UNIVERSITY OF WESTMINSTER

Sensors and Signals

5ELEN021C

Group Coursework – Report

Module Leader : **Ms. Saumya Reni**

Assessment : **Coursework**

Due Date : **Thursday 9th January 2025 at 13:00:00**

Group Number : **03**

Group Members :

Name	IIT Number	UOW Number
E.M.A.I Ekanayake	20230061	w2051550
Uditha Vithanage	20230220	w2083051
Nisal Rukshan	20230206	w2083177

I. Acknowledgement

We would like to express our sincere gratitude to Ms. Hasini Kasthuriarachchi, our esteemed lecture, for her invaluable guidance, encouragement, and support throughout the completion of this project. Her expertise and insightful feedback have been instrumental in shaping the direction and depth of our work. Finally, we also express our appreciation to all our peers, family, and institutions, whose encouragement and collaboration have been a constant source of motivation, enabling us to keep moving forward with our project.

II. Table of Content

III. List Of Figures

1 Part - 1

1.1 Matlab Code

Main GUI function

```
%% Main GUI function
function plantDetectGUI

    % Clear environment and close all figures
    clc;
    clear all;
    close all;

    % Create the main figure window
    fig = figure('Name', 'Plant Detection','NumberTitle','off', 'Position',
[200 50 1200 700]);

    % Create UI controls - buttons
    uicontrol('Style', 'pushbutton', 'String', 'Load Images', ...
        'Position', [40 560 100 30], 'Callback', @loadImages);

    uicontrol('Style', 'pushbutton', 'String', 'Calculate Contrast', ...
        'Position', [40 520 100 30], 'Callback', @calculateContrast);

    uicontrol('Style', 'pushbutton', 'String', 'Convert to Grayscale', ...
        'Position', [40 480 100 30], 'Callback', @convertGrayscale);

    uicontrol('Style', 'pushbutton', 'String', 'Show Histogram', ...
        'Position', [40 440 100 30], 'Callback', @showHistogram);

    uicontrol('Style', 'pushbutton', 'String', 'Segmentation', ...
        'Position', [40 400 100 30], 'Callback', @segmentedImage);


    % Create static text for RGB weights
    uicontrol('Style', 'text', 'String', 'RGB Weights:', ...
        'Position', [40 360 100 20], 'HorizontalAlignment', 'left');
    uicontrol('Style', 'text', 'String', 'R: 0.1140', ...
        'Position', [40 340 100 20], 'HorizontalAlignment', 'left');
    uicontrol('Style', 'text', 'String', 'G: 0.5870', ...
        'Position', [40 320 100 20], 'HorizontalAlignment', 'left');
    uicontrol('Style', 'text', 'String', 'B: 0.2989', ...
        'Position', [40 300 100 20], 'HorizontalAlignment', 'left');

    % Create static text for contrast values
```

```
    global contrastText;
    contrastText = cell(1,3);

    % Contrast value display area
    uicontrol('Style', 'text', 'String', 'Contrast Values:', ...
        'Position', [40 260 100 20], 'HorizontalAlignment', 'left');

    for i = 1:3
        contrastText{i} = uicontrol('Style', 'text', ...
            'String', sprintf('Image %d: --', i), ...
            'Position', [40 240-((i-1)*20) 100 20], ...
            'HorizontalAlignment', 'left');
    end
end
```

Load and display images function

```
%% Load and display images function
function loadImages(~, ~)
    global originalImages;

    originalImages = cell(1,3);

    % Load up to 3 images with file dialog
    for i = 1:3
        [filename, pathname] = uigetfile('*.jpg;*.png;*.jpeg;*.jfif',
strcat('Select Image ',num2str(i)));
        if filename ~= 0
            img = imread(fullfile(pathname, filename));
            originalImages{i} = img;

            subplot(2,3,i);
            imshow(img);
            title(['Original Image ' num2str(i)]);
        end
    end
end
```

