VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

June 2020

CANNY EDGE DETECTION AND RECOGNITION OF FRUITS USING ALEXNET

A Project Report

Under the guidance of

Prof. B. Gladys Gnana Kiruba

By

18BCE2434-Samartha Pal
18BCE2438-Aakrit Sharma Lamsal
18BCE2439-Aryak Dangol
18BCE2442-Shradaya Sthapit

DECLARATION BY THE CANDIDATE

We hereby declare that the project report entitled "**CANNY EDGE DETECTION AND RECOGNITION OF FRUITS USING ALEXNET**" submitted by us to Vellore Institute of Technology, Vellore in partial fulfilment of the requirement for the award of the degree of B. Tech (B. Tech CSE) is a record of J- component of project work carried out by us under the guidance of Prof. B.Gladys Gnana Kiruba. We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore Institute of Technology, Vellore.

Date : Thursday, June 4, 2020

Signature of the faculty

Signature of the Candidate

TABLE OF CONTENTS

1. Introduction

1.1 ABSTRACT

In this project, a method to eliminate and suppress speckle noise which usually affects medical images such as ultrasounds and Synthetic Aperture Radar (SAR) images is proposed. Speckle noises are multiplicative in nature. Noise removal is one of the most important steps in the field of image processing as it removes unwanted data and preserves information from the images that are actually useful. The method used in this experiment works in two phases. The first phase uses a fuzzy inference system to classify the image into various regions using coefficient of variance and gradient magnitude as inputs and the second phase involves applying different fuzzy based filters to the regions so as to provide edge preservation while suppressing speckle noise. Fuzzy logic based technique is used because of its ability to accommodate inexact and approximate data instead of crisp values. The results of the experiment are evaluated using metrics such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM).

1.2. BACKGROUND

Edge detection includes a variety of mathematical methods that aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed *edges*. The same problem of finding discontinuities in one-dimensional signals is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction . John Canny considered the mathematical problem of deriving an optimal smoothing filter given the criteria of detection, localization and minimizing multiple responses to a single edge. He showed that the optimal filter given these assumptions is a sum of four exponential terms. He also showed that this filter can be well approximated by first-order derivatives of Gaussians. Canny also introduced the notion of non-maximum suppression, which means that given the pre-smoothing

filters, edge points are defined as points where the gradient magnitude assumes a local maximum in the gradient direction. For automatic fruit/vegetable harvesting systems, it is extremely important to effectively detect the object in outdoor conditions. There are several problems on fruit detection in outdoor condition, which can be classified into two groups: lighting and occlusion. Overcoming these problems is very crucial for the success of robotic harvesting. The first major task of a harvesting robot is to recognize and localize the fruit on the tree.

2. Overview and Planning

2.1 Proposed Work

With the vigorous development of fruit industry, the use of effective technical methods to classify all kinds of fruits is a general trend. As we all know, manual check is not popular any more, we should apply pattern recognition methods to deal with this problem. The emergence and development of pattern recognition technology is based on the people who use visual and auditory to identify various information. Pattern recognition is a state-of-the-art technique to process complex information automatically using computer and mathematical theory. For these reasons, the researchers use pattern recognition as an intelligent technology to replace and even expand human daily mental activities. The Canny method finds edges by looking for local maxima of the gradient of I. The gradient is calculated using the derivative of a Gaussian filter. The method uses two thresholds, to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. This method is therefore less likely than the others to be fooled by noise, and more likely to detect true weak edges. The Canny edge detector is a popular method for finding edges that begins by smoothing an image by convolving it with a Gaussian of a given sigma value.

Implementing an efficient methodology to edge detection process detects outlines of fruits and boundaries between objects and the background in the image. The fruit size is another quality attribute used by farmers – the bigger size of some fruits is considered of better quality.Developing a high-performance fruit detection system that can be rapidly trained with a small number of images using a DCNN that has been pre-trained on a large dataset, such as AlexNet.

This project tends to investigate the performance of basic Convolutional Neural Network (CNN), Alexnet and Googlenet in recognizing different types of fruits from a publicly available dataset. The experimental results indicate that all these techniques produce excellent recognition accuracy, but basic CNN achieves the fastest recognition result compared with Alexnet and Googlenet. Quality

inspection of fruits and vegetables using image processing technique within Alexnet involves various steps, as depicted namely, convolution, max pooling, image layering etc.

