Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A new machine learning mango damage segmentation dataset and application models

Abubakhari Sserwadda

asserwadda@sun.ac.ug

Soroti University

Patricia Ankunda

Soroti University

Desire Kawala

Soroti University

Aisha Babirye

Soroti University

Research Article

Keywords: Mango segmentation, U-Net, Computer vision, Deep learning, Image processing, DeepLabV3, Masks

Posted Date: March 17th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8903916/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A new machine learning mango damage segmentation dataset and application models.

Abubakhari Sserwadda[1*], Patricia Ankunda[1], Desire Kawala[1], Aisha Babirye[1]

[1]Daimarilab School of Engineering and Technology, Soroti University, Soroti, Uganda.

*Corresponding author(s). E-mail(s): asserwadda@sun.ac.ug;
Contributing authors: patrishatrish01@gmail.com; desirekawala@gmail.com;
babiryeaisha5001@gmil.com;

Abstract

Mango fruit diseases significantly impact both market quality and yield. Early and accurate detection is of paramount importance for supporting farm and processing decisions. However, traditional manual inspection is typically slow, subjective, and error-prone. This research proposes a deep learning-based approach for segmenting diseased regions on mango fruits by comparing two popular semantic segmentation models, U-Net and DeepLabV3. A dataset of 1000 mango fruit images was acquired under natural lighting conditions, pixel-level, and augmentation was carried out which gave 4167 images, and after all the preprocessing steps, grayscale K-means masks were generated to provide ground truth. The two models were then trained and evaluated using a total of 500 images and 500 ground truth masks. Experimental results indicated that DeepLabV3+ showed better segmentation performances and lesion boundary estimation with stronger generalization compared to U-Net. These results confirm the ability of deep learning-based segmentation to provide reliable pixel-level lesion localization that can be used for automatic and objective assessment of mango disease for practical applications in agriculture.

Keywords: Mango segmentation, U-Net, Computer vision, Deep learning, Image processing, DeepLabV3, Masks

1 INTRODUCTION

Mango, also known as the "King of Fruits," is one of the most popular fruits consumed worldwide because of its taste and nutritional value [4]. Due to this demand, mangoes are cultivated in various regions around the world but are highly vulnerable to several diseases that can affect the quality and quantity of mango and the production of mango by-products [11]. Infections and diseases in mangoes are caused by microbes such as algae, parasites, bacteria, and fungi at every step of the plant's development from seedling through fruit consumption. The illness might appear as sooty mold, canker, spots, wilt, or stem haemorrhage, anthracnose, blotch, scab, necrosis, mildew, dieback, rot, and deformity [12].

As the worldwide demand for mangoes continues to grow, the detection of infections and damage levels in mango fruits is crucial to ensure productivity in orchards and processing factories [10]. Traditional methods are manual and very tiring, leading to inaccuracy because human judgment is prone to errors. This research focuses on creating a dataset for developing a machine

learning model capable of detecting healthy and infected mango regions with high precision.

2 LITERATURE REVIEW

Recent work demonstrates that segmentation-based methods have become ever more important in plant disease analysis because they provide lesion localization at the pixel level, as opposed to mere image-level detection, allowing disease severity and treatment to be quantified with better precision. In [7], the authors showed that lesion extraction from mango fruit allows automatic infected surface area measurement and enables reliable severity scoring under real farming conditions. However, this approach relies mostly on colour-based thresholding and does not perform any comparative study of the state-of-the-art deep learning-based segmentation models.

As observed in [14], the authors applied U-Net to fruit segmentation, demonstrating that encoder-decoder architectures can separate objects effectively under natural orchard settings with complex lighting and backgrounds. While it produced very strong segmentation results, the study focused on fruit extraction rather than the segmentation of disease lesions.

More recent studies on segmentation have emphasised multi-scale designs that enhance small or early-stage lesion detection. The research in [15] showed that the addition of a leaf detector to lesion segmentation enhances robustness to cluttered backgrounds, although with increased computational complexity. labelling

The availability of datasets continues to be a major hurdle. In [6], the authors presented a significant dataset of mango leaves, but most were labelled for classification tasks, not segmentation tasks, which reflects the general shortage of pixel-annotated images for mango diseases. This explains why semi-automatic labelling or limited manual automation is widely used by researchers in general.