Custom function for RMS contrast calculation

```
%% Custom function for RMS contrast calculation
function contrast = calculateRMSContrast(img)
    % Convert to grayscale if RGB
    if size(img,3) == 3
        img = rgb2gray_custom(img);
    end

    % Calculate RMS contrast
    img = double(img)/255;
    mean_intensity = mean(img(:));
    contrast = sqrt(mean((img(:) - mean_intensity).^2));
end
```

Calculate and display contrast for all images

```
%% Calculate and display contrast for all images
function calculateContrast(~, ~)
    global originalImages contrastText;

    % Check if any images are loaded
    if isempty(originalImages) || all(cellfun(@isempty, originalImages))
        msgbox('Please upload at least one image.');
        return;
    end

    % Calculate and display contrast for each image
    for i = 1:3
        if ~isempty(originalImages{i})
            contrast = calculateRMSContrast(originalImages{i});
            % Update contrast value in UI
            set(contrastText{i}, 'String', sprintf('Image %d: %.4f', i,
contrast));
        end
    end
end
```

Custom RGB to Grayscale conversion

```
%% Custom RGB to Grayscale conversion
function gray_img = rgb2gray_custom(rgb_img)
    % Optimized weights for better contrast
    r_weight = 0.2989;
    g_weight = 0.0870;
    b_weight = 0.6140;

    % Apply weighted conversion
    gray_img = r_weight * double(rgb_img(:,:,1)) + ...
               g_weight * double(rgb_img(:,:,2)) + ...
               b_weight * double(rgb_img(:,:,3));
    gray_img = uint8(gray_img);
end
```

Convert to grayscale with optimized weights and display

```
%% Convert to grayscale with optimized weights and display
function convertGrayscale(~, ~)
    global originalImages;

    % Check if any images are loaded
    if isempty(originalImages) || all(cellfun(@isempty, originalImages))
        msgbox('Please upload at least one image.');
        return;
    end

    % Convert and display each image
    for i = 1:3
        if ~isempty(originalImages{i})
            gray_img = rgb2gray_custom(originalImages{i});
            subplot(2,3,i+3);
            imshow(gray_img);
            title(['Grayscale Image ' num2str(i)]);
        end
    end
end
```

Custom histogram calculation

```
%% Custom histogram calculation
function hist_values = calculateHistogram(img)

    % Convert image to grayscale
    if size(img,3) == 3
        img = rgb2gray_custom(img);
    end

    % Calculate histogram values
    hist_values = zeros(1,256);
    for i = 0:255
        hist_values(i+1) = sum(img(:) == i);
    end
end
```

Display histogram

```
%% Display histogram
function showHistogram(~, ~)
    global originalImages;

    % Check if any images are loaded
    if isempty(originalImages) || all(cellfun(@isempty, originalImages))
        msgbox('Please upload at least one image.');
        return;
    end

    % Create new figure for histograms
    figure('Name', 'Image Histograms','NumberTitle','off','Position', [300 100
1000 500]);

    % Display histogram for each image
    for i = 1:3
        if ~isempty(originalImages{i})
            subplot(1,3,i);
            hist_values = calculateHistogram(originalImages{i});
            bar(0:255, hist_values);
            title(['Histogram ' num2str(i)]);
            xlabel('Pixel Intensity');
            ylabel('Frequency');
        end
    end
end
```

