 is chosen as the backbone for its balance between performance and computational efficiency. It extracts rich features from input images while maintaining a relatively small model size.
2. Upsampling Path: The upsampling path progressively restores the spatial resolution of the feature maps, using Conv2D layers to refine the features at each step. BatchNormalization is applied after each convolution to improve training stability and model performance.
3. Final Output: The final convolution layer reduces the feature maps to a single channel, and the sigmoid activation function produces a probability map, indicating the presence or absence of a specific class (e.g., diseased region) in each pixel.

7. Proposed Efficient Architecture-U-Net with EfficientNetB7 and Attention Gates-

The proposed model, UnetEfficientNetB7WithAttention, combines several advanced deep learning techniques to create a powerful architecture for image segmentation tasks. Below is a comprehensive breakdown of the model's components, how they work together, and the significance of each element.[26][27].

7.1. EfficientNetB7 as the Backbone (Encoder)

The model utilizes EfficientNetB7 as the backbone for feature extraction. EfficientNetB7 is part of the EfficientNet family, which is known for achieving state-of-the-art accuracy with fewer parameters by scaling up the network width, depth, and resolution in a balanced manner.

- Input_shape=input_shape: Specifies the input size (128x128x3), where 3 corresponds to the RGB channels.
- Include_top=False: Excludes the fully connected layers at the top of the network, as these are not needed for segmentation.
- Weights='imagenet': Uses pre-trained weights from ImageNet, which helps in leveraging features learned from a large and diverse dataset.

7.2. Skip Connections

Skip connections are extracted from specific layers of EfficientNetB7. These connections allow the model to retain and reuse high-resolution spatial information, which is crucial for accurate segmentation.

- These layers are selected because they represent different stages of down sampling and feature extraction, each providing a different level of detail.

7.3. Decoder Path with Attention Gates

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 793-813, 2024
DOI: 10.55214/25768484.v8i6.2167

The decoder path is designed to up sample the feature maps back to the original image size while integrating the skip connections. Attention gates are introduced at each skip connection to focus on the most relevant features.

7.3.1. Attention Gates

Attention gates are used to refine the skip connections by focusing on the most relevant spatial features before concatenating them with the decoder features. This helps in improving the segmentation accuracy by highlighting important regions in the feature maps.

- **Theta_x** and **phi_g**: These are feature transformations of the input x (skip connection) and gating signal g (decoder feature) respectively, which are then added together.
- **Upsampled_theta_x**: Upsamples the transformed skip connection to match the size of the gating signal.
- **Psi**: A sigmoid activation is applied to create an attention map, which is then used to weight the original skip connection features.
- **Final Multiplication**: The original skip connection features are multiplied by the attention map, effectively focusing on the most relevant features.

7.3.2. Decoder Operations

The decoder upsamples the features step-by-step, concatenating the attention-refined skip connections at each stage to recover spatial resolution.

- **Conv2D Layers**: These layers refine the upsampled features by applying convolutions.
- **BatchNormalization**: Stabilizes and accelerates training by normalizing the output of each convolutional layer.
- **UpSampling2D**: Doubles the spatial dimensions of the feature maps.

This process is repeated for each skip connection until the feature map is upsampled back to the original input size.

7.4. Final Output Layer

The final layer uses a 1x1 convolution to reduce the feature map to the desired number of output classes and applies a sigmoid activation for binary segmentation.

- **Conv2D(num_classes, (1, 1))**: Reduces the channel dimension to num_classes (typically 1 for binary segmentation).
- **Activation('sigmoid')**: Outputs a probability map indicating the likelihood of each pixel belonging to the target class.

7.5. Model Compilation

The model is then compiled into a Keras model with inputs from the EfficientNetB7 encoder and outputs from the final upsamling layer.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 793-813, 2024
DOI: 10.55214/25768484.v8i6.2167