2.2 Hardware Requirements

The hardware required is PC/laptop.

2.3 Software Requirements

The required software is MATLAB for running the program, and web browser to access the internet and train CNN using the AlexNet.

3. Literature Survey and Review

3.1 Literature Summary

- An image segmentation approach for fruit defect detection using k-means clustering and graph-based algorithm
  -Van Huy Pham, Byung Ryong Lee

In this paper, a hybrid algorithm, which is based on split and merge approach, is proposed for an image segmentation that can be used in fruit defect detection. The algorithm firstly uses k- means algorithm to split the original image into regions based on Euclidean color distance in L∗a∗b∗ space to produce an over-segmentation result. Then, based on a graph representation, a merge procedure using minimum spanning tree is then taken into account to iteratively merge similar regions into new homogenous ones. This combination is an efficient approach to employ the local and global characteristic of intensities in the image. The experiment showed good results in the terms of human observation and in processing time.

- A Study on Image Processing Methods for Fruit Classification
  -D Surya Prabha and J Satheesh Kumar

The classification of fruits also provides benefits in quality evaluation and defect finding. Color and shape are primary properties of fruit images which help

for better classification. The size of the image is also considered as another criterion to improve the accuracy. The uniformity in the classification of fruit is determined based on color, size and shape . Classification of fruits can be based either on each of these aspects or on a combined decision considering all of them. In recent years, computer vision, machine vision and image processing techniques have been found increasingly useful in the fruit industry. Applications of image processing are wider in range and its application in agriculture sector includes monitoring the growth of crops, controlling weeds, grading and sorting of fruits and vegetables and classification of the fruits and vegetables etc. This paper reviews the image processing methods used in fruit classification.

