# Semantic Segmentation of Biotic Stress in Coffee Leaves using U-Net and DeepLabV3: A Comparative Study

Cristian Granchelli
cristian.granchelli@studenti.unipd.it

Cameron Kelahan
cameronmatthew.kelahan@studenti.unipd.it

GitHub Repository:
https://github.com/cameronkelahan/VisCogRep

## Abstract

*Biotic stress in coffee plants, caused by various pests and pathogens, poses a significant threat to agricultural productivity and sustainability. Accurate and early identification of such stress can aid in timely intervention and mitigation. This study explores the application of deep learning techniques for the semantic segmentation of biotic stress in coffee leaves. A dataset comprising 400 preprocessed images of coffee leaves, both healthy and stressed, was utilized. Two state-of-the-art semantic segmentation models, U-Net and DeepLabV3, were employed to perform the segmentation tasks. Performance metrics including pixel accuracy, DICE score, and Intersection over Union (IoU) were collected and analyzed. This comparative study provides insights into the strengths and limitations of each model, contributing to the development of robust diagnostic tools for coffee plant health monitoring.*

## 1. Introduction

Biotic stress refers to the negative impact on plants caused by living organisms such as pests, pathogens (including viruses, fungi, and bacteria), and weeds. This type of stress can lead to significant damage, reducing the yield and quality of crops. Accurate and early identification of biotic stress is crucial for effective management and mitigation, helping to ensure agricultural productivity and sustainability.

Semantic segmentation is a critical task in computer vision, involving the classification of each pixel in an image into predefined categories. In the context of agriculture, semantic segmentation enables precise identification and localization of diseased or stressed regions in plant images, facilitating early intervention and treatment. By segmenting the symptoms of biotic stress, it becomes possible to estimate the severity of the symptoms by calculating the ratio between the symptomatic area and the total leaf area. This quantitative analysis aids in assessing the extent of damage, guiding targeted interventions, and optimizing resource use to enhance crop health and productivity.

Deep learning has revolutionized the field of image analysis, with Convolutional Neural Networks (CNNs) at the forefront of this transformation. Among the myriad of CNN architectures, U-Net and DeepLabV3 have proven to be highly effective for semantic segmentation tasks. U-Net, originally designed for biomedical image segmentation, employs a symmetric encoder-decoder structure with skip connections, allowing it to capture both spatial and contextual information. DeepLabV3, on the other hand, utilizes atrous convolution and pyramid pooling to achieve high-resolution segmentation with reduced computational complexity. This study leverages the capabilities of U-Net and DeepLabV3 to address the challenge of segmenting biotic stress in coffee leaves.

We have curated a dataset comprising 400 preprocessed images of coffee leaves, both healthy and affected by biotic stress. The primary goal of this study is to compare the performance of U-Net and DeepLabV3 in segmenting biotic stress regions, providing insights into their practical applications in agricultural diagnostics.

We found that U-Net outperformed DeepLabV3 in all metrics for each class, achieving consistently high performance overall.

## 2. Related Work

Traditional methods for semantic segmentation in agricultural images primarily relied on techniques such as thresholding, clustering, wavelet transforms, and random forests. These methods have been fundamental in the early development of agricultural image analysis [11].

Thresholding methods, such as Otsu's method, segment images based on grayscale differences. Otsu's method [13] determines an optimal threshold to minimize intra-class variance, but it struggles with non-uniform lighting conditions commonly found in outdoor agricultural settings. Adaptive thresholding techniques, which compute local thresholds based on image regions, have been proposed to address this limitation.

Semantic segmentation methods based on clustering involve grouping pixel points with similar features into the same area through repeated iteration and clustering until convergence. Superpixels, which are blocks of neighboring pixels with similar features like color and texture, are used to represent image features more efficiently than individual pixels, preserving edge information and reducing post-processing complexity. A prominent algorithm for generating superpixels is Simple Linear Iterative Clustering (SLIC) [1], derived from the K-means clustering algorithm.