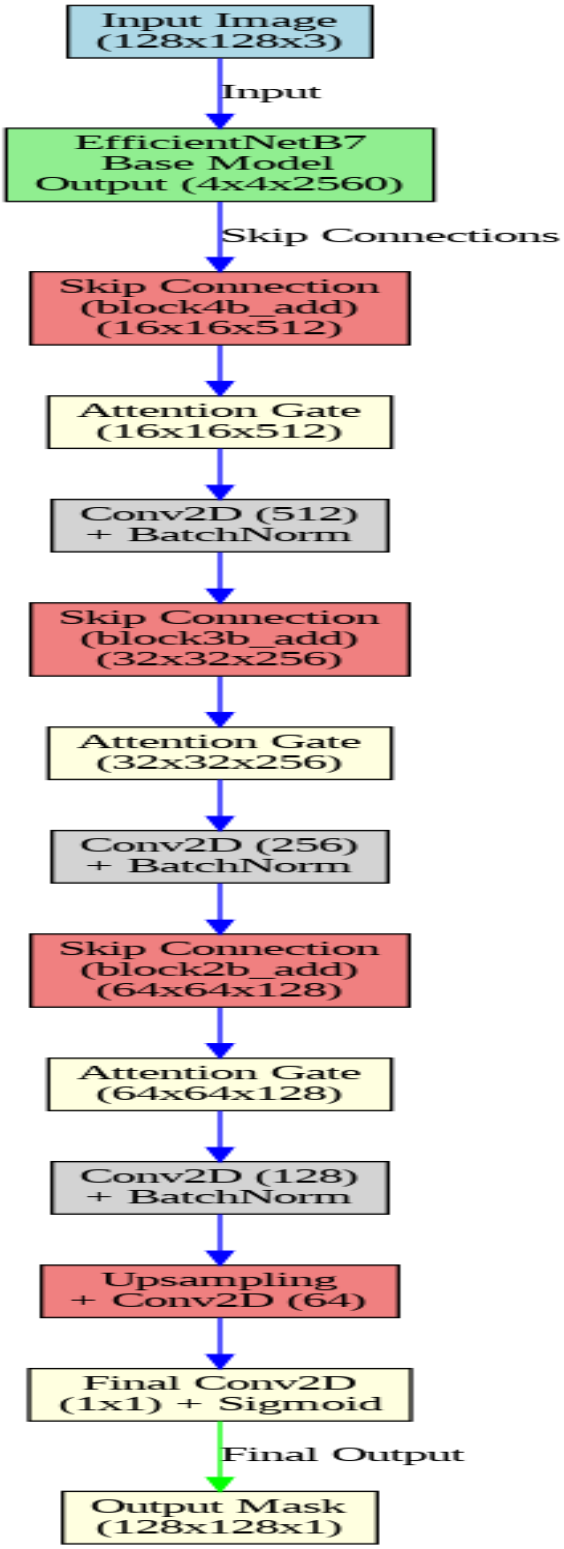

Figure 3.
U-Net with efficientNetB7 and attention gates.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 793-813, 2024
DOI: 10.55214/25768484.v8i6.2167

this model represents a significant advancement in the field of image segmentation, particularly for applications like medical imaging, remote sensing, and precision agriculture. Here's why:

1. Use of EfficientNetB7: Leveraging EfficientNetB7 as the backbone provides a powerful and efficient feature extractor, offering state-of-the-art performance with fewer parameters.
2. Attention Gates: The integration of attention gates at each skip connection is a novel approach to focus on the most relevant features, improving the model's ability to differentiate between important and less important information in the feature maps.
3. Advanced Decoder: The combination of upsampling layers, convolutional layers, and attention gates in the decoder path ensures that the model can accurately reconstruct high-resolution segmentation maps.
4. Generalization: The use of EfficientNetB7 with attention mechanisms can generalize well to various segmentation tasks, making it a versatile tool for different applications.
5. State-of-the-Art Performance: Given the architecture's components and their combination, this model is likely to achieve high accuracy in segmentation tasks while being computationally efficient, which is crucial for real-time applications.

8. Results & Discussions

8.1. MODEL-I U-Net Model with MobileNetV2 Encoder

Based on the training results of the MobileNetV2 model, here's a summary of the model's performance over 100 epochs on blast rice diseases images that include 150 images of true mask and actual image:

8.1.1. Training and Validation Performance

- Initial Performance:
  - The model began with a relatively strong starting point in terms of accuracy and loss. The first epoch achieved an accuracy of 79.8% and a loss of 0.1487, with validation accuracy at 77.7% and validation loss at 0.1955.
- Fluctuations:
  - Throughout the training process, both training and validation accuracy exhibited fluctuations. For instance, training accuracy ranged between 70% and 80%, while validation accuracy varied similarly, occasionally dipping into the 70% range before recovering.
  - Loss values also fluctuated, with periods where the training loss decreased, followed by increases, indicating potential challenges in learning stability.
- Midway Improvements:
  - Around epoch 14, the model reached a notable improvement, with validation accuracy peaking at 81.5% and validation loss dropping to 0.1722, indicating better generalization at that point.
- Overfitting Signs:
  - From the later epochs (around epoch 40 onwards), there are signs of overfitting. The validation loss occasionally increased while training accuracy remained relatively high, suggesting that the model was becoming too specialized to the training data.
  - Particularly, validation loss values increased significantly in some epochs, for example, during epoch 38 (loss = 0.4128), showing poor generalization in those phases.
- Final Performance:
  - By the end of the training, the model managed to achieve a decent performance with validation accuracy reaching as high as 88.25% in epoch 43. However, there were considerable variations in validation loss, suggesting that while the model could reach high accuracy, its predictive stability was inconsistent.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 793-813, 2024
DOI: 10.55214/25768484.v8i6.2167