Overall, though much has been accomplished in the recent literature, few have experimentally evaluated different segmentation architectures, such as U-Net and DeepLabv3 on mango datasets, which again may be considered a good justification for the current study. Below, we list the main contributions of this work.

- This work provides a detailed comparison between the U-Net and DeepLabv3 to analyze critically their ability to segment mango damage, demonstrating the trade-offs between fine boundary detection and multi-scale contextual understanding.
- Provides scalability to other fruits and crops since the methodology is generic and can be extended to other fruits and agricultural products with minimal modifications.
- This contributes a labelled dataset for mango images, which can serve as a valuable resource for future research in agricultural computer vision.

3 METHODOLOGY

3.1 Data collection

The dataset was collected using a method of image dataset acquisition from orchards, taken by Android phone cameras, captured under natural daylight at different orientations. Fruits were collected, focusing on capturing various kinds of damage and infections.

3.2 Data pre-processing

The images were pre-processed through different stages, first, by removing the background and resizing to a uniform size for all images and replacing them with a white background. This was followed by removing duplicate, corrupt, and blurry images. To increase the size of the dataset, image augmentation was carried out, where images were transformed in shape and size. These augmentation techniques were dynamically applied in the training process of the model to minimise overfitting, and they include HSV random transformation, i.e., hue, saturation, brightness, Image rotation and flipping [1].

We employed three augmentation techniques, which are HSV random transformation, image rotation, and flipping, as shown in Fig. 1 after all techniques have been applied on one mango image. Through this, we were able to increase our dataset from 1000 to 4000 so as minimise the chances of overfitting



Fig. 1: Same image after deploying augmentation techniques

3.3 Creating segmentation masks

We developed masks using unsupervised models following the steps below.

3.3.1 Choosing best light segmentation

We implemented two segmentation styles from which we choose the one best for our task, that is, a threshold-based approach and k-means (k=2). HSV and gray clusters were implemented altogether in order to compare performance using segmented damage percentage, overlay histograms.

HSV clustering isolates dark and discolored infected areas effectively, while grayscale clustering provides stable but less colour-sensitive masks as shown in Fig.2.

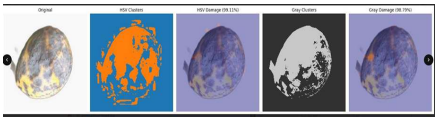


Fig. 2: HSV and Grayscale comparison

From the statistics in Table 1, below, it is evident that the damage detection was too high. The HSV approach yields lower average damage and higher variance than greyscale; hence, it is most sensitive to true colour differences.

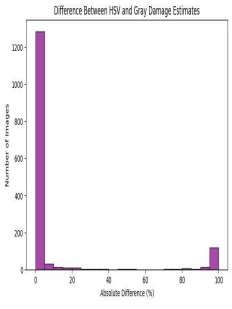


Fig. 3: Difference distribution graph

The average correlation between HSV and gray is 0.532, indicating a moderate linear relationship.

Table 1: Damage detection statistics of HSV and Gray scale

	HSV Damage (%)	Gray Scale Damage (%)
count	1500.000	1500.000
mean	89.148	98.959
std	28.947	5.841
min	0.000	0.450
25%	98.625	99.980
50%	99.950	100.000
75%	100.000	100.000
max	100.000	100.000

The evaluation was carried out using silhouette analysis, first on one image, which gave the best k as 4 with a silhouette of 0.738. This was done on 8 samples and gave the best silhouette score 0.658. The red line on the plot of Fig. 4 represents the

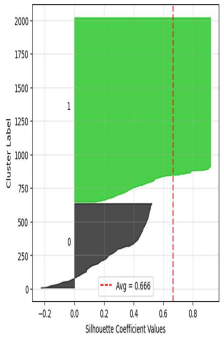


Fig. 4: Sample Silhouette plot for sample image

average silhouette for the particular clustering.

