**MATLAB CODE FOR FRUIT DETECTION**

s=serialport("COM9",9600)

% Define the paths to the folders containing the images of healthy and diseased apples

healthyFolder = 'my\_images/healthy';

diseasedFolder = 'my\_images/diseased';

% Load the images from the folders and extract features

healthyDir = dir(fullfile(healthyFolder, '\*.jpg'));

diseasedDir = dir(fullfile(diseasedFolder, '\*.jpg'));

healthyImages = [];

diseasedImages = [];

for i = 1:length(healthyDir)

img = imread(fullfile(healthyFolder, healthyDir(i).name));

img = imresize(img, [256 256]);

healthyImages{i} = img;

end

for i = 1:length(diseasedDir)

img = imread(fullfile(diseasedFolder, diseasedDir(i).name));

img = imresize(img, [256 256]);

diseasedImages{i} = img;

end

% Capture a single frame from the camera

cam = webcam(1);

frame = snapshot(cam);

clear cam;

% Convert the frame to grayscale and enhance contrast

grayImage = im2gray(frame);

enhancedImage = imadjust(grayImage);

% Apply median filtering and thresholding to reduce noise and obtain a binary image

filteredImage = medfilt2(enhancedImage, [3 3]);

threshold = graythresh(filteredImage);

bwImage = im2bw(filteredImage, threshold);

% Extract color and texture features from the image

redChannel = frame(:,:,1);

greenChannel = frame(:,:,2);

blueChannel = frame(:,:,3);

colorFeatures = [mean(redChannel(:)), mean(greenChannel(:)), mean(blueChannel(:))];

GLCM = graycomatrix(filteredImage);

stats = graycoprops(GLCM);

textureFeatures = [stats.Contrast, stats.Correlation, stats.Energy, stats.Homogeneity];

% Compute distances between the features of the captured image and the healthy/diseased images

healthyDistances = [];

diseasedDistances = [];

for i = 1:length(healthyImages)

healthyFeatures = extractFeatures(healthyImages{i});

distance = pdist2([colorFeatures, textureFeatures], healthyFeatures);

healthyDistances(i) = distance;

end

for i = 1:length(diseasedImages)

diseasedFeatures = extractFeatures(diseasedImages{i});

distance = pdist2([colorFeatures, textureFeatures], diseasedFeatures);

diseasedDistances(i) = distance;

end

% Classify the image as either healthy or diseased

tolerance = 0.5; % set the tolerance factor

healthyThreshold = mean(healthyDistances) + std(healthyDistances)\*tolerance;

diseasedThreshold = mean(diseasedDistances) - std(diseasedDistances)\*tolerance;

if (any(min(healthyDistances) < healthyThreshold) && all(min(diseasedDistances) > diseasedThreshold))

% If the image is closer to the healthy images than to the diseased images

% and the distance to the closest diseased image is greater than the threshold,

% classify the image as healthy

diseaseDetected = false;

else

diseaseDetected = true;

end

% Control the relay based on the detection result

if (diseaseDetected)

write(s, 'A','char'); % turn on the relay

else

write(s,'B','char')

end

% Clean up

clear s;