Deep learning methods have become prevalent in agricultural computer vision due to their superior performance in tasks like classification, detection, and semantic segmentation. Numerous papers ([16], [9], [7]) have investigated the use of large CNNs to identify plant diseases and damage. More complex approaches have started to emerge as well, such as in Li X. et al. [17] where they explored the use of both U-Net and DeepLabV3, as well as PSPNet, for semantic segmentation of potato leaves to identify different blights. Very recently, Alam et al. 2024 [2] desired to develop an agricultural disease segmentation program that avoided numerous drawbacks found in other deep learning approaches, such as data augmentation via translation, rotation, and flip, as well as an overall lack of generalization when applied to different backgrounds. They utilized two different General Adversarial Network (GAN) models, CycleGAN and Pix2Pix, to synthetically generate diseased potato leaf images created from genuine images of healthy potato leaves. This not only expanded the dataset size but it also improved generalization by increasing the variety of disease appearances. These images were fed to different CNN semantic segmentation models, yielding high accuracy when tested.

The emergence of more powerful deep learning models in recent years has bolstered their use in this field of crop-disease identification. The combined use of different model and architecture types is a promising approach to continue improving the ability of deep learning models to identify diseases in agricultural plants.

## 3. Dataset

The dataset used in this study is a subset derived from the dataset cited in the paper "Deep Learning for Classification and Severity Estimation of Coffee Leaf Biotic Stress" by José G. M. Esgario, Renato A. Krohling, and José A. Ventura [6]. This subset consists of 400 images of coffee leaves affected by biotic stresses such as leaf miner, rust, brown leaf spot, and cercospora leaf spot (see figure 1). The identification and labeling of the diseased foliage was performed by a specialist to ensure accuracy and legitimacy of the dataset.


Figure 1. *Examples of coffee leaves affected by different biotic stresses: (a) leaf miner, (b) rust, (c) brown leaf spot, and (d) cercospora leaf spot. [6]*

The images in this dataset were captured using various smartphones under partially controlled conditions, ensuring a heterogeneous collection that reflects real-world variability. The dataset includes both healthy leaves and those affected by one or more types of biotic stresses. Of the original 1,747 leaves, 372 leaves presented two or more different biotic stresses. 62 of these multi-stressed leaves suffered from these diseases to such a degree that distinguishing where one ended and another began were too difficult for the expert, and thus these samples were discarded. This left 1,685 leaves of differing health/disease types as broken down in table 1.

| Biotic Stress | # in Dataset |
|---|---|
| **Healthy** | 272 |
| **Leaf Miner** | 387 |
| **Rust** | 531 |
| **Brown Leaf Spot** | 348 |
| **Cercospora Leaf Spot** | 147 |
| **TOTAL** | 1,685 |

Table 1. Breakdown of the different types of images, and their amounts, found in the dataset.

Figure2 shows two example mask images alongside their original image counterpart. These masks were created using the methods outlined in Manso et al. 2019 [12], which are a combination of a version of k-means clustering and the Otsu thresholding technique [13]. These segmentation masks were then vetted by humans, comparing them to their original image by eye.

## 4. Methods

We chose to investigate two popular architectural approaches that were designed for the task of semantic segmentation: U-Net and DeepLabV3. By constructing these models from the ground up, following the papers they were originally published in, we explored why they perform this

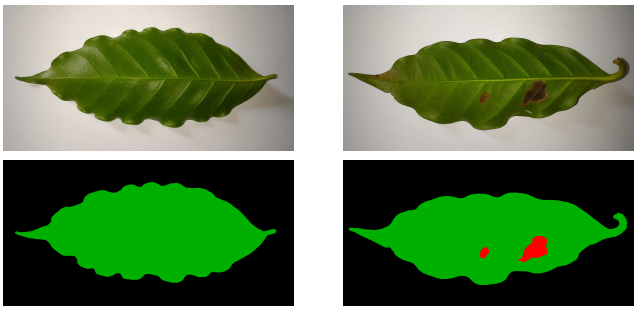

Figure 2.
*Top Left: example of a healthy leaf. Top Right: example of a leaf with biotic stress. Bottom Left: mask of the healthy leaf. Bottom right: mask of the leaf with biotic stress.* [6]

task well and how the two approaches compare to each other when applied to a specific project.