- Application of Image Processing in Fruit and Vegetable Analysis: A Review

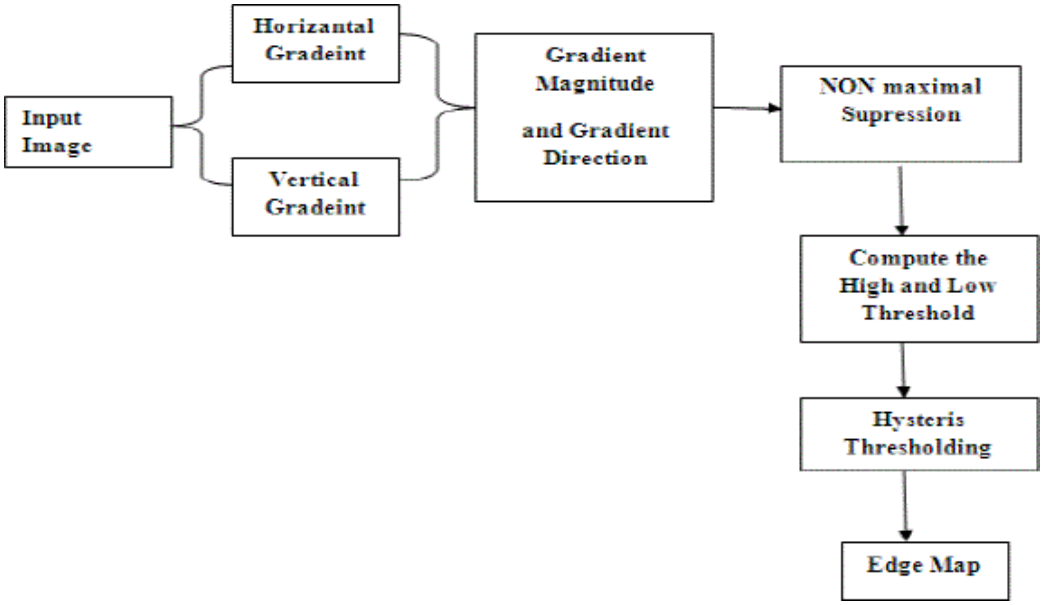
-Shiv Ram Dubey,Anand Singh

Images are an important source of data and information in the agricultural sciences. The use of image-processing techniques has outstanding implications for the analysis of agricultural operations. Fruit and vegetable classification is one of the major applications that can be utilized in supermarkets to automatically detect the kinds of fruits or vegetables purchased by customers and to determine the appropriate price for the produce. Training on-site is the underlying prerequisite for this type of arrangement, which is generally caused by the users having little or no expert knowledge. They explored various methods used in addressing fruit and vegetable classification and in recognizing fruit disease problems. They surveyed image- processing approaches used for fruit disease detection, segmentation and classification. We also compared the performance of state-of-the-art methods under two scenarios, i.e., fruit and vegetable classification and fruit disease classification. The methods surveyed in this paper are able to distinguish among different kinds of fruits and their diseases that are very alike in color and texture.

4. Methodology

4.1 Method Used

The canny edge detector first smoothens the image to eliminate noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum using non-maximum suppression. The gradient array is now further reduced by hysteresis to remove streaking and thinning the edges.



- **Canny edge detection algorithm**

AlexNet is a convolutional neural network that is trained on more than a million images from the ImageNet database. The network is 8 layers deep and can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 227-by-227.

