SCHOOL OF ENGINEERING

Project Report

on

"*Be My Eye*"

Submitted for the course

Skill Enhancement I

in

3rd Semester of Bachelor of Technology

in

ELECTRONICS AND COMMUNICATION ENGINEERING

by

Chethan Raj U

USN: ENG22EC0014

1. ***“ Be-My-Eye ”***

Aim:

To Detect Obstacles and

Alert the Blind Person wearing the Camera

Theory:

We have all hurt our legs in the places we daily walk for years yet we tend to hurt ourselves. Now, think of the Blind People. They must walk through everything without seeing anything. They always need assistance even after having heightened power of their present senses in either the way of the stick or a person guiding them.

This project brings these people to the light and provide assistance to them for being independent and able to do various things without the help of others . This project also allows them to live like a normal person. It basically involves :

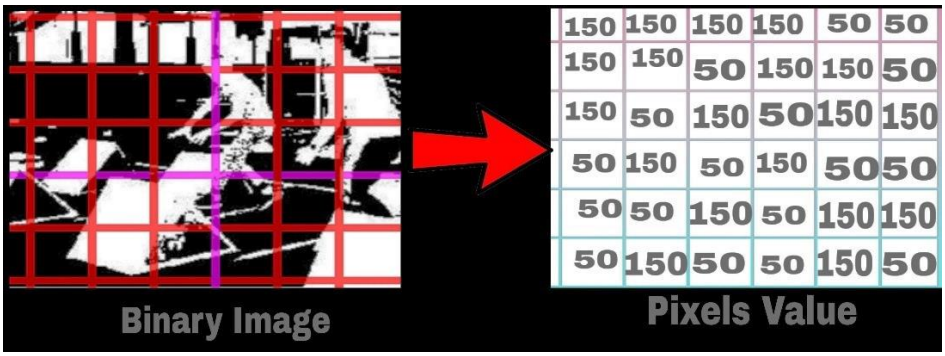
I. *Monitoring The Video Live 24x7 :*
II. *Detecting the Obstacles and Alerting the Person through speaker attached in the device*
III. (*OPTIONAL*) *Processing Words, Images, etc., to guide them better.*

This project uses MATLAB Programming Language for Image and Video Processing of the given Video data by using Matrix Functions for Better Understanding of the data. The Video captured from the camera is first fed to the code and saves it as a RGB video file and also creates a audio output using Zeros matrix with 0 to 44.1K Hertz indicating a "BEEP" for a Obstacle ahead. Then it takes a snapshot from the saved video and sends it to the function called "DetectObstacles" where it detects the obstacles by first converting the image to Grey scale image.

Later, the edges are dilated and converted for Morphological Operations where the Placeholder Logic for detecting true edges in image is applied.

{ *edgepercentage = sum(dilatededges(:)) / numel(dilatededges)* }.

Then, the Threshold is adjusted according the situation and using the threshold value, grey image(The image is divided into pixels and checked if the threshold value is greater than grey image's particular pixel value) and edge percentage, the output is generated (if the output pixel has either white or 1 as its value, it is considered to be TRUE) and returns true to the main function if there is an obstacle and creates a loop to continuously run the program to check the obstacles 24x7.



This is just a simple Program to demonstrate the project whereas in a Real-World project, the complexity of the program increases where the target country and its situations has to be tackled and also the optional features such as indicating the person to move either right or left to avoid obstacle or helping the to read something they point on using image processing can also be included. In this simple program, there is already a given image and the processed images are displayed in the workspace for demonstration of the program processing.

Code:

1. *Main Function (obst.m) :*

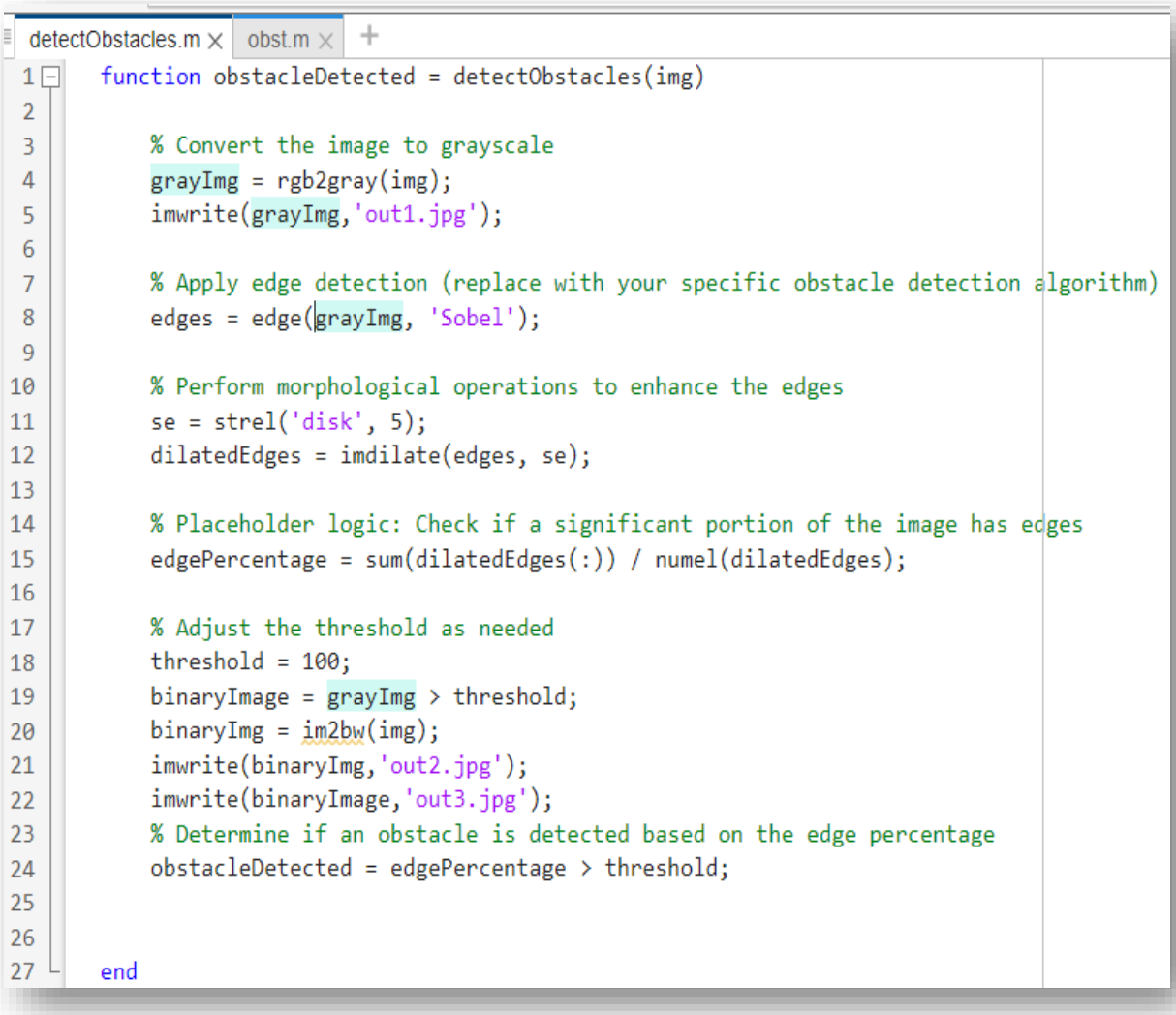
   This section has the video input and producing the output alert for the user where the fumction call is also made.

detectObstacles.m | obst.m

```
% Lifelong Guide for Blind Person
% Ensure you have the necessary Computer Vision and Audio Processing Toolbox installed.

% Create a video input object (assuming webcam)
%vid = videoinput('winvideo', 1, 'RGB');

% Create audio player object
player = audioplayer(zeros(1, 44100), 44100);

% Main loop for continuous image processing
while true
    % Capture image from the webcam
    %img = getsnapshot(vid);
    img = imread('img.jpg');

    % Process the image to detect obstacles (replace with your obstacle detection code)
    obstacleDetected = detectObstacles(img);

    % Display the processed image
    subplot(2,2,1);
    imshow(img);  % Replace with your processed image display code
    title('Original Image');
    % Play a sound if an obstacle is detected
    if obstacleDetected
        play(player);
        fprintf('\n There is a obstacle ahead!! \n');% Replace with your sound output code
    end

    subplot(2,2,2);
    imshow('out1.jpg');
    title('Grey Scale Image');

    subplot(2,2,3);
    imshow('out2.jpg');
    title('Binary Image');

    subplot(2,2,4);
    imshow('out3.jpg');
    title('Processed Image');
    % Add a delay to control the loop rate
    pause(1);  % Adjust as needed
end
```