## 4.1. U-Net

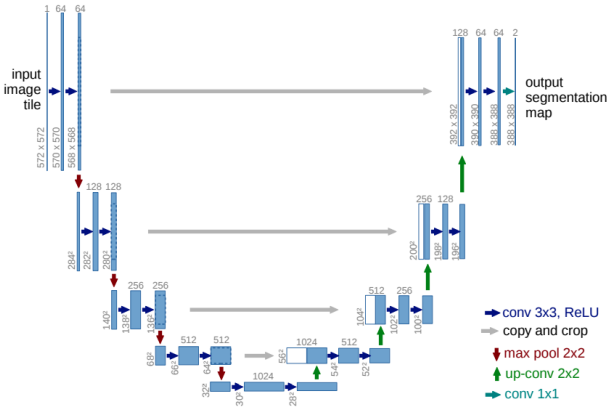


Figure 3. *U-Net Architecture example from Ronneberger et al. 2015 [14]. Clear example of the "U" shape from which it gets its name. Encoder left side, decoder right side, with skip connections shown as grey arrows.*

U-Net is a powerful, "fully convolutional" network that is able to achieve impressive results while avoiding common drawbacks of CNNs. CNNs have historically been used for image classification or object recognition, where the output is simply a label for what appears in the image. The original designers, Ronneberger et al. [14], aspired to create a model that could perform semantic segmentation on biomedical images, a genre that often suffers from small datasets of training images. At the time in 2015, the majority of CNN models relied on very large training datasets to perform well [10].

The idea of U-Net being "fully convolutional" comes from the idea that after performing standard down-sampling convolution (encoder) in its contraction steps, it also "deconvolutes" and up-samples (decoder) afterwards. While the encoder half of U-net extracts the larger context of the image, capturing high-level features, the decoder half attempts to recreate the image's spatial dimensions, achieving localization as it does, creating a symmetrical architecture. On top of this, U-Net makes use of skip connections, attaching the encoder/decoder counterparts in order to maintain high-resolution features (see figure 3 for an example U-Net architecture model).

While U-Net outperformed its predecessors by a significant margin when it was first created, especially on small datasets, the performance increase in terms of accuracy was counterbalanced by a great increase in computational complexity in both space and time.

### 4.1.1 Down-Sampling Encoder

During the encoding steps, double convolutional layers and max-pooling layers are repeated to learn intermediate features. These are two 3x3 convolutional layers followed by a 2x2 max-pooling layer. After the double convolution step, the ReLU activation function is applied element-wise to each of these intermediate features. This helps the model to capture high-level features from the images. Our implementation of U-Net begins with 1024x512x3 input images, convolving and pooling down to a 32x64x1024 bottleneck.

### 4.1.2 Up-Sampling Decoder

The extracted encoder's features are once again subjected to double 3x3 convolutional layers that are followed by a ReLU activation function, then an up-sampling 2x2 convolution which is used to restore the spatial resolution of the features.

### 4.1.3 Skip Connections

The saved features from the encoder are concatenated onto their symmetrical counterparts from the decoder. The idea behind this is to combine the beneficial behaviors of the encoder and decoder of extracting context about a specific part of an image (like the diseased part of a coffee leaf) and the pixel-precise labeling (the actual segmentation part).

## 4.2. DeepLabV3

DeepLabV3 [4] is a model developed by google for the task of semantic segmentation. It represents an evolution within the DeepLab series, significantly improving upon its predecessors by refining and optimizing atrous convolution techniques.

The architecture of DeepLabV3 consists of two main components: the backbone and the head. The backbone is responsible for extracting features from the input image, while the head processes these features to produce the final segmentation map. For the scope of this project we have

chosen to use ResNet50 as the backbone for DeepLabV3. ResNet50 has been pre-trained on the ImageNet1k dataset [15], which contains over a million images across a thousand classes.

#### 4.2.1 ResNet50

ResNet50 [8] is composed of a series of convolutional layers, batch normalization layers, and identity or projection shortcut connections, structured into four main stages. Each stage contains multiple residual blocks. Here is the architecture outline:

- **Initial layers**
  - A 7x7 convolutional layer with 64 filters and a stride of 2.
  - A 3x3 max-pooling layer with a stride of 2.
- **Residual blocks**
  The core of ResNet50 is made up of 16 residual blocks, each containing three convolutional layers (1x1, 3x3, and 1x1 convolutions) and a shortcut connection that bypasses these layers.
- **Stages**
  - Stage 1: 3 residual blocks, each with 256 filters.
  - Stage 2: 4 residual blocks, each with 512 filters.
  - Stage 3: 6 residual blocks, each with 1024 filters.
  - Stage 4: 3 residual blocks, each with 2048 filters.
- **Final layers**
  The original architecture of the ResNet model ends with a global pooling layer and a fully connected layer that we removed to attach the DeepLabV3 head.