Evaluation using three metrics was done on a sample image, and the mean results are presented in Table 2.

Table 2: Comparison of mean results of three metrics on one sample image

Evaluation metric	Values
Silhouette Score	0.654
calinski-Harabasz Index	60530.597
Davies-Bouldin Index	0.790

The analysis on percentage damage detection by both the clustering methods showed that greyscale has better mask generations compared to HSV clustering, as shown in Table. 1

3.3.2 Creating Grayscale masks

The ground truth masks were developed using K-means grayscale clustering. This uses k=4 and clusters pixels into 4 intensity groups. IT picks the darkest cluster, thus creating a binary mask of 255 and 0 for the damaged and healthy parts, respectively, as shown in Fig. 5.



Fig. 5: Grayscale segmentation sample image

3.4 Data description

In this project, we created a dataset of 500 damaged/diseased mangoes. These images were collected by taking images at different orientations using camera phones.

3.5 Model Training

We trained two popular segmentation models including the U-Net [5] and the DeepLabV3 [8][9].

U-Net was selected to be used because of its ability to generate precise pixel-level segmentations from relatively small datasets. The U-Net's symmetric encoder and decoder architecture, coupled with a deep learning based skip connections model, allows it to simultaneously capture global context and fine-grained details [5].

The DeepLabV3, which was developed in 2017, was chosen as a model to use because of the



Fig. 6: Sample image of a damaged mango

improved ASSP by adding batch normalization and removing CRF. DeepLabV3 has modifications to handle multi-scale object segmentation, utilizes atrous convolution modules, in cascade or parallel, to utilize multi-scale context [8][9]

3.5.1 U-NET model

The mango images were loaded in 500 numbers, plus the corresponding 500 masks from the drive.

Preprocessing steps

1. Images and masks were resized to a uniform dimension of 224x224 pixels.
2. Images were converted from BGR to RGB format.
3. pixel values for both images and masks were normalized to the range [0,1] by dividing by 255.
4. Masks, initially loaded as grayscale, had an additional channel dimension added to maintain consistency with image dimensions.

Dataset conversion and splitting

The processed images and masks were converted into Numpy arrays and were randomly split into training and validation sets. 80% of the data (400 samples) was used for training and 20% (100 samples) for validation.

U-Net Model Implementation

The model implemented for this segmentation task takes an input image with a shape of 224, 224, 3, meaning images of 224x224 pixels with 3 RGB channels. The classic U-net design, consisting of a symmetric encoder-decoder with skip connections,

was employed. The architecture is symmetric and has three key parts:

1. Contracting Path (Encoder)
   Uses small filters to scan the image and find features. Each block typically involves two conv2D layers and relu activation and batch normalization, which help the model learn better [2].
   Uses max pooling to shrink the image size while keeping important images, which helps the network focus on bigger features [2].
2. Bottleneck: The bottleneck block follows the encoder, with two conv2D layers and 1024 filters, processing the most abstract features [2].
3. Decoder (Upsampling path): the path reconstructs the spatial information. It also comprises four blocks, each with an 'UpSampling2D' layer to increase spatial dimensions, followed by a 'Concatenate' layer that merges the upsampled features with their corresponding skip connections. This concatenation provides crucial fine-grained information lost during downsampling. The combined features then pass through two 'Conv2D' layers with 'relu' activation and 'BatchNormalization [2].
4. The final layer is a 'Conv2D' layer with a single filter and a 'sigmoid' activation, producing a binary mask (probability map) of the segmented object with an output shape of '(224, 224, 1)'.

Model training and compilation specifications

The Adam optimizer was used with a learning rate of 0.0001, and binary-cross entropy was employed as the loss function, suitable for image segmentation tasks [3].

Training was performed with a batch size of 8 and 5 epochs as part of the hyperparameters, as shown in Table 1. The validation set was used to monitor performance during training.