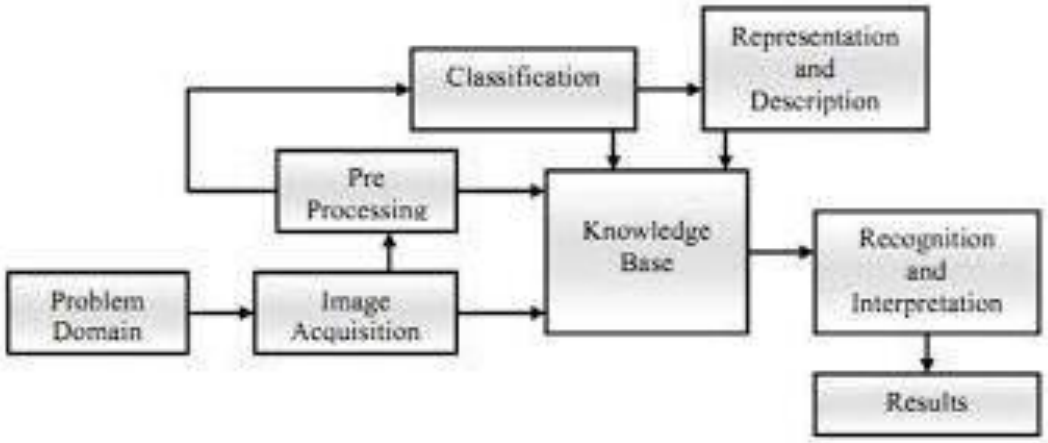


Fig. Image recognition methodology

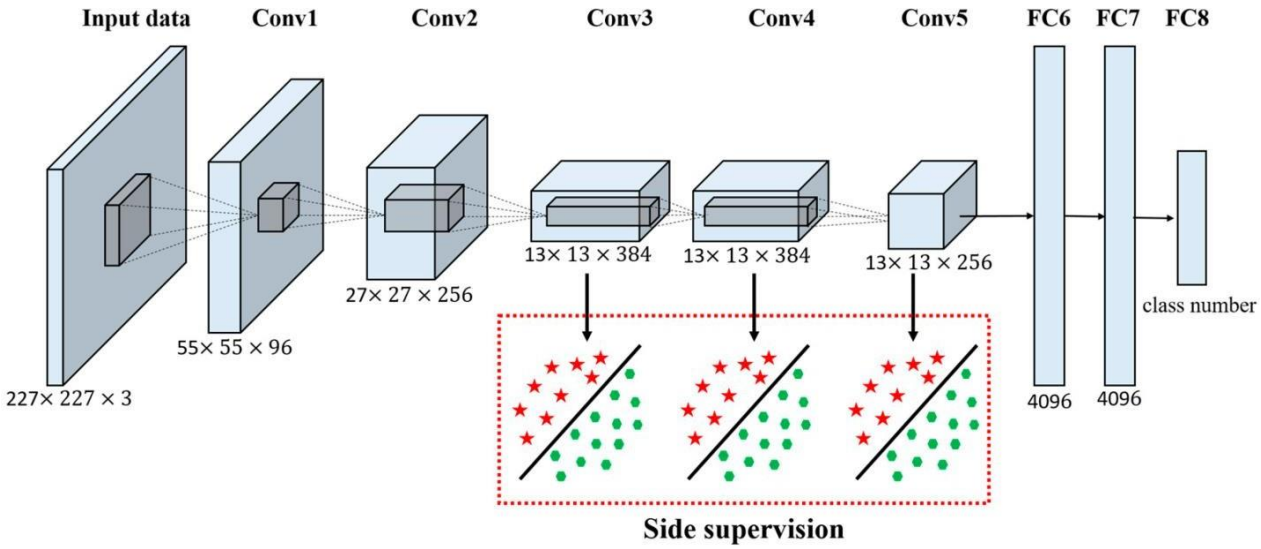


Fig. Alexnet Architechture

The Process of Canny edge detection algorithm can be broken down to 5 different steps:

1. Apply Gaussian filter to smooth the image in order to remove the noise
2. Find the intensity gradients of the image
3. Apply non-maximum suppression to get rid of spurious response to edge detection
4. Apply double threshold to determine potential edges
5. Track edge by hysteresis Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

Gaussian filter

- Since all edge detection results are easily affected by image noise, it is essential to filter out the noise to prevent false detection caused by noise. To smooth the image, a Gaussian filter is applied to convolve with the image. This step will slightly smooth the image to reduce the effects of obvious noise on the edge detector. The equation for a Gaussian filter kernel of size (2k+1)×(2k+1) is given by

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i-(k+1))^2 + (j-(k+1))^2}{2\sigma^2} \right); 1 \leq i, j \leq (2k+1)$$

Here is an example of a 5×5 Gaussian filter, used to create the adjacent image, with 1.4.(The asterisk denotes a convolution operation.)

$$\mathbf{B} = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * \mathbf{A}.$$

It is important to understand that the selection of the size of the Gaussian kernel will affect the performance of the detector. The larger the size is, the lower the detector's sensitivity to noise. Additionally, the localization error to detect the edge will slightly increase with the increase of the Gaussian filter kernel size. A 5×5 is a good size for most cases, but this will also vary depending on specific situations.

Finding the intensity gradient of the image

An edge in an image may point in a variety of directions, so the Canny algorithm uses four filters to detect horizontal, vertical and diagonal edges in the blurred image. The edge detection operator (such as Roberts, Prewitt, or Sobel) returns a value for the first derivative in the horizontal direction (Gx) and the vertical direction (Gy). From this the edge gradient and direction can be determined:

$$\mathbf{G} = \sqrt{\mathbf{G}_x^{\,2} + \mathbf{G}_y^{\,2}}$$
$$\mathbf{\Theta} = \operatorname{atan2}(\mathbf{G}_y, \mathbf{G}_x),$$

where G can be computed using the hypot function and atan2 is the arctangent function with two arguments. The edge direction angle is rounded to one of four angles representing vertical, horizontal and the two diagonals (0°, 45°, 90° and 135°). An edge direction falling in each color region will be set to a specific angle values, for instance θ in [0°, 22.5°] or [157.5°, 180°] maps to 0°.

Non-maximum suppression

Non-maximum suppression is an edge thinning technique.

Non-maximum suppression is applied to find "the largest" edge. After applying gradient calculation, the edge extracted from the gradient value is still quite blurred. With respect to criterion 3, there should only be one accurate response to the edge. Thus non-maximum suppression can help to suppress all the gradient values (by setting them to 0) except the local maxima, which indicate locations with the

sharpest change of intensity value. The algorithm for each pixel in the gradient image is:

- Compare the edge strength of the current pixel with the edge strength of the pixel in the positive and negative gradient directions.
- Compare the edge strength of the current pixel with the edge strength of the pixel in the positive and negative gradient directions.

In some implementations, the algorithm categorizes the continuous gradient directions into a small set of discrete directions, and then moves a 3x3 filter over the output of the previous step (that is, the edge strength and gradient directions). At every pixel, it suppresses the edge strength of the center pixel (by setting its value to 0) if its magnitude is not greater than the magnitude of the two neighbors in the gradient direction. For example,

- if the rounded gradient angle is 0° (i.e. the edge is in the north-south direction) the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes at pixels in the east and west directions,
- if the rounded gradient angle is 90° (i.e. the edge is in the east-west direction) the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes at pixels in the north and south directions
- if the rounded gradient angle is 135° (i.e. the edge is in the northeast-southwest direction) the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes at pixels in the north west and south-east directions,
- if the rounded gradient angle is 45° (i.e. the edge is in the north west–south east direction) the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes at pixels in the north east and south west directions.