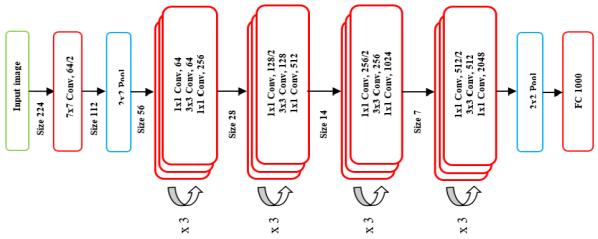


Figure 4. *ResNet50 architecture [3]*

#### 4.2.2 DeepLabV3 head

DeepLabV3's head is made of two main building blocks:

- **Atrous Convolution**
  Atrous convolution, also known as dilated convolution, is a technique that expands the kernel by inserting holes between its consecutive elements. By introducing gaps, atrous convolutions effectively enlarge the receptive field of the filter without increasing its size or the number of parameters. This means the filter can cover a larger area of the input image, capturing more context. The dilation rate controls the space between the values in the kernel, if the dilation rate is equal to one, we obtain a standard convolution.
- **Atrous Spatial Pyramid Pooling**
  The ASPP module processes the feature maps produced by the backbone network. The ASPP module employs several parallel atrous convolutions with different dilation rates, allowing it to capture features at multiple scales. In addition to these parallel atrous convolutions, the ASPP module includes a global average pooling layer. This layer captures the global context of the input image by aggregating the entire feature map into a single value per channel. The output of this pooling layer is then upsampled to match the size of the input feature map and concatenated with the outputs from the atrous convolutions. The concatenation of these feature maps, which includes information captured at various scales, is performed along the channel dimension. This process effectively combines multi-scale information into a single, rich feature map. To fuse these multi-scale features and reduce the number of channels, a 1x1 convolution is applied to the concatenated feature map. This is followed by batch normalization and a ReLU activation function, which enhance training stability and performance. The final output from the ASPP module undergoes further processing by additional convolutional layers. It is then upsampled to the original image resolution, producing the final pixel-wise classification map.

## 5. Experiments

### 5.1. Data Preprocessing

In order to reduce computational load, all images and masks used in this project were resized from their original resolution of 2048x1024 pixels to 1024x512 pixels. This resizing step was essential to make the training process more efficient without compromising the integrity of the image details.

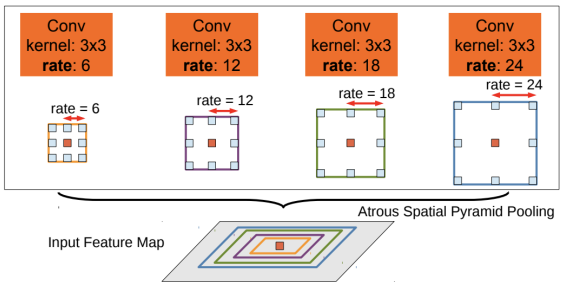


Figure 5. *Atrous Spatial Pyramid Pooling (ASPP). [5]*

Additionally, histogram equalization was applied to each image to enhance contrast and improve the overall quality of the input data. This technique adjusts the intensity distribution of the images, making features more distinguishable and thereby aiding the segmentation algorithm in identifying relevant patterns and structures.

Finally the images have been normalized by adjusting their pixel values to have a mean of zero and a standard deviation of one for each color channel. This preprocessing step helps to improve the convergence of the training process and enhances the performance of the neural network.


Figure 6. *Example of a leaf photo before (left) and after (right) the histogram equalization transformation.*

## 5.2. Data Augmentation

In this study, we experimented with various data augmentation techniques to enhance the robustness and generalization of our models. These techniques included random rotations, horizontal and vertical flips, scaling, and color jittering. Despite these efforts, the augmented data did not lead to any significant improvement in the models' performance. This suggests that the models were already effectively capturing the necessary features from the original dataset, or that the specific augmentations applied were not suitable for further enhancing the segmentation capabilities for this particular task.