Table 3: U-Net model parameters

Parameters	Values
Learning rate	0.0001
Batch size	8
Epochs	5

Early stopping was used to monitor validation loss and patience of the training model. The restored best weights were set to ensure model weights from the epoch with restored best validation were used.

Model checkpoint was set to save the best model weights.

3.5.2 DeepLabV3 Model

This segmentation model performs semantic segmentation on the images and corresponding masks utilizing a ResNet50 backbone in its architecture [9].

Below is the explanation of its key features. [13]

1. Semantic segmentation: The model's primary task is to assign a class label to each pixel in an image.
2. Atrous convolution: This is the core technology that allows the model to control the receptive field without increasing the number of parameters or losing spatial resolution.
3. Multi-scale context: By applying atrous convolution with different rates in parallel, the model can capture information at multiple scales.
4. Backbone architectures: DeepLabV3 can be built on various backbone networks, such as ResNet, Xception, and MobileNet.

Training Process

The model was trained on the dataset of 500 image-mask pairs and split into 400 samples for training and a validation set of 100 samples.

The images and masks were resized to 256x256 pixels, normalized to a range [0,1].

The model training used alearning rate of 0.001, a batch size of 4, 5 epochs, a cross-entropy loss function, and Adam optimizer, which combines the advantages of momentum and RMSprop techniques to adjust learning rates during training. [3].

Table 4: DeepLabV3 model parameters

Parameters	Values
Learning rate	0.001
Batch size	4
Epochs	5

4 RESULTS

Table 5: Metrics results for both models

Metric	Values (U-Net)	AVG (DeepLabV3)
Precision	0.9575	**0.7543**
Recall	0.8059	**0.9086**
Val binary crossentropy	0.3655	**0.1022**
Val Dice loss	0.2195	**0.1778**
Val Dice coefficient	0.7805	**0.8222**
IoU	0.7781	**0.7008**
F1-Score	0.8752	**0.8222**

4.1 U-Net Results

4.1.1 Segmentation visualisation

The visualisation of the U-NET performance in Fig. 7 showing the original, mask, and predicted mask shows that the model achieved quite a good accuracy and the masks have been well predicted.

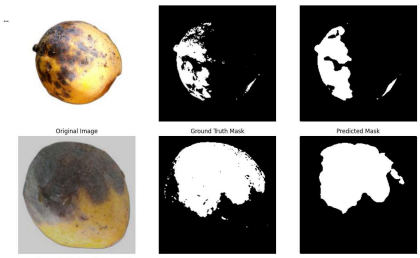


Fig. 7: Segmentation visualisation for U-Net model

4.1.2 Analysis

1. The training overall loss consistently decreased from 0.3928 in epoch 1 to 0.1891 in epoch 5, showing that the model was effectively learning from the training data.
2. The validation overall loss initially decreased from 0.8197 to 0.3304 from epoch 1 to epoch 2; it then fluctuated, showing a slight increase in epochs 3 and 4 before decreasing again to 0.2186 in epoch 5.
3. The training Dice loss showed a steady and consistent decrease from 0.3928 to 0.1891 across epochs, showing continuous improvement in segmentation quality.
4. The validation Dice loss showed a sharp drop from 0.8218 to 0.3328 by epoch 2, then increasing in epochs 3, and finally decreasing in epoch 5.The final validation dice loss is quite close to the training dice loss, which is a good thing.
5. The validation IoU started low at 0.0978 in epoch 1, improved significantly to 0.5006 in epoch 2, then decreased slightly in epoch 3 and 4, then finally increased to 0.6391 in epoch 5. This aligns with the validation loss, thus showing the model's ability to overlap with ground truth masks on unseen data improved considerably.
6. Training BCE (Binary-crossentropy) decreased steadily from 0.8315 to 0.4980 across the epochs.
7. Validation BCE decreased from 0.4013 to 0.2439 by epoch 2, then increased to 0.4347 in epoch 4, then dropped to 0.3374 in epoch 5.
8. There is good convergence with the model, and the validation dice coefficient, training dice coefficient being close, plus the validation loss and training loss being close to each other, shows that there is no significant overfitting on the data.