In more accurate implementations, linear interpolation is used between the two neighboring pixels that straddle the gradient direction. For example, if the gradient angle is between 89° and 180°, interpolation between gradients at the north and north east pixels will give one interpolated value, and interpolation between the south and south west pixels will give the other (using the conventions of the last paragraph). The gradient magnitude at the central pixel must be greater than both of these for it to be marked as an edge.

Double threshold

After application of non-maximum suppression, remaining edge pixels provide a more accurate representation of real edges in an image. However, some edge pixels remain that are caused by noise and color variation. In order to account for these spurious responses, it is essential to filter out edge pixels with a weak gradient value and preserve edge pixels with a high gradient value. This is accomplished by selecting high and low threshold values. If an edge pixel's gradient value is higher than the high threshold value, it is marked as a strong edge pixel. If an edge pixel's gradient value is smaller than the high threshold value and larger than the low threshold value, it is marked as a weak edge pixel. If an edge pixel's value is smaller than the low threshold value, it will be suppressed. The two threshold values are empirically determined and their definition will depend on the content of a given input image.

Edge tracking by hysteresis

So far, the strong edge pixels should certainly be involved in the final edge image, as they are extracted from the true edges in the image. However, there will be some debate on the weak edge pixels, as these pixels can either be extracted from the true edge, or the noise/color variations. To achieve an accurate result, the weak edges caused by the latter reasons should be removed. Usually a weak edge pixel caused from true edges will be connected to a strong edge pixel while noise responses are unconnected. To track the edge connection, blob analysis is applied by looking at a weak edge pixel and its 8-connected neighborhood pixels. As long as there is one strong edge pixel that is involved in the blob, that weak edge point can be identified as one that should be preserved.

4.2 Applications

Our project can be applied to software on mobile phones to classify objects. It can also be applied to medical image processing field where our program will reduce the noise and generates informational results. It can also be deployed in the face detection applications. With the increase in the speed of processors, the future application of our project is not limited to a particular field but in the entire field of intelligent image processing.

5. System Implementation

5.1 Code

AlexNet image detection through neural networking

```
net=alexnet;
I=imread('orange.jpg');
imshow(I)

sz = net.Layers(1).InputSize;

I = I(1:sz(1),1:sz(2),1:sz(3));
figure
imshow(I)

label = classify(net,I)
figure
imshow(I)
title(char(label))
```

