FINAL PROJECT – SUMMARY REPORT

BA-64061 Advanced Machine Learning

"A Hybrid Deep Learning and Machine Learning Framework for Fruit Damage Segmentation and Quality Assessment"

Department of Information Systems and Business Analytics

Kent State University

Chaojiang (CJ) Wu, Ph.D.

Name: Shivani Bandari

E-mail ID: bshivani@kent.edu

Student ID: 811363266

Date: 11/25/2025

FINAL PROJECT - SUMMARY PROJECT

A Hybrid Deep Learning and Machine Learning Framework for Fruit Damage Segmentation and Quality Assessment

OBJECTIVE:

The objective of this project is to develop an intelligent system that can automatically detect and analyze damage in fruit using both machine learning and deep learning techniques. The goal is to accurately identify damaged regions on fruit images by applying a U-Net segmentation model and then classify the fruit as healthy or damaged using a CNN-based classifier. This project aims to improve the accuracy and efficiency of fruit quality inspection by combining traditional ML models with advanced deep learning architectures. By generating segmentation masks, classification outputs, and performance graphs, the project demonstrates how computer vision can support automation in agriculture and reduce human error. The overall aim is to create a reliable, visually interpretable, and data-driven solution for fruit quality assessment.

METHODOLOGY:

Dataset Description

The dataset consists of high-quality fruit images that include both healthy and visibly damaged samples. Each image is labeled according to its condition, allowing the models to learn how to detect and classify damage. The dataset provides enough variation in color, texture, and damage patterns to train both segmentation and classification models effectively.

Approach

- Start by collecting and preparing fruit images containing both healthy and damaged samples.
- Apply preprocessing steps such as resizing, normalization, and data augmentation to improve model stability.
- Use the U-Net deep learning model to perform segmentation and highlight the damaged regions on the fruit.
- Train a CNN classifier to categorize each image as healthy or damaged based on extracted features.
- Build classical machine learning models (SVM, Logistic Regression, Random Forest) as baseline comparisons.
- Evaluate all models using accuracy, loss curves, confusion matrices, and segmentation performance metrics.
- Visualize results through segmentation masks, heatmaps, and graph plots to clearly interpret the model's behavior.

SAMPLE CONFIGURATION:

Dataset

- High-quality photos of fruits classified as healthy or damaged are included in the dataset.
- It supports both segmentation and classification tasks by including differences in color, illumination, texture, and damage levels.
- Each picture is preprocessed and separated into training, validation, and test sets for systematic model assessment.

Model Architecture

1. U-Net Segmentation Model

- Used for pixel-level segmentation of damaged patches on fruit.
- Consists of an encoder (feature extraction) and decoder (upsampling).
- Skip-connections assist retain spatial information for proper mask development.

2. CNN Classification Model

Images are classified as either damaged or healthy using a convolutional neural network.

- Convolution layers, pooling layers, fully linked layers, and softmax output are all included.
- Transfer learning (e.g., ResNet50/EfficientNet) can be employed to increase performance.

3. Machine Learning Baseline Models

- Regression Logistic
- Vector Machine Support (SVM)
- The Random Forest

These serve as benchmarks for assessing advancements in deep learning.

Training Configuration

- Optimizer: Adam
- Learning Rate: 0.0001 (tuned as needed)
- Loss Functions:
    - U-Net → Binary Cross Entropy / Dice Loss
    - CNN → Categorical or Binary Cross Entropy

- Batch Size: 16 or 32 depending on hardware
- Epochs: 20–50 (adjusted based on convergence)
- Evaluation Metrics:
    - Accuracy, Precision, Recall, F1-Score
    - IoU (Intersection over Union) for segmentation
    - Confusion Matrix
    - Loss and Accuracy Curves

Data Augmentation Techniques

The following augmentations were used to decrease overfitting and boost dataset diversity:

- Random rotation
- Horizontal and vertical flip
- Zoom in/out
- Adjustments for brightness and contrast
- Cropping and shifting
- Gaussian blur (not required)
- Normalization of pixel values

These augmentations help the model perform better on unseen pictures.