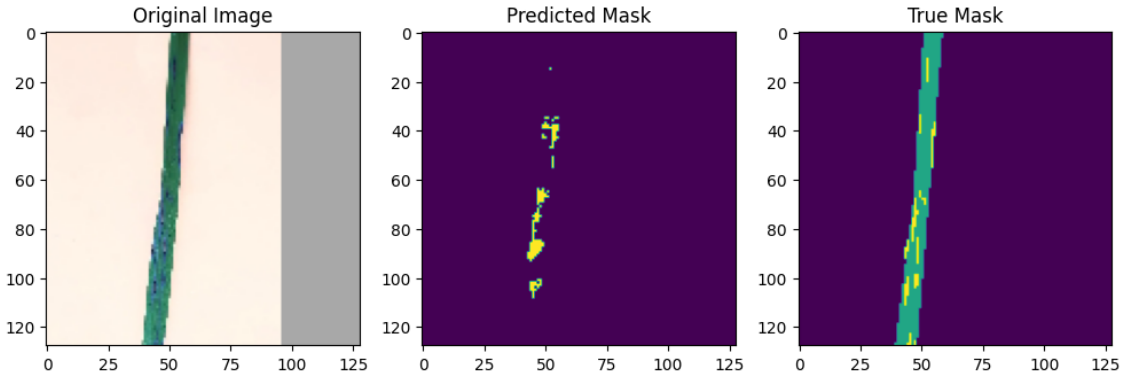

Figure 4.
Blast (Original image, predicted mask, true mask) model-I.

- Final Training Accuracy: The model achieved a final accuracy of 78.95% on the training data, with a corresponding loss of 0.1909.
- Final Validation Performance:
    - Validation Loss: The model's overall validation loss was 0.2052, which indicates the average difference between the predicted and actual outputs on the validation set.
    - Validation Accuracy: The model's overall validation accuracy was 75.95%, reflecting its ability to generalize to unseen data.

The MobileNetV2 model demonstrated a decent overall performance with a training accuracy of 78.95% and a validation accuracy of 75.95%. The slightly higher training accuracy compared to validation accuracy suggests some overfitting, but the model performed reasonably well on the validation set, indicating good generalization with some room for improvement.

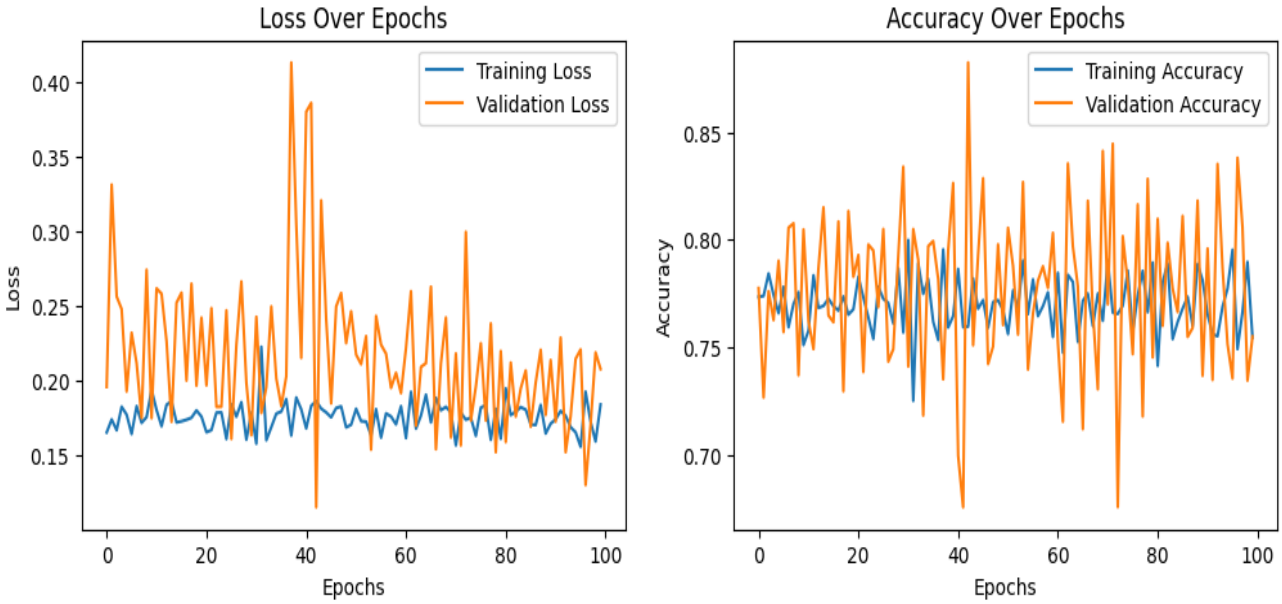

Figure 5.
Loss and accuracy graph.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 793-813, 2024
DOI: 10.55214/25768484.v8i6.2167