2. *Sub Function(DetectObstacles):*

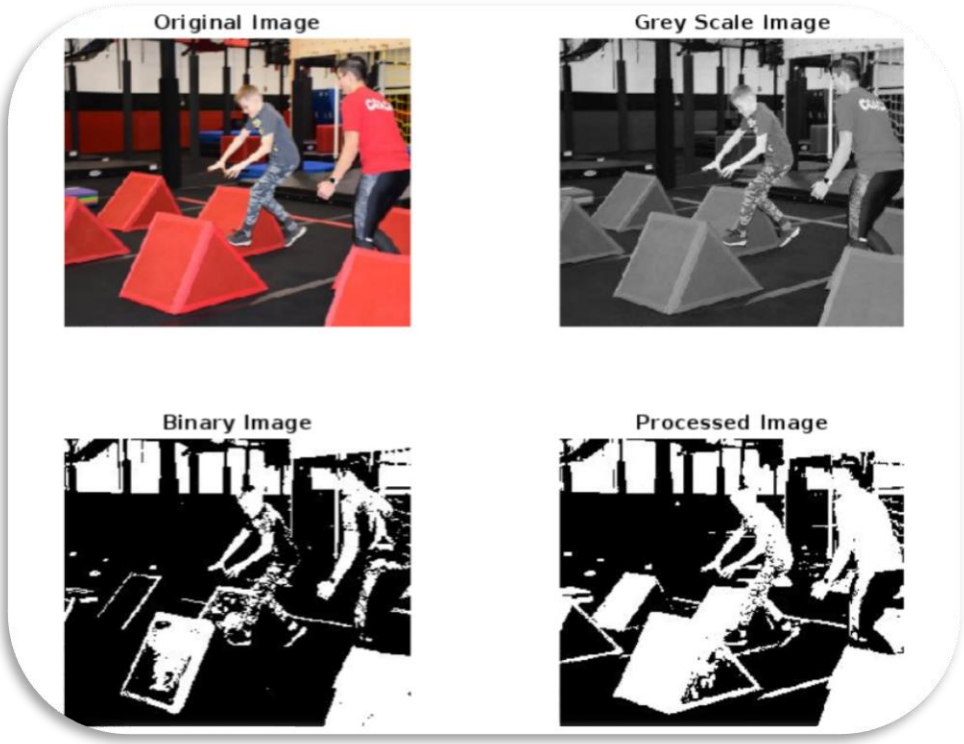
This Function processes the snapshot of the video to detect obstacles using MATLAB where the final image has just white and black patches where white indicates an obstacle ahead alerts the user that there is an obstacle ahead.



detectObstacles.m | obst.m

```
function obstacleDetected = detectObstacles(img)

    % Convert the image to grayscale
    grayImg = rgb2gray(img);
    imwrite(grayImg,'out1.jpg');

    % Apply edge detection (replace with your specific obstacle detection algorithm)
    edges = edge(grayImg, 'Sobel');

    % Perform morphological operations to enhance the edges
    se = strel('disk', 5);
    dilatedEdges = imdilate(edges, se);

    % Placeholder logic: Check if a significant portion of the image has edges
    edgePercentage = sum(dilatedEdges(:)) / numel(dilatedEdges);

    % Adjust the threshold as needed
    threshold = 100;
    binaryImage = grayImg > threshold;
    binaryImg = im2bw(img);
    imwrite(binaryImg,'out2.jpg');
    imwrite(binaryImage,'out3.jpg');
    % Determine if an obstacle is detected based on the edge percentage
    obstacleDetected = edgePercentage > threshold;


end
```

Results:



The synthesis of the above code will result in a modified Binary Image of the snapshot image and here, just to depict the flow the flow of the program, the output is in a form plot of four figures which includes the original image, the grey scale image, the binary image and the final image i.e., Modified Binary Image.