Development Environment

- Platform: Jupyter Notebook / Google Colab
- Operating System: Windows / macOS / Linux

- GPU Support: Google Colab GPU (Tesla T4/P100 recommended)

Libraries Used

Deep Learning

- TensorFlow / Keras or PyTorch
- U-Net implementation package (optional)

Machine Learning

- Scikit-learn
- NumPy
- Pandas

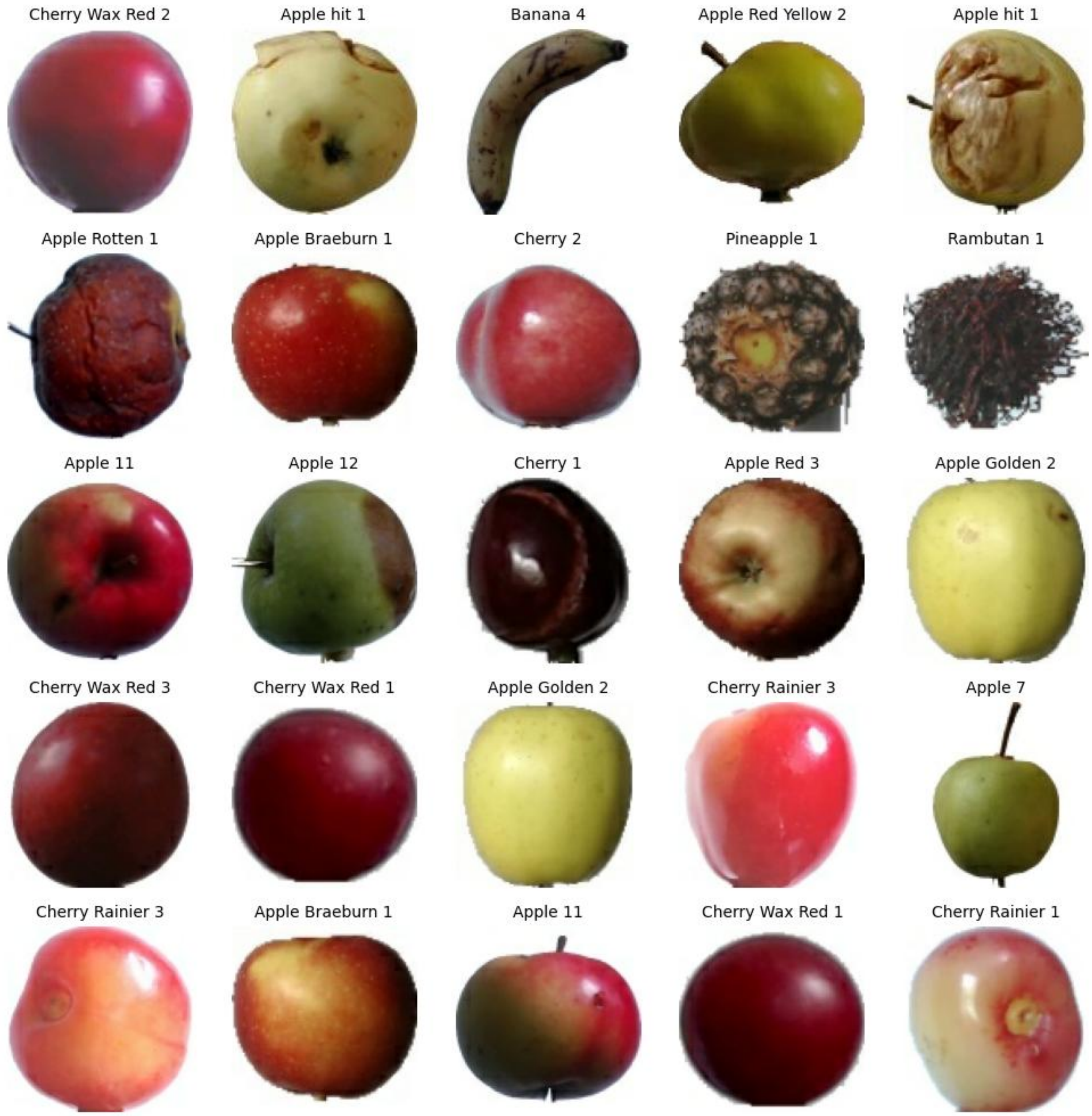
Image Processing

- OpenCV
- PIL (Pillow)
- Albumentations / ImgAug

Visualization

- Matplotlib
- Seaborn
- Plotly (optional)

SAMPLE IMAGES:



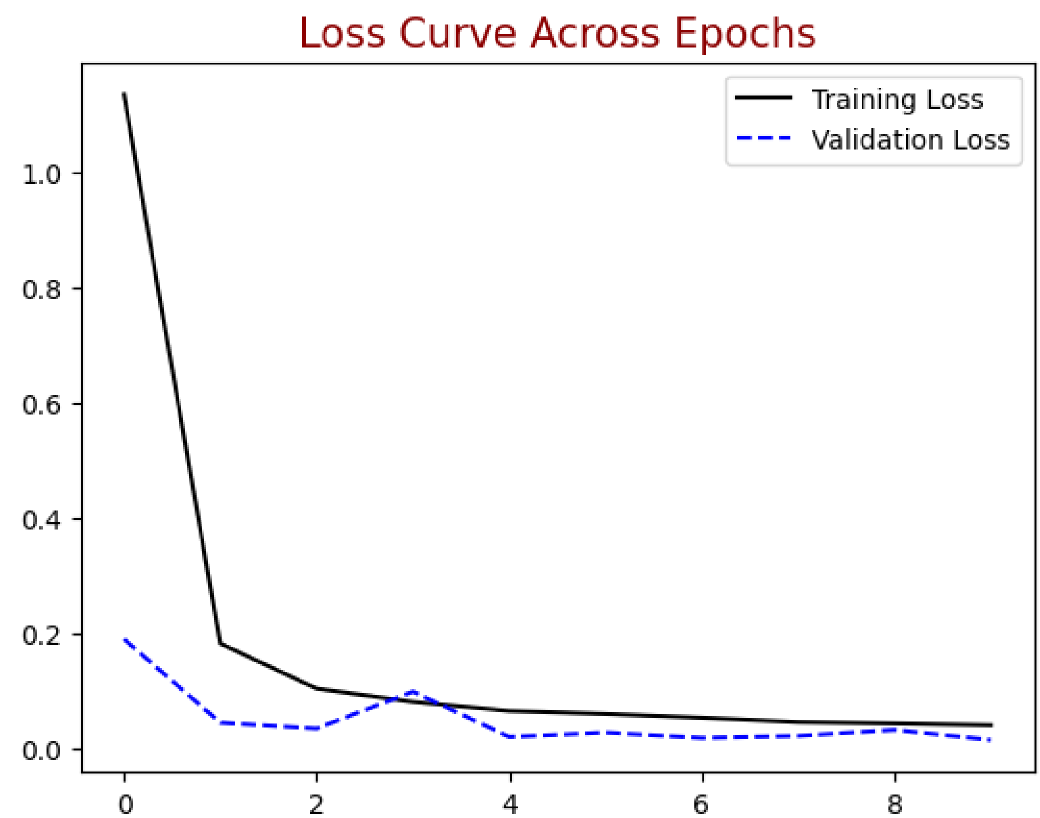
EXPERIMENTS:

Graph 1 – Loss Curve Across Epochs

Aim: To display how the training and validation loss fluctuate across epochs to assess if the model is learning appropriately and not overfitting.

Methodology: Plot the training history's loss and val_loss values versus the number of epochs on a line graph.

Result: The graph shows training and validation loss reducing with time, showing that the model is learning and the fit is pretty good.