8.2. MODEL-II of the Simple DeepLabV3+ Architecture Using EfficientNetB4

8.2.1. Initial Observations

1. Epoch 1: The model starts with an accuracy of 48.09% on the training set and 61.79% on the validation set, with corresponding loss values of 0.7419 and 0.8259, respectively.
2. Overfitting Concerns: Early in the training (around Epoch 2 to Epoch 4), the validation loss increases significantly (up to 2.0053) while the validation accuracy drops, indicating possible overfitting.

8.2.2. Mid Training Observations

3. Improvement Phase: By Epoch 7, the model shows substantial improvement, with a validation accuracy of 85.9% and a reduced loss of 0.3456. This suggests that the model is learning to generalize better.
4. Fluctuations: Throughout the training, the validation accuracy fluctuates. For example, around Epoch 21, there is a dip in performance with a validation accuracy of 77.74% and an increase in loss to 0.3079.

8.2.3. Final Observations

5. Final Epochs: By Epoch 47, the model reaches a validation accuracy of 84.91% with a validation loss of 0.1500, indicating strong performance.

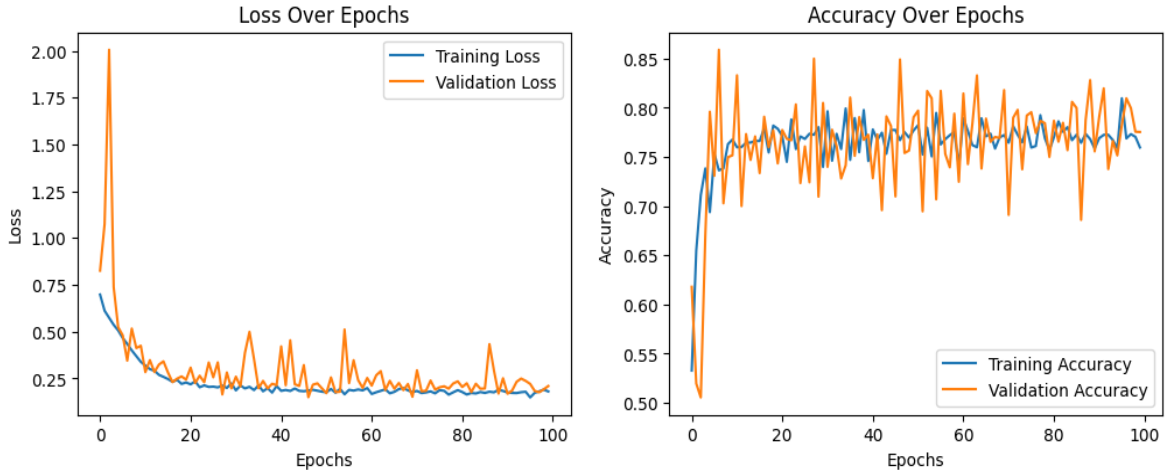

Figure 6.
Loss and accuracy graph.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 793-813, 2024
DOI: 10.55214/25768484.v8i6.2167