Canny Edge detection in matlab code

```
clear all;
clc;
%Input image
img = imread ('appl4.jpg');
%Show input image
figure, imshow(img);
img = rgb2gray(img);
img = double (img);
%Value for Thresholding
T_Low = 0.075;
T_High = 0.175;
%Gaussian Filter Coefficient
B = [2, 4, 5, 4, 2; 4, 9, 12, 9, 4;5, 12, 15, 12, 5;4, 9, 12, 9, 4;2, 4, 5, 4, 2 ];
B = 1/159.* B;
%Convolution of image by Gaussian Coefficient
A=conv2(img, B, 'same');
%Filter for horizontal and vertical direction
KGx = [-1, 0, 1; -2, 0, 2; -1, 0, 1];
KGy = [1, 2, 1; 0, 0, 0; -1, -2, -1];
%Convolution by image by horizontal and vertical filter
Filtered_X = conv2(A, KGx, 'same');
Filtered_Y = conv2(A, KGy, 'same');
%Calculate directions/orientations
arah = atan2 (Filtered_Y, Filtered_X);
arah = arah*180/pi;
```

```
pan=size(A,1);
leb=size(A,2);
%Adjustment for negative directions, making all directions positive
for i=1:pan
   for j=1:leb
      if (arah(i,j)<0)
         arah(i,j)=360+arah(i,j);
      end;
   end;
end;
arah2=zeros(pan, leb);
%Adjusting directions to nearest 0, 45, 90, or 135 degree
for i = 1 : pan
   for j = 1 : leb
      if ((arah(i, j) >= 0 ) && (arah(i, j) < 22.5) || (arah(i, j) >= 157.5) && (arah(i, j) < 202.5) ||
(arah(i, j) >= 337.5) && (arah(i, j) <= 360))
         arah2(i, j) = 0;
      elseif ((arah(i, j) >= 22.5) && (arah(i, j) < 67.5) || (arah(i, j) >= 202.5) && (arah(i, j) <
247.5))
         arah2(i, j) = 45;
      elseif ((arah(i, j) >= 67.5 && arah(i, j) < 112.5) || (arah(i, j) >= 247.5 && arah(i, j) <
292.5))
         arah2(i, j) = 90;
      elseif ((arah(i, j) >= 112.5 && arah(i, j) < 157.5) || (arah(i, j) >= 292.5 && arah(i, j) <
337.5))
         arah2(i, j) = 135;
      end;
   end;
end;
figure, imagesc(arah2); colorbar;
%Calculate magnitude
magnitude = (Filtered_X.^2) + (Filtered_Y.^2);
magnitude2 = sqrt(magnitude);
BW = zeros (pan, leb);
%Non-Maximum Supression
for i=2:pan-1
   for j=2:leb-1
      if (arah2(i,j)==0)
         BW(i,j) = (magnitude2(i,j) == max([magnitude2(i,j), magnitude2(i,j+1),
magnitude2(i,j-1)]));
      elseif (arah2(i,j)==45)
         BW(i,j) = (magnitude2(i,j) == max([magnitude2(i,j), magnitude2(i+1,j-1),
magnitude2(i-1,j+1)]));
      elseif (arah2(i,j)==90)
         BW(i,j) = (magnitude2(i,j) == max([magnitude2(i,j), magnitude2(i+1,j), magnitude2(i-
1,j)]));
      elseif (arah2(i,j)==135)
         BW(i,j) = (magnitude2(i,j) == max([magnitude2(i,j), magnitude2(i+1,j+1),
magnitude2(i-1,j-1)]));
      end;
```

```
    end;
end;
BW = BW.*magnitude2;
figure, imshow(BW);
%Hysteresis Thresholding
T_Low = T_Low * max(max(BW));
T_High = T_High * max(max(BW));
T_res = zeros (pan, leb);
for i = 1 : pan
    for j = 1 : leb
        if (BW(i, j) < T_Low)
            T_res(i, j) = 0;
        elseif (BW(i, j) > T_High)
            T_res(i, j) = 1;
        %Using 8-connected components
        elseif ( BW(i+1,j)>T_High || BW(i-1,j)>T_High || BW(i,j+1)>T_High || BW(i,j-
1)>T_High || BW(i-1, j-1)>T_High || BW(i-1, j+1)>T_High || BW(i+1, j+1)>T_High ||
BW(i+1, j-1)>T_High)
            T_res(i,j) = 1;
        end;
    end;
end;
edge_final = uint8(T_res.*255);
%Show final edge detection result
figure, imshow(edge_final);

net=alexnet;


%% Input Size of Net Layers
sz = net.Layers(1).InputSize;
%% resize the image according to Input size
edge_final = edge_final(1:sz(1),1:sz(2));
figure
imshow(edge_final)
%% Classify the image
label = classify(net,edge_final)
%% Plot the classify image
figure
imshow(edge_final)
title(char(label))
```