## 5.3. Training

In order to compare U-Net and DeepLabV3 as fairly as possible, both underwent training with the same hyperparameters where applicable. These hyperparameters went through numerous iterations, updating them to achieve better metric performance until the final versions were reached, as shown in table 2.

| **Shared Parameters** | | |
|---|---|---|
| **Loss Function** | Cross Entropy | |
| **Optimizer** | ADAM | |
| **Learning Rate** | 1e-4 | |
| **# of Epochs** | 100 | |
| **Early Stopping Patience** | 5 ($\Delta$ = .001) | |
| **Individual Parameters** | U-Net | DeepLabV3 |
| **Batch Size Training** | 2 | 4 |
| **Batch Size Validation** | 2 | 4 |
| **Batch Size Test** | 1 | 2 |

Table 2. Overview of shared and unique hyperparameters for the best versions of the two model architectures.

To train the DeepLabV3 model, we employed a learning rate scheduler to optimize the training process. The scheduler was configured to monitor the validation loss and reduce the learning rate by a factor of 0.1 when no improvement in validation loss was observed for 3 consecutive epochs.

For the purpose of this project, a subsample of the coffee leaves dataset was utilized, in order to work within the constraints of our computational machinery. The models were trained on the same train/validation/test split of the 400 images selected from the larger base dataset: 70% train, 10% validation, 20% test. This training took place inside of jupyter notebooks running on Google Colab and utilized pytorch.

The final version of U-Net achieved the early stopping condition after only 10 epochs. DeepLabV3 achieved early stopping after 23 epochs. Figure 7 displays the loss values of each respective model across epochs.

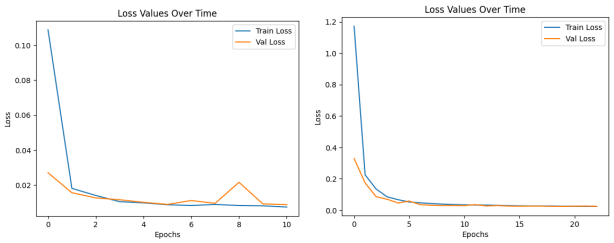


Figure 7. *Train and validation loss over time (epochs) for U-Net (Left) and DeepLabV3 (Right).*

## 5.4. Metrics

The metrics we have chosen to evaluate the models performances are:

- **Intersection over Union**
  Intersection over Union (IoU), also known as Jaccard index, measures the overlap between the predicted segmentation map and the ground truth.

$$\text{IoU} = \frac{|P \cap G|}{|P \cup G|}$$

- **Dice score**
  The dice score, like the IoU, measures the overlap between predictions and labels, but it is more sensitive to the size of the objects being segmented.

$$\text{Dice} = \frac{2|P \cap G|}{|P| + |G|}$$

- **Pixel accuracy**
  Pixel accuracy measures the proportion of correctly classified pixels out of the total number of pixels in the image.

$$\text{Pixel accuracy} = \frac{\sum_i 1(P_i = G_i)}{N}$$

The final metrics obtained for the two models are displayed in table 3. "Background" refers to pixel classification of the background of the image, "leaf" refers to pixels classified as healthy parts of the leaf, and "symptom" refers to pixels classified as symptomatic parts of the leaf.

| Metrics | U-Net | DeepLabV3 |
|---|---|---|
| IoU background | 0.9966 | 0.9906 |
| IoU leaf | 0.9845 | 0.9681 |
| IoU symptom | 0.7842 | 0.6713 |
| Dice score background | 0.9983 | 0.9953 |
| Dice score leaf | 0.9922 | 0.9837 |
| Dice score symptom | 0.8706 | 0.7974 |
| Pixel accuracy background | 0.9982 | 0.9971 |
| Pixel accuracy leaf | 0.9919 | 0.9803 |
| Pixel accuracy symptom | 0.8796 | 0.7941 |

Table 3. Performance of U-Net vs. DeepLabV3 across the chosen metrics.

U-Net outperformed DeepLabV3 across all 3 metrics on the test dataset. Figure 8 contains comparisons between predicted masks and the original mask for both U-Net and DeepLabV3.