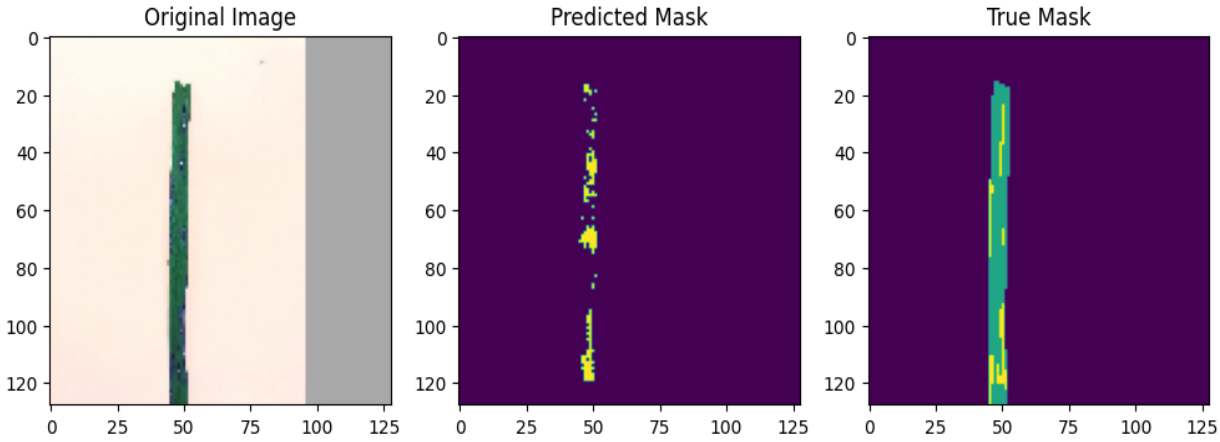

Figure 7.
Blast (Original Image, Predicted Mask, True Mask) Model-II.

The model shows a general trend of improving accuracy and reducing loss on both the training and validation datasets, with some fluctuations. The final validation accuracy of 84.91% suggests that the model is performing quite well, though there may still be room for fine-tuning to minimize fluctuations and further improve performance.

8.3. MODEL III -Results of Proposed Unet Efficient NetB7With Attention Model

The training log of the UnetEfficientNetB7WithAttention model reveals several insights into its performance over the course of 48 epochs:

8.3.1. Accuracy and Loss Trends

- **Training Accuracy**: The model's accuracy improved significantly from 0.5361 in the first epoch to values exceeding 0.8 by epoch 43. This indicates that the model is learning effectively over time.
- **Validation Accuracy**: Similarly, validation accuracy also showed an upward trend, peaking around 0.8431 in epoch 44. However, there were fluctuations, such as a drop in epoch 21 (0.2946), which might suggest overfitting or noisy validation data.
- **Training Loss**: The loss consistently decreased, signifying that the model is getting better at minimizing errors during training.
- **Validation Loss**: The validation loss was more volatile, with significant spikes (e.g., epoch 2 at 6.9256 and epoch 28 at 5.5382). This could indicate overfitting, where the model performs well on training data but struggles with unseen data.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 793-813, 2024
DOI: 10.55214/25768484.v8i6.2167

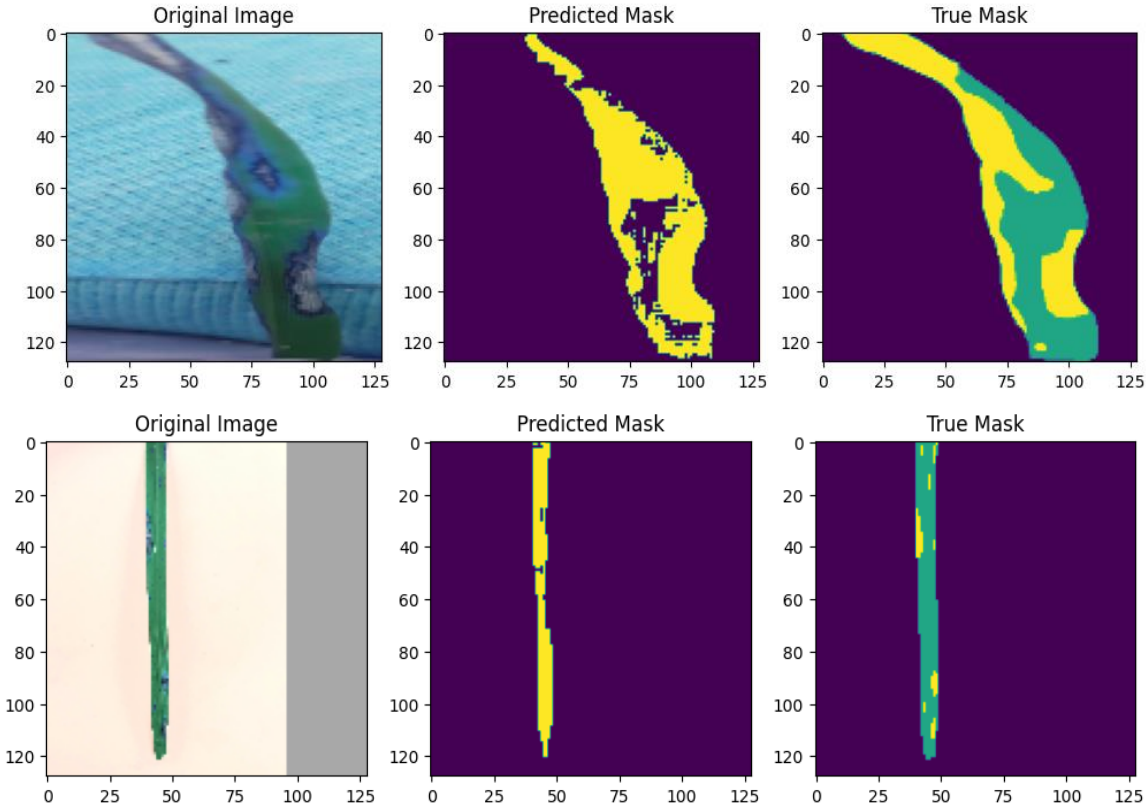

Figure 8.
Blast (Original Image, Predicted Mask, True Mask) Model-III.