Leaf detection using image processing

```
%% Leaf detection using image processing
function segmentedImage(~, ~)
    global originalImages;

    % Check if any images are loaded
    if isempty(originalImages) || all(cellfun(@isempty, originalImages))
        msgbox('Please upload at least one image.');
        return;
    end

    % Create a figure for display segmentation results
    figure('Name', 'Leaf Detection','NumberTitle','off','Position', [300 100
1000 500]);

    % Process each image
    for i = 1:3
        if ~isempty(originalImages{i})
            % Extract RGB channels
            image = originalImages{i};
            greenChannel = image(:,:,2); % Extract the green channel
            redChannel = image(:,:,1);  % Extract the red channel
            blueChannel = image(:,:,3); % Extract the blue channel

            % Create Green mask with enhased contrast
            greenMask = (greenChannel > redChannel * 1.001) & (greenChannel >
blueChannel * 1.001);
            enhancedGreen = imadjust(greenChannel,
stretchlim(greenChannel(greenMask), [0.02, 0.98]), [0 1]);

            % Appy otsu thresholding
            thresholdValue = graythresh(enhancedGreen); % Otsu's thresholding
            binaryMask = imbinarize(enhancedGreen, thresholdValue * 0.65);

            % Remove Noise and Small objects
            minArea = round(size(greenChannel, 1) * size(greenChannel, 2) *
0.00005);
            noiseFreeMask = bwareaopen(binaryMask & greenMask, minArea);

            % Apply Morphological Processing with Structuring Element
            se = strel('disk', 18);
            refinedMask = imclose(noiseFreeMask, se); % Close small gaps
            refinedMask = imfill(refinedMask, 'holes'); % Fill holes within
the leaf

            % Extract the largest leaf reigon
```

```
            largestLeaf = bwareafilt(refinedMask, 1); % Retain only the
largest region

            % Overlay Segmentation on Original Image
            overlayImage = imoverlay(image, largestLeaf, [1, 0, 0]); %
Highlight the leaf in red

            % Display Results
            subplot(3, 3, (i-1)*3 + 1); imshow(image); title(['Original Image
' num2str(i)]);
            subplot(3, 3, (i-1)*3 + 2); imshow(largestLeaf); title(['Segmented
Leaf ' num2str(i)]);
            subplot(3, 3, (i-1)*3 + 3); imshow(overlayImage); title(['Overlay
on Original Image' num2str(i)]);
        end
    end
end
```

1.2 Screenshots

Figure 1- Main GUI interface for uploading images, calculating contrast values, and accessing additional analysis tools.