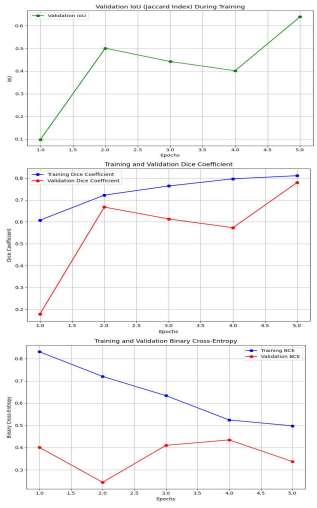


Fig. 8: Training and validation Graphs

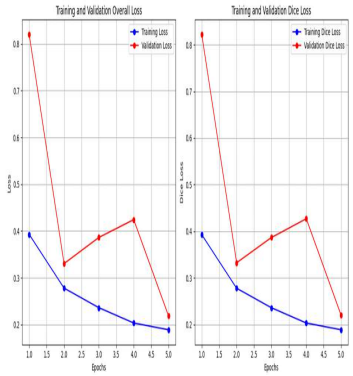


Fig. 9: Loss curves for U-net model

Image

4.2 DeepLabV3 Results

4.2.1 Segmentation Visualisation.

The visualisation of the DeepLabV3 performance in Fig. 10 showing the original, mask, and predicted mask shows that the model achieved quite a good accuracy and the masks have been well predicted.

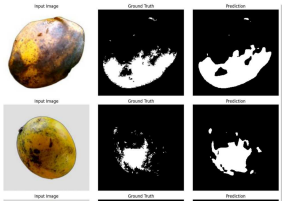


Fig. 10: Segmentation visualisation of DeepLabV3

4.2.2 Analysis

1. The model achieved its best validation IoU of 0.7008 at Epoch 5.
2. Evaluation metrics indicate strong performance, that is to say, Average IoU of 0.7008, Average Dice Coefficient of 0.8222, Average F1 Score of 0.8222, Average Precision of 0.7543, Average Recall of 0.9086, and an excellent AUC of 0.9886 indicate strong performance of the model.
3. cross-entropy loss revealed a mean loss of 0.1022 but a maximum loss of 6.8985 and a standard deviation of 0.2967, indicating the presence of "hard examples."
4. Qualitative performance, as observed through sample prediction visualizations, showed good alignment between predicted and ground truth masks, confirming the model's effective segmentation ability.
5. The AUC(Area Under the Curve) of 0.9886 signifies the model's strong ability to distinguish between foreground and background pixels across various thresholds.

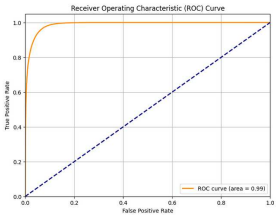


Fig. 11: Receiver Operating Characteristic (ROC) curve for DeepLabV3

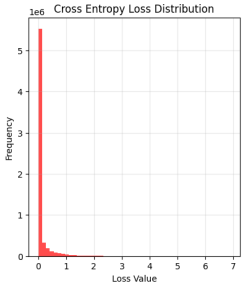


Fig. 12: Cross Entopy Loss distribution for DeepLabV3 model training

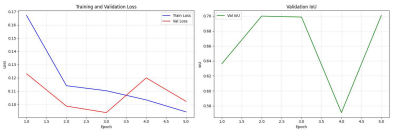


Fig. 13: Loss curves for DeepLabV3

4.2.3 Models' compulsion

The DeepLabV3 is observed to consistently outperform the U-net for both the dice coefficient and IoU segmentation accuracy; this is due to its stronger generalisation ability, evidenced by closer variation between its training and validation loss curves in Fig. 15. In comparison, the U-Net maintained a relatively significant performance gap between training and validation scores; the validation curve has more fluctuations.

4.3 Limitations

Limited RAM led to the crushing of sessions during training leading to the use of a small dataset and limited generalization of the models.

Variations in lighting and background where photographs taken in natural daylight resulted in inconsistent conditions that affected model training.