8.4. Model Performance Over Time

- Early Improvement: From epoch 1 to epoch 8, both training and validation accuracies increased steadily, and losses decreased, indicating effective learning.
- Mid Training: Around epochs 20-30, there were some fluctuations in validation metrics, with notable drops in accuracy and spikes in loss, suggesting possible overfitting or encountering difficult validation batches.
- Later Epochs: Despite fluctuations, the model continued to improve overall, reaching a high validation accuracy of 0.8432 in epoch 48 with a corresponding decrease in validation loss to 0.1261, indicating good generalization by the end of the recorded epochs.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 793-813, 2024
DOI: 10.55214/25768484.v8i6.2167

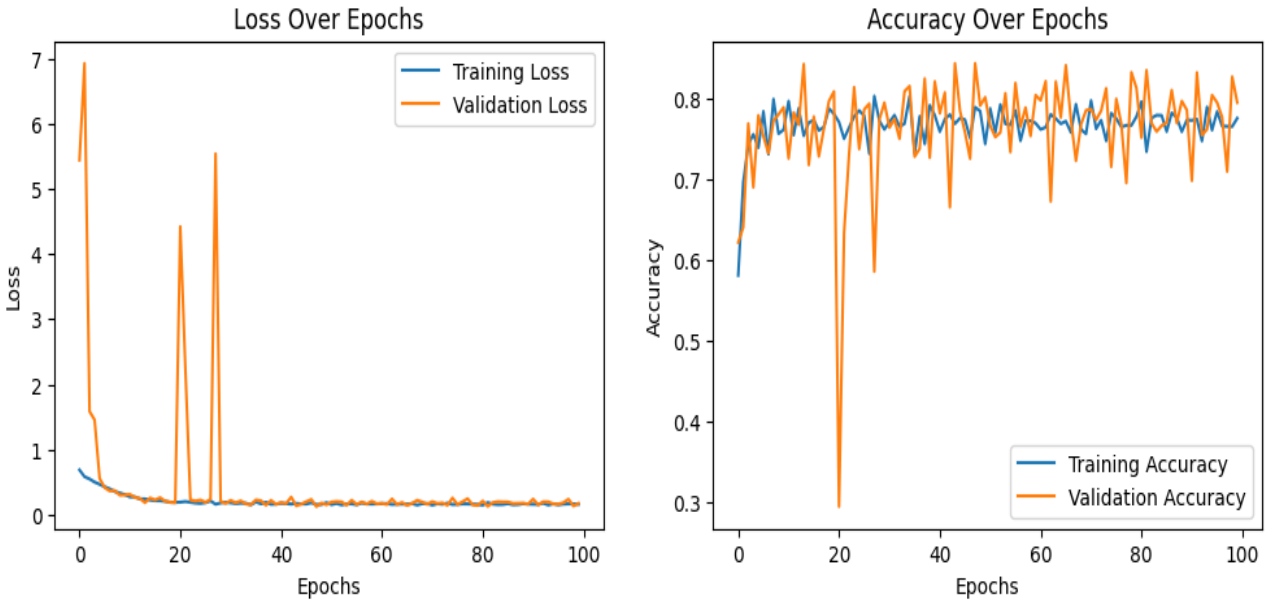

Figure 9.
Loss and accuracy graph.

Here is a comparative table summarizing the results of the three image segmentation models:

Table 1.
Comparative summary of proposed models.

Model	Validation loss	Validation accuracy	Efficiency	Complexity	Segmentation quality
U-Net with MobileNetV2	0.2052	0.7595	Highest efficiency; ideal for real-time use	Moderate; lightweight and fast	Good for simpler tasks; may struggle with complex details
DeepLabV3+ with EfficientNetB4	0.2342	0.7879	Moderate efficiency; balanced performance	Higher than MobileNetV2; efficient and powerful	Superior to MobileNetV2; handles complex boundaries well
U-Net with EfficientNetB7 and Attention Gates	0.172	0.8028	Lowest efficiency; high computational cost	Highest; most powerful and complex	Best-in-class; excels at fine details and complex segmentation

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 793-813, 2024
DOI: 10.55214/25768484.v8i6.2167

This table provides a clear comparison of the three models based on their validation results, efficiency, complexity, and segmentation quality