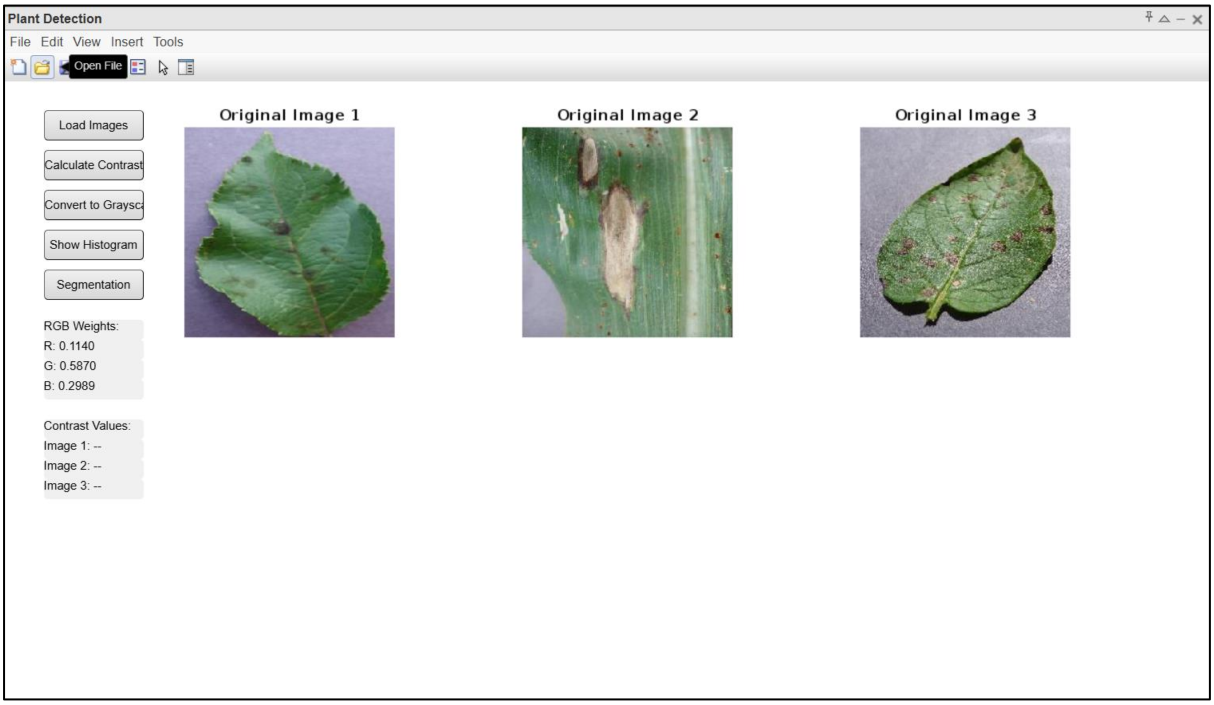

Figure 2 - Interface for selecting and displaying up to three images for processing.

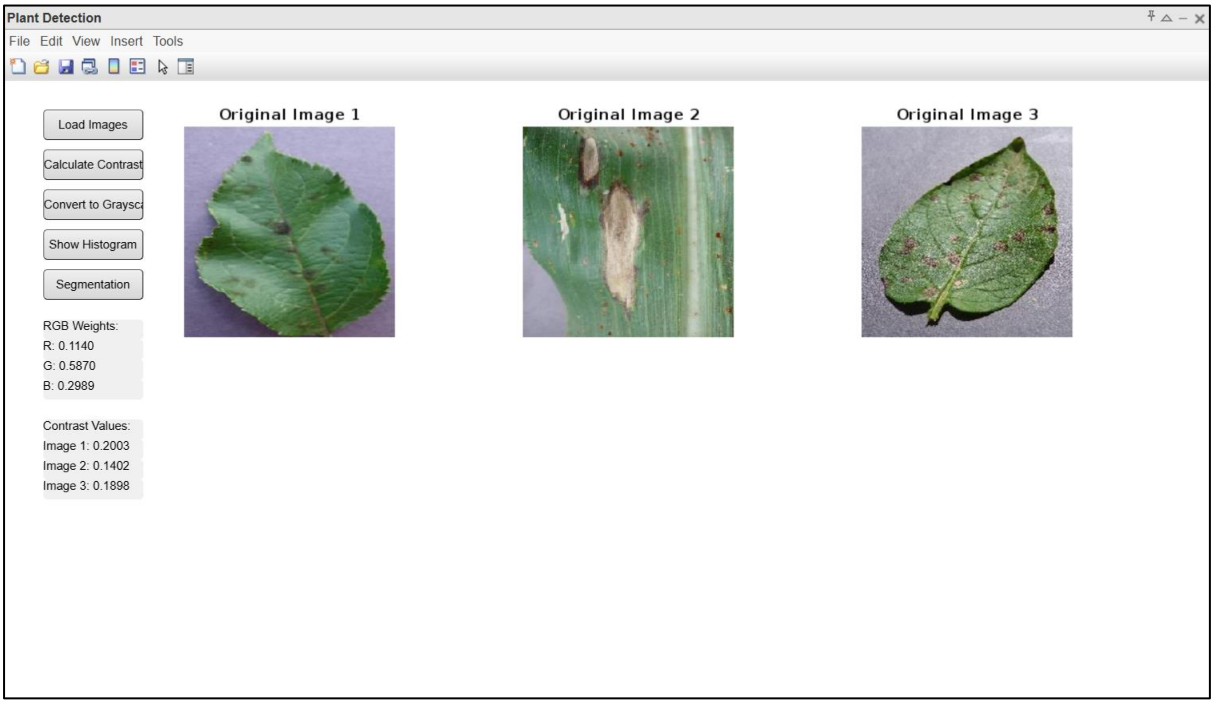

Figure 3 - Functionality to compute and display the contrast values of the uploaded images.