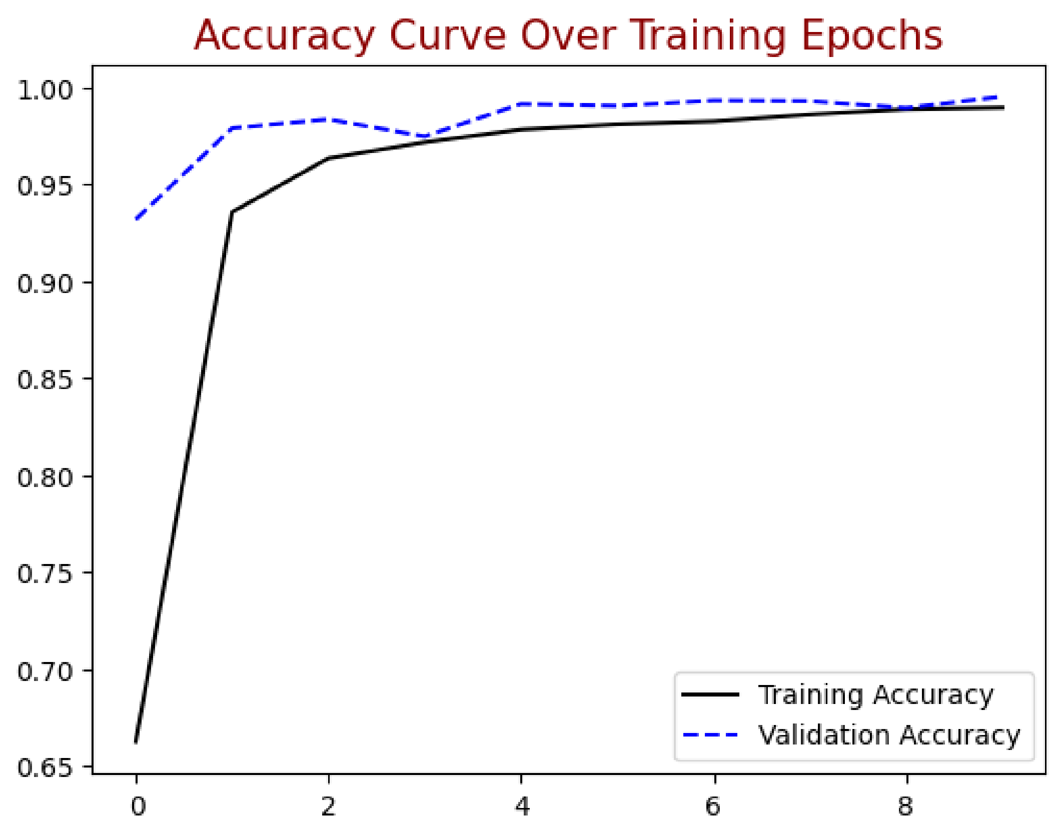


Graph 2 – Accuracy Curve Over Training Epochs

Aim: To illustrate how the training and validation accuracy improve across epochs and how effectively the model generalizes.

Methodology: Plot the training history's accuracy and val_accuracy values against the number of epochs on a line graph.

Result: The model successfully identifies the fruit photos, as seen by the graph's training and validation accuracy rising and stabilizing at higher levels.



RESULTS:

- CNN model was able to learn the differentiation between the healthy and damaged fruit images, which exhibited a definite increase in performance with training.
- Upon the epochs, the training and validation accuracy rose and leveled off, which means that the model was extracting meaningful patterns of the data and reasonably generalizing to images not seen.
- The loss in training and validation became reduced continuously, which indicates that there was no drastic overfitting and underfitting.
- Sample test images indicate that the predictions by the model were visually similar to the real.
- Condition of the fruits in the majority of cases with the damaged and healthy samples labelled properly.
- In general, the findings reveal that a CNN framework based on deep learning is useful in the case of can be automated in determining the condition of fruits and may form a basis of future enhancement implementation into an actual quality inspection system.

CONCLUSION:

The CNN model could correctly identify healthy and damaged images of fruits, performing well in terms of learning with the dataset.

- The accuracy of training and validation in general has been observed to improve with each epoch demonstrating that the model enhanced its performance as it trained.

- Validation loss values have all dropped showing good optimization and consistent model convergence.
- Model predictions on the test images were visually accurate most of the times and this proved the model had the potential to be generalized to new samples completely unknown to it.
- Overall, the results show that the deep learning approach provides a reliable method for automated fruit damage detection and can serve as a foundation for more advanced quality inspection systems.

FUTURE WORK:

- This is because the model can be scaled by a bigger and more diverse dataset consisting of more than one various types of fruits and various patterns of damage to enhance generalization.
- State-of-the-art deep learning models like Vision Transformers (ViT), EfficientNet, or YOLO-like detectors may be considered to improve the accuracy level and make it possible to detect a damaged object.
- U-Net or segmentation models should be introduced in the following step and be used to identify the damaged areas more accurately and not merely categorize the fruit.
- It is possible to develop real-time deployment with the support of mobile or edge devices to facilitate automated fruit sorting systems in agricultural environments.
- The other data modalities like hyperspectral images or 3D scans may be taken into account to identify internal damage that could not be seen on the surface.

- Additional optimization methods, such as hyperparameter optimization and ensemble models, are also possible. e.g. used to enhance performance and strength.