9. Conclusions and Future Plans

This analysis focuses on three advanced models for image segmentation, evaluating their architecture, efficiency, prediction accuracy, segmentation performance, and comparative results after 100 epochs of training. The models are:

1. U-Net with MobileNetV2
2. DeepLabV3+ with EfficientNetB4
3. U-Net with EfficientNetB7 and Attention Gates

9.1. Model Architecture and Components

- U-Net with MobileNetV2:
    - Encoder: MobileNetV2, optimized for efficiency.
    - Decoder: Standard U-Net decoder with skip connections.
    - Complexity: Moderate, with a focus on lightweight and fast performance.
    - Key Features: Efficient for mobile and resource-constrained environments, but may sacrifice some accuracy.
- DeepLabV3+ with EfficientNetB4:
    - Encoder: EfficientNetB4, designed for a balance between performance and efficiency.
    - Decoder: Convolutional upsampling layers.
    - Complexity: Higher than U-Net with MobileNetV2, balancing feature extraction and efficiency.
    - Key Features: Strong backbone and optimized for accurate segmentation.
- U-Net with EfficientNetB7 and Attention Gates:
    - Encoder: EfficientNetB7, the most powerful of the three.
    - Decoder: U-Net-style upsampling with attention gates.
    - Complexity: Highest, with attention mechanisms to enhance feature focus.
    - Key Features: High accuracy due to attention gates, but at the cost of increased computational requirements.

9.2. Efficiency (Computational Cost and Speed)

- U-Net with MobileNetV2:
    - Efficiency: Highest due to the lightweight backbone, ideal for real-time and resource-constrained applications.
    - Computational Cost: Low, making it suitable for scenarios where computational resources are limited.
- DeepLabV3+ with EfficientNetB4:
    - Efficiency: Moderate, balancing performance with computational cost.
    - Computational Cost: Moderate, making it feasible for more demanding applications.
- U-Net with EfficientNetB7 and Attention Gates:
    - Efficiency: Lowest due to the complexity of the model.
    - Computational Cost: High, making it less ideal for real-time applications unless on high-performance hardware.

9.3. Prediction Accuracy

- **U**-Net with MobileNetV2:
    - Accuracy: Good, particularly in simpler tasks or where efficiency is prioritized.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 793-813, 2024
DOI: 10.55214/25768484.v8i6.2167

    - Validation Accuracy: 0.7595
- DeepLabV3+ with EfficientNetB4:
    - Accuracy: Higher accuracy than MobileNetV2, suitable for more complex tasks.
    - Validation Accuracy: 0.7879
- U-Net with EfficientNetB7 and Attention Gates:
    - Accuracy: Highest accuracy due to attention gates and advanced feature extraction.
    - Validation Accuracy: 0.8028

4. Segmentation Performance

- U-Net with MobileNetV2:
    - Segmentation Quality: Good, but may struggle with fine details in complex images.
    - Validation Loss: 0.2052
- DeepLabV3+ with EfficientNetB4:
    - Segmentation Quality: Superior to MobileNetV2, with better handling of complex boundaries.
    - Validation Loss: 0.2342
- U-Net with EfficientNetB7 and Attention Gates:
    - Segmentation Quality: Best-in-class, excelling in distinguishing subtle details.
    - Validation Loss: 0.1720

9.4. Comparison with Earlier Work

The three models discussed represent significant advancements over earlier image segmentation models that typically relied on simpler backbones like VGG or ResNet. These earlier models often required a trade-off between accuracy and computational efficiency.

- **Advancements**:
    - **EfficientNet Integration**: Offers a strong balance between accuracy and resource use.
    - **Attention Mechanisms**: In U-Net with EfficientNetB7, attention gates enhance segmentation quality by focusing on relevant features.

10. Conclusion and Recommendations

- **Best for Efficiency**: **U-Net with MobileNetV2** is recommended for scenarios where efficiency and computational cost are paramount, such as mobile applications.
- **Best for Balance**: **DeepLabV3+ with EfficientNetB4** provides a good compromise between accuracy and efficiency, making it suitable for general-purpose applications.
- **Best for Accuracy**: **U-Net with EfficientNetB7 and Attention Gates** is ideal for high-stakes, precision-critical tasks, although it demands more computational resources.

Overall, these models demonstrate impressive capabilities in modern image segmentation tasks, each excelling in different aspects depending on the specific requirements of the application. The choice of model should align with the balance of accuracy, efficiency, and computational resources available.