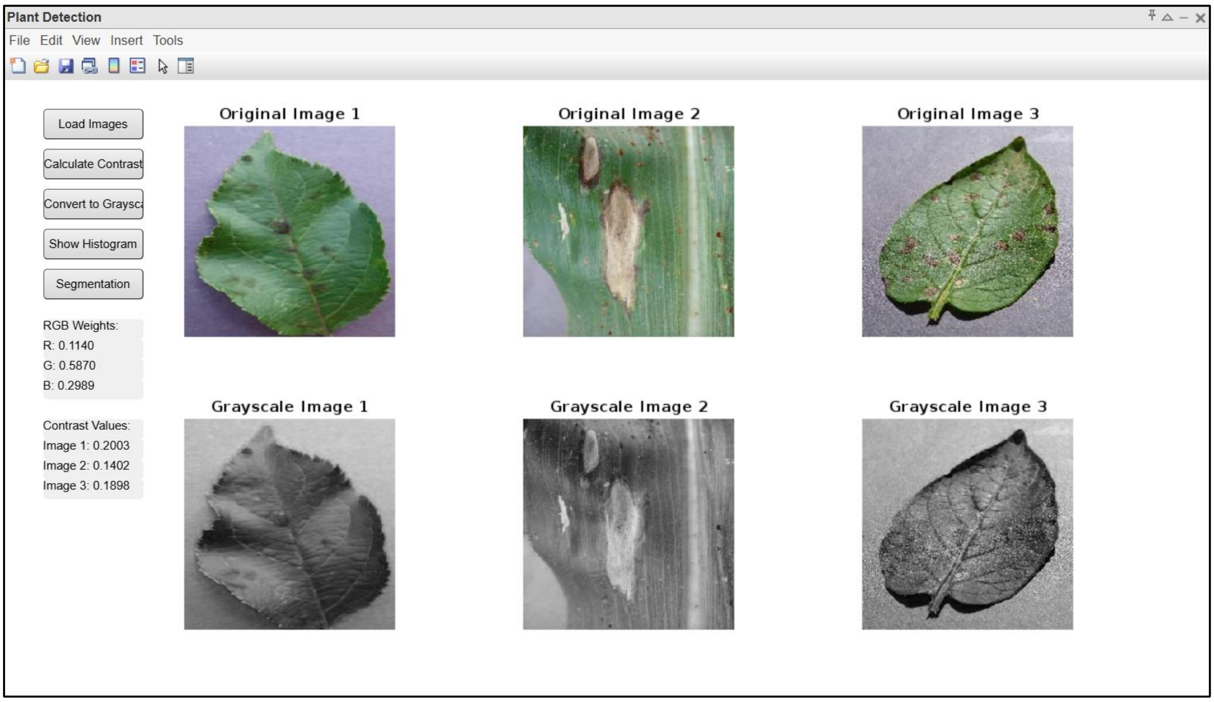

Figure 4 - Feature to convert and display the uploaded images in grayscale format.

