|  |  |
| --- | --- |
|  | **PES UNIVERSITY**  **(Established under Karnataka Act No. 16 of 2013)**  **100 Ft. Road, BSK III Stage, Bengaluru – 560 085**  **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING** |

|  |  |  |
| --- | --- | --- |
| **Course Title: Image Processing and Data Visualization Using MATLAB** | | |
| **Course code: -UE19CS257B** | | |
| **Semester:** 4th sem | **Branch:** Computer Science | **Team Id:** 23 |
| **SRN:** PES1UG19CS134 | **Name:** Deepa S | |
| **SRN:** PES1UG19CS352 | **Name:** Prerana Hadadi | |
| **SRN:** PES1UG19CS486 | **Name:** Smriti Tilak | |

**PROJECT REPORT**

**Problem Statement:** EAN-13 barcode scanner using Image processing in MATLAB.

**Objectives:**

To scan the barcode, extract the digits and then decode it.

**Description:**

* We perform a search on the selected rows of the input image called scanlines.
* We analyse from left to right and from right to left and choose the best match.

The best match is chosen if the checksum is correct and matching score is higher than the set threshold

**New Concept Learnt (Explanation):**

* **Morphological image processing**

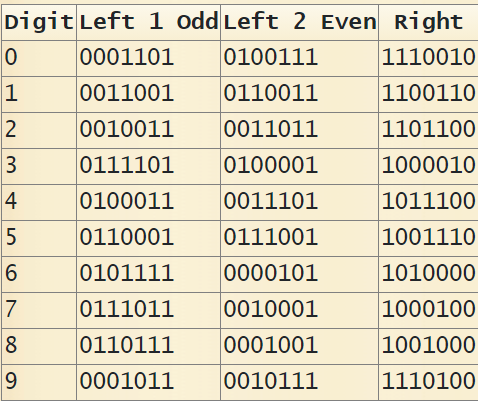
Morphological image processing is a collection of non-linear operations related to the shape or morphology of features in an image. In our project we used a line structuring element to extract the barcode lines and also used the top hat filtering technique.

* **Image re-scaling**

In computer graphics and digital imaging, image scaling refers to**the resizing of a digital image with no loss of image quality. We rescaled our barcode image to only extract the encoded bits.**

* **Algorithm for decoding the barcode**

The EAN-13 barcode encodes 13-digit numbers via 7-bit binary codes using three correspondence tables:



A complete EAN13 barcode begins and ends with a *normal guard zone*coded 101 and contains a *central guard zone*coded 01010 which separates the first 6 digits of the next 6.

The final encoded value is a *95-bit* binary number.

Reference: [Barcode EAN-13 Cipher - Decoder, Encoder, Solver, Translator (dcode.fr)](https://www.dcode.fr/barcode-ean13)

**Learning Outcome:**

We learnt how to use image morphology and scaling in image processing. We learnt about the different types of barcodes and how to extract the digits from an EAN-13 type of barcode.

**Code:**

**main.m:**

clc

clear

close all

%% Image Reading

imgReaded = imread ('img5.png');

imgReaded = rgb2gray (imgReaded);

img = im2double (imgReaded);

figure, imshow (img);

%% Morphology

se = strel ('line', 45, 0); %flat structuring the element

imgMorph = imtophat (~(img), se); %top hat filtering

figure, imshow (imgMorph);

%% Rescale Image

[x,y] = size (imgMorph);

midx = round (x/2);

i = 1;

yinit = i+1;

while (imgMorph (midx, i) == 0)

i=i+1;

yinit = i;

end

i = y;

yend = i-1;

while (imgMorph (midx, i) == 0)

i=i-1;

yend = i;

end

imgRescaled = imgMorph (midx:midx, yinit:yend);

figure, imshow (imgRescaled);

imgRescaled = imresize (imgRescaled, [1 10\*95]);

figure, imshow (imgRescaled);

imgBits = zeros (95);

for i=1:95

imgBits(1,i) = imgRescaled(1, 10\*(i-1)+5);

end;

imgRescaled = imgBits(1,:);

%% BarCode Decodification

% Check C1 - guard code - [1 0 1]

digit = 1;