## 6. Conclusion

In this project, we explored semantic segmentation of coffee leaves to detect biotic stress using two advanced deep learning models: U-Net and DeepLabV3 with ResNet50 as a backbone. Our experiments demonstrated that U-Net outperformed DeepLabV3 across all evaluated metrics.

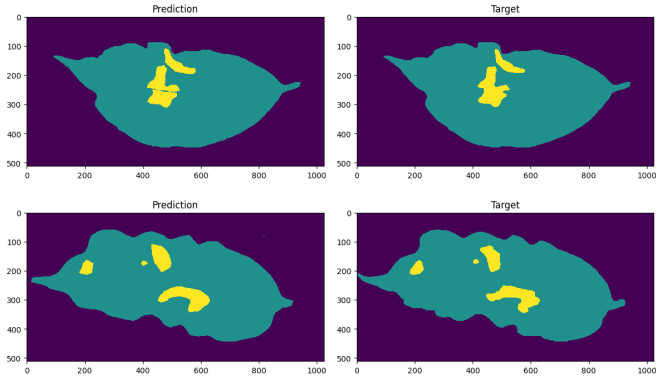


Figure 8. *Top: U-Net Predicted Mask vs Original Mask of a leaf. Bottom: DeepLabV3 Predicted Mask vs Original Mask of a leaf.*

Through this study, we have learned that model architecture plays a critical role in segmentation tasks, especially in domains requiring precise boundary delineation. The effectiveness of U-Net highlights the importance of encoder-decoder structures in capturing and reconstructing fine details from input images. Additionally, the ResNet50 backbone in DeepLabV3, while powerful for feature extraction, may not be as well-suited for the specific requirements of biotic stress segmentation in coffee leaves compared to U-Net's design.

For future extensions of this work, several avenues can be explored:

- **Data Augmentation and Preprocessing**: Investigating more sophisticated data augmentation techniques to enhance model generalization and robustness.
- **Advanced Architectures**: Exploring other state-of-the-art segmentation architectures, such as attention mechanisms or transformer-based models, which might offer improved performance.
- **Larger and more diverse dataset**: Expanding the dataset to include more images from different geographical regions and various stages of biotic stress. This will help improve the model's generalization and robustness across different conditions.

## References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurélien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels. *Technical report, EPFL*, 06 2010.

[2] Mohammad Shafiul Alam, Fatema Tuj Johora Faria, Mukaffi Bin Moin, Ahmed Al Wase, Md. Rabius Sani, and Khan Md Hasib. Potatogans: Utilizing generative adversarial

networks, instance segmentation, and explainable ai for enhanced potato disease identification and classification, 2024.

[3] Ridha Ilyas Bendjillali, Mohammed Beladgham, Khaled Merit, and Abdelmalik Taleb-Ahmed. Illumination-robust face recognition based on deep convolutional neural networks architectures, 2020.

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, 2017.

[6] J. G. M. Esgario, R. A. Krohling, and J. A. Ventura. Deep learning for classification and severity estimation of coffee leaf biotic stress, 2019.

[7] Hritwik Ghosh, Irfan Rahat, Kareemulla Shaik, Syed Khasim, and Manava Yesubabu. Potato leaf disease recognition and prediction using convolutional neural networks. *ICST Transactions on Scalable Information Systems*, 09 2023.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[9] Nour Eldeen Khalifa, Mohamed Taha, Lobna Abou El-Magd, and Aboul Ella Hassanien. Artificial intelligence in potato leaf disease classification: A deep learning approach, 01 2021.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[11] Zifei Luo, Wenzhu Yang, Yunfeng Yuan, Ruru Gou, and Xiaonan Li. Semantic segmentation of agricultural images: A survey. *Information Processing in agriculture*, 11:172–186, 2024.

[12] Giuliano L. Manso, Helder Knidel, Renato A. Krohling, and Jose A. Ventura. A smartphone application to detection and classification of coffee leaf miner and coffee leaf rust, 2019.

[13] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

[16] Utkarsh Yashwant Tambe, A. Shobanadevi, A. Shanthini, and Hsiu-Chun Hsu. Potato leaf disease classification using deep learning: A convolutional neural network approach, 2023.

[17] Li X., Zhou Y., Liu J., Wang L., Zhang J., and Fan X. The detection method of potato foliage diseases in complex background based on instance segmentation and semantic segmentation, 2022.