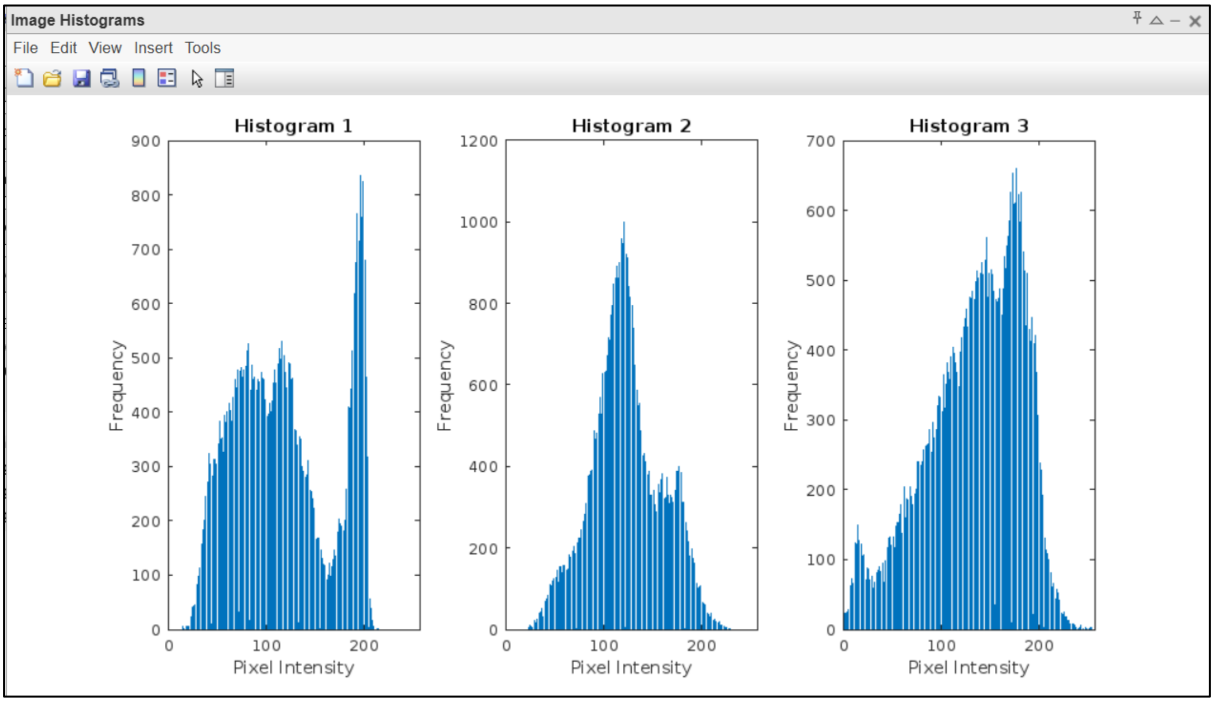

Figure 5 - Visual representation of pixel intensity distribution for the grayscale image.

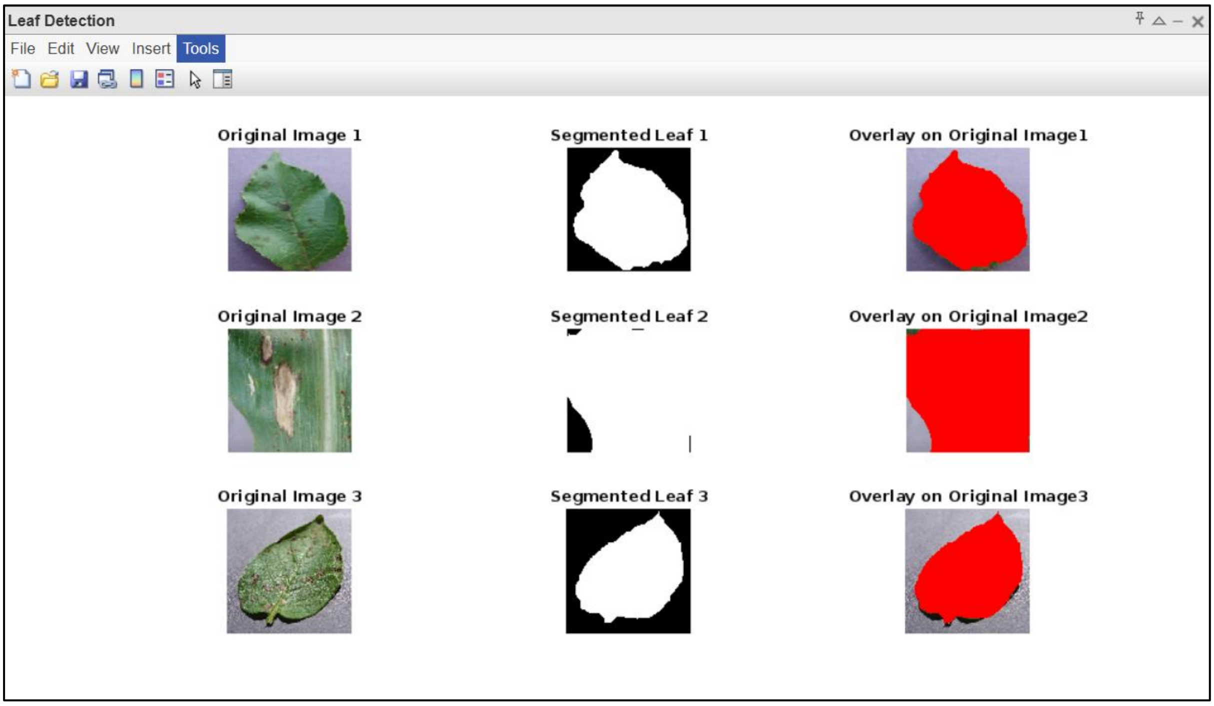

Figure 6 - Specialized interface for detecting and analyzing leaf structures in uploaded images.