Challenges and Future Directions: While significant progress has been made in the field of image segmentation for leaf diseases, challenges remain. The variability in environmental conditions, the presence of noise in images, and the need for large, annotated datasets for training deep learning models are ongoing concerns. Future research is likely to focus on developing more generalized models that can perform well across different conditions, improving the computational efficiency of these models, and integrating segmentation techniques with other data sources, such as meteorological data, to enhance disease prediction and management.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 793-813, 2024
DOI: 10.55214/25768484.v8i6.2167

References
[1] T. Archana and A. Sahayadhas, "Automatic Rice Leaf Disease Segmentation: Detection and Segmentation of Bacterial Leaf Blight and Brown Spot in Rice Leaves under Varying Environmental Conditions," *International Journal of Image Processing and Vision Sciences*, 2018.
[2] P. K. Sethy, A. K. Rath, and S. K. Behera, "A Survey on Image Processing Techniques for Rice Plant Disease Detection and Classification," *Journal of Agricultural Engineering*, 2020.
[3] C. Thanawiparat, N. Iam-On, and V. Boonjing, "Improving Rice Disease Detection Using Deep Learning and Level Set Methods," *Expert Systems with Applications*, 2023.
[4] S. M. Abdullah, et al., "Bi-Level Thresholding for Disease Diagnosis: A Portable Tool for Predicting Rice Diseases," *Proceedings of the International Conference on Computer Vision and Graphics*, 2016.
[5] D. M. Mai and Q. Meng, "Lesion Segmentation in Rice Leaf Blast Images Using Superpixel Segmentation and Random Forest Classifiers," *Biosystems Engineering*, 2016.
[6] P. Devi and M. Muthukannan, "Comparison of Image Segmentation Methods for Identifying Diseased Rice Leaves," *International Journal of Computer Applications*, 2014.
[7] Y. Jiang, et al., "Machine Vision for Disease Spot Extraction in Rice Leaves: A Gray-Level Transformation Approach," *Computers and Electronics in Agriculture*, 2015.
[8] F. G. Pascual, et al., "Automated Detection of Rice Diseases Using SVM and Random Forest Classifiers," *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2019.
[9] X. Li, et al., "Copy Paste and Semantic Segmentation for Accurate Disease Region Segmentation and Severity Assessment in Rice Leaves," *International Conference on Image Processing (ICIP)*, 2022.
[10] T. Daniya and S. Vigneshwari, "Texture and Deep Features for Disease Detection in Rice Plants: A Deep Learning Approach Using SegNet and Deep RNN," *IEEE Access*, 2021.
[11] R. Krishnamoorthy, et al., "Transfer Learning with CNN for Predicting Rice Leaf Diseases," *International Journal of Computational Intelligence and Applications*, 2021.
[12] L. Pothen and M. M. Pai, "Otsu's Method Combined with Local Binary Patterns and Histogram of Oriented Gradients for Rice Disease Classification," *Pattern Recognition Letters*, 2020.
[13] S. Zhou, et al., "Fusion of FCM-KM Clustering and Faster R-CNN for Rapid Detection of Rice Diseases," *IEEE Transactions on Geoscience and Remote Sensing*, 2019.
[14] Y. Lu, et al., "Deep CNN for Identifying 10 Common Rice Diseases: An Evaluation of Deep Learning Approaches," *Computers and Electronics in Agriculture*, 2017.
[15] J. Chen, et al., "Attention Networks for Rice Disease Detection: Using Lightweight Models Based on MobileNet-V2," *Computers and Electronics in Agriculture*, 2021.
[16] C. Orillo, et al., "Backpropagation Neural Network for Detecting Three Common Rice Diseases," *International Journal of Computer Applications*, 2014.
[17] V. S. Shah, et al., "A Survey on Rice Disease Detection Using Image Processing and Machine Learning Techniques," *Journal of Computational and Theoretical Nanoscience*, 2016.
[18] N. S. Patil and A. Kumar, "Rice-Fusion: A Multimodality Data Fusion Framework for Rice Disease Diagnosis," *Journal of Agricultural and Food Information*, 2022.
[19] S. S. Chawathe, "Image Analysis for Rice Disease Detection Using Leaf and Lesion Attributes," *Proceedings of the International Conference on Image Processing and Vision Science*, 2020.
[20] P. Joshi and R. Jadhav, "An Automated System for Monitoring and Controlling Rice Diseases Using Image Processing Techniques," *International Journal of Computer Applications*, 2016. [Online]. Available: https://www.kaggle.com/datasets/slygirl/rice-leaf-disease-with-segmentation-labels.
[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
[22] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.
[23] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with AtrousSeparable Convolution for Semantic Image Segmentation," *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
[24] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.
[25] O. Oktay, J. Schlemper, L. Le Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, "Attention U-Net: Learning Where to Look for the Pancreas," *arXiv preprint arXiv:1804.03999*, 2018.