5.2 Results and discussion

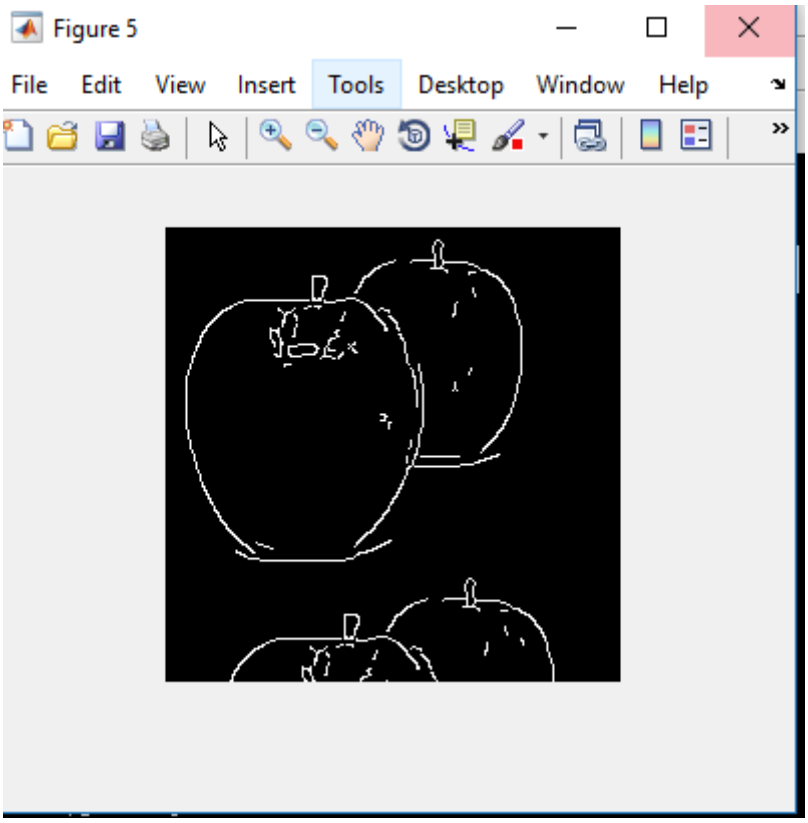
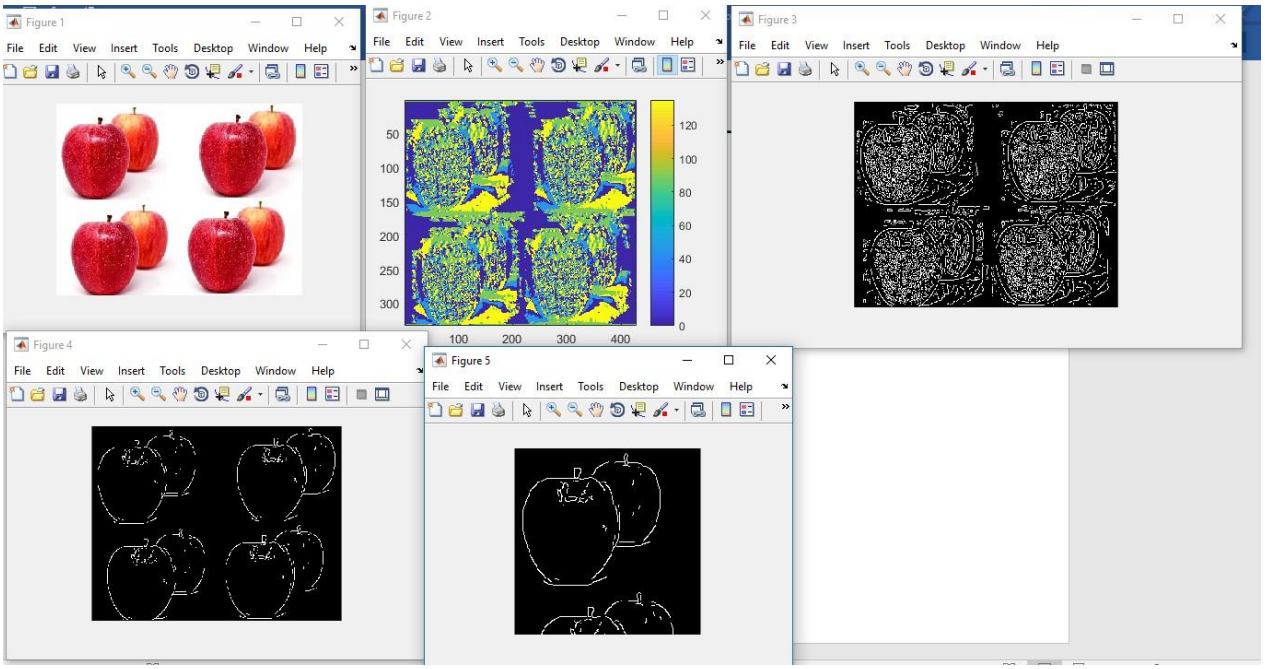


Fig. Canny's output