2 Part - 2

Plant Disease Detection is one among the projects that use smart sensors and image processing for the identification of diseases in leaves of plants Thus the proposed project has vast potential for environmental sustainability and improvement of the society. The system is set to change the course of agriculture by diagnosing plant diseases early and treating them appropriately while addressing ethical and societal issues as well as the environment.

Environmental Impact

In a great way, the project supports sustainable agriculture by minimizing the impact of typical farming practices on the natural environment. The fact is that early detection enables farmers to spray only the spots that require pesticides and herbicides, reducing the volume used. This avoids pollutive flow of water bodies and decimation of soil health, in addition to maintaining the health of ecosystems and microorganisms such as pollinators. In addition, by controlling large scale crop failure, the project encourages proper farming procedures that will not drain a lot of energy and resources hence reducing the energy wastage and therefore helping in the fight against climate change.

Societal Impact

There are also significant social impacts of this project. This helps in preventing crop damages. Farmers gain economically through increased production and reduced expenses that enhances their living standards and the development of communities. Furthermore, incorporating the application of smart technologies to agricultural practices creates challenges that lead to innovation and assists farmers in embracing new techniques, which fosters the application of technological change between the application of old and traditional farming techniques and the modern technologies.

Ethical Concerns and Resolutions

There are various ethical issues that the project brings out and which should not be overlooked. Data privacy is a major concern, since many of the gathered data points, including location and environmental ones, may be used to violate privacy. To this end, compliance with data protection regulations and subsequent anonymization of collected data is inevitable.

Availability represents another problem because such tools can lack affordability or awareness among smallholders. Local organisations can be partnered with so that they provide subsidies and educational programs that can help reduce this gap. system bias is that all does not benefit when the performance difference is based on regions or crops. With validation usually performed on several datasets, incorporating them is a good way to maintain both inclusiveness and reliability. Ensuring that ethical practices are practiced, the project aims at ensuring that solutions that are open source or financially affordable are given out. In terms of algorithms, and data processing, equal concerns are achieved through the disclosure of data processing algorithms and models, and providing opportunities for stakeholder feedback.

The Plant Disease Detection project aligns with technology, ethical considerations and sustainability. This way, it assists in promoting agriculture progress sustainably, though addressing environmental issues, improving the standard of living of the society and maintaining high ethical standards. These enhancements highlighted the significance of advancing new ideas to create improvements to some of the pressing international issues leading to healthier globe and spirited communities.

References

Liu, Jun, and Xuewei Wang. "Plant Diseases and Pests Detection Based on Deep Learning: A Review." *Plant Methods*, vol. 17, no. 1, 24 Feb. 2021, Available at - https://www.sciencedirect.com/science/article/abs/pii/S0959652620347016?via%3Dihub [Accessed 7th January 2025].

Rani, Lata, et al. "An Extensive Review on the Consequences of Chemical Pesticides on Human Health and Environment." *Journal of Cleaner Production*, vol. 283, no. 124657, 14 Oct. 2020, p. 124657, Available at - www.sciencedirect.com/science/article/pii/S0959652620347016#sec6 [Accessed 8th January 2025].

Scharr, Hanno, et al. "Leaf Segmentation in Plant Phenotyping: A Collation Study." *Machine Vision and Applications*, vol. 27, no. 4, 12 Dec. 2015, pp. 585–606, Available at – https://link.springer.com/article/10.1007/s00138-015-0737-3 [Accessed 4th January 2025].

***