C1 = imgRescaled (digit:digit+2);

digit = digit + 3;

if (C1 ~= [1 0 1])

disp ('Error on C1');

end;

ean13 = [0 0 0 0 0 0 0 0 0 0 0 0 0];

ean13 (2) = EAN13digits(imgRescaled(digit: digit+6));

digit = digit + 7;

ean13 (3) = EAN13digits(imgRescaled(digit: digit+6));

digit = digit + 7;

ean13 (4) = EAN13digits(imgRescaled(digit: digit+6));

digit = digit + 7;

ean13 (5) = EAN13digits(imgRescaled(digit: digit+6));

digit = digit + 7;

ean13 (6) = EAN13digits(imgRescaled(digit: digit+6));

digit = digit + 7;

ean13 (7) = EAN13digits(imgRescaled(digit: digit+6));

digit = digit + 7;

% Check C2 - guard code - [0 1 0 1 0]

C2 = imgRescaled (digit:digit+4);

digit = digit + 5;

if (C2 ~= [0 1 0 1 0])

disp ('Error on C2');

end;

ean13 (8) = EAN13digits(imgRescaled(digit: digit+6));

digit = digit + 7;

ean13 (9) = EAN13digits(imgRescaled(digit: digit+6));

digit = digit + 7;

ean13 (10) = EAN13digits(imgRescaled(digit: digit+6));

digit = digit + 7;

ean13 (11) = EAN13digits(imgRescaled(digit: digit+6));

digit = digit + 7;

ean13 (12) = EAN13digits(imgRescaled(digit: digit+6));

digit = digit + 7;

ean13 (13) = EAN13digits(imgRescaled(digit: digit+6));

digit = digit + 7;

%% Checksum Digit

mult = [3 1 3 1 3 1 3 1 3 1 3 1];

checkDigit = ean13 (2:13).\*mult; %element-wise multiplication

c\_checkDigit = sum(checkDigit);

sub = ceil(c\_checkDigit / 10) \* 10;

checkDigit = sub - c\_checkDigit;

ean13(1) = checkDigit;

ean13str = mat2str (ean13);

disp ('The value is ');

disp (ean13str);

**EAN13digits.m:**

function [number, code] = EAN13digits( vector7)

if isequal(vector7, [0 0 0 1 1 0 1])

number = 0; code = 1; return;

elseif isequal(vector7, [0 1 0 0 1 1 1])

number = 0; code = 2; return;

elseif isequal(vector7, [1 1 1 0 0 1 0])

number = 0; code = 3; return;

elseif isequal(vector7, [0 0 1 1 0 0 1])

number = 1; code = 1; return;

elseif isequal(vector7, [0 1 1 0 0 1 1])

number = 1; code = 2; return;

elseif isequal(vector7, [1 1 0 0 1 1 0])

number = 1; code = 3; return;

elseif isequal(vector7, [0 0 1 0 0 1 1])

number = 2; code = 1; return;

elseif isequal(vector7, [0 0 1 1 0 1 1])

number = 2; code = 2; return;

elseif isequal(vector7, [1 1 0 1 1 0 0])

number = 2; code = 3; return;

elseif isequal(vector7, [0 1 1 1 1 0 1])

number= 3; code = 1; return;

elseif isequal(vector7, [0 1 0 0 0 0 1])

number = 3; code = 2; return;

elseif isequal(vector7, [1 0 0 0 0 1 0])

number = 3; code = 3; return;

elseif isequal(vector7, [0 1 0 0 0 1 1])

number = 4; code = 1; return;

elseif isequal(vector7, [0 0 1 1 1 0 1])

number = 4; code = 2; return;

elseif isequal(vector7, [1 0 1 1 1 0 0])

number = 4; code = 3; return;

elseif isequal(vector7, [0 1 1 0 0 0 1])

number = 5; code = 1; return;

elseif isequal(vector7, [0 1 1 1 0 0 1])

number = 5; code = 2; return;

elseif isequal(vector7, [1 0 0 1 1 1 0])

number = 5; code = 3; return;

elseif isequal(vector7, [0 1 0 1 1 1 1])

number = 6; code = 1; return;