Few training epochs due to limited RAM. Both U-NET and DeepLabV3 were trained for a few epochs, reducing performance potential.

5 CONCLUSION

This study developed a deep learning based system for automatic detection and localization of disease damage on mango fruits by semantic segmentation. U-Net and DeepLabV3, after training on masks generated through greyscale clustering, could provide accurate pixel-level segmentation of the infected regions. DeepLabV3 performed better than the other model, with higher IoU, Dice score, and precision, while performing better in the qualitative lesion boundary extraction. This was improved by its atrous spatial pyramid pooling, which provided stronger multi-scale feature learning, along with better handling of complex visual patterns that characterise real-world mango images. In this way, the system offers a fast, objective, and reproducible alternative to manual inspection to support improved disease monitoring, grading, and decision-making in mango production and post-harvest processing. The present study lays the foundation for practical AI-based assessment tools in fruit agriculture and underlines the possibility of further extension of deep learning segmentation to other crops and types of diseases in the near future. For future work, we shall aim at increasing the dataset size and diversity to improve model robustness and do more sub-classifications, including mild and severe damages for grading efficiency, and deploy the model to mobile platforms for real-time mango damage detection.

Declarations
Consent to Participate declaration: not applicable.
Consent to Publish declaration: not applicable.
Ethics declaration: not applicable.
Funding: not applicable.
For transparency and reproducibility of the results, our code is accessible at https://github.com/Patricia256-ug/MANGO-DAMAGE-SEGMENTATION.
Our new dataset is publicly accessible at https://data.mendeley.com/datasets/cy3ccgkx3n/1

References

[1] (2025) Identifying cocoa flower visitors: A deep learning dataset. URL https://www.nature.com/articles/s41597-025-05631-3

[2] (2025) U-net architecture explained. URL https://www.geeksforgeeks.org/machine-learning/u-net-architecture-explained/

[3] (2025) What is adam optimizer? URL https://www.geeksforgeeks.org/deep-learning/adam-optimizer/

[4] A.A.Salamai (2023) Enhancing mango disease diagnosis through eco-informatics: A deep learning approach. Ecological Informatics 77

[5] Agarwal S, Eapen S, Upadhyay M, et al (2023) What makes u-net a winner in biomedical image. URL https://www.excelra.com/blogs/u-net-biomedical-image-segmentation/

[6] Ahmed SI, Ibrahim M, Nadim M, et al (2022) Mangoleafbd: A comprehensive image dataset to classify diseased and healthy mango leaves URL arxiv.org/abs/2209.02377

[7] Faye D, Diop I, Mbaye N, et al (2025) Mango fruit diseases severity estimation

based on image segmentation and deep learning. Springer Nature 7

[8] fekadu T (2024) Deeplabv3. URL https://medium.com/@elalan_fekadu/deeplabv3-3d4679becf93

[9] Kouidri A (2023) Understanding deeplabv3 in image segmentation. URL https://www.ikomia.ai/blog/understanding-deeplabv3-image-segmentation

[10] M D, C VK, M V, et al (2024) Detection and classification of disease from mango fruit using convolutional recurrent neural network with metaheuristic optimizer. International Research Journal of Education and Technology 06

[11] Panuganti K (2024) Classification of various diseases in the mango crop using machine learning. Master's thesis, School of Computing National College of Ireland

[12] Singh J, Jadhav-Mane S (2024) Mango skin disease detection techniques based on machine learning techniques: A review. Springer nature 139

[13] Singh V (2022) Deeplabv3 deeplabv3+ the ultimate pytorch guide. URL https://www.ikomia.ai/blog/understanding-deeplabv3-image-segmentation

[14] Swaroop DV, Mahesh D, Dheeraj GNSSKP (2022) Fruit detection and localization using unet. nternational Journal for Research in Applied Science Engineering Technology (IJRASET) 10

[15] Yang T, Zhou S, Xu A, et al (2023) An approach for plant leaf image segmentation based on yolov8 and the improved deeplabv3+. URL https://www.mdpi.com/2223-7747/12/19/3438