elseif isequal(vector7, [0 0 0 0 1 0 1])

number = 6; code = 2; return;

elseif isequal(vector7, [1 0 1 0 0 0 0])

number = 6; code = 3; return;

elseif isequal(vector7, [0 1 1 1 0 1 1])

number = 7; code = 1; return;

elseif isequal(vector7, [0 0 1 0 0 0 1])

number = 7; code = 2; return;

elseif isequal(vector7, [1 0 0 0 1 0 0])

number = 7; code = 3; return;

elseif isequal(vector7, [0 1 1 0 1 1 1])

number = 8; code = 1; return;

elseif isequal(vector7, [0 0 0 1 0 0 1])

number = 8; code = 2; return;

elseif isequal(vector7, [1 0 0 1 0 0 0])

number = 8; code = 3; return;

elseif isequal(vector7, [0 0 0 1 0 1 1])

number = 9; code = 1; return;

elseif isequal(vector7, [0 0 1 0 1 1 1])

number = 9; code = 2; return;

elseif isequal(vector7, [1 1 1 0 1 0 0])

number = 9; code = 3; return;

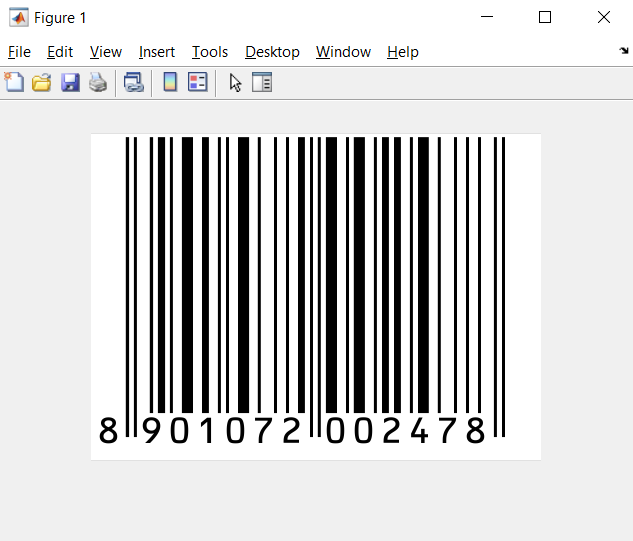
else number = -2; code = 0; return;

end;

end

**Output Screenshots**

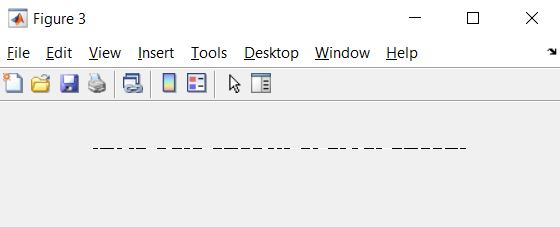
**Original enhanced image**

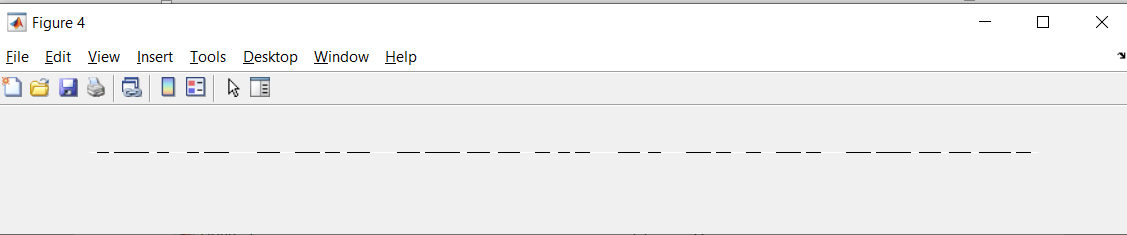
****

**Image obtained after morphological image processing and top hat filtering**

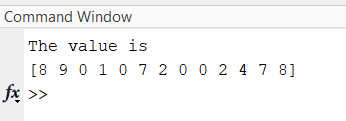
****

**Images obtained after rescaling to extract just the encoded bits**

****

****

**Final Output**

****

**Name and Signature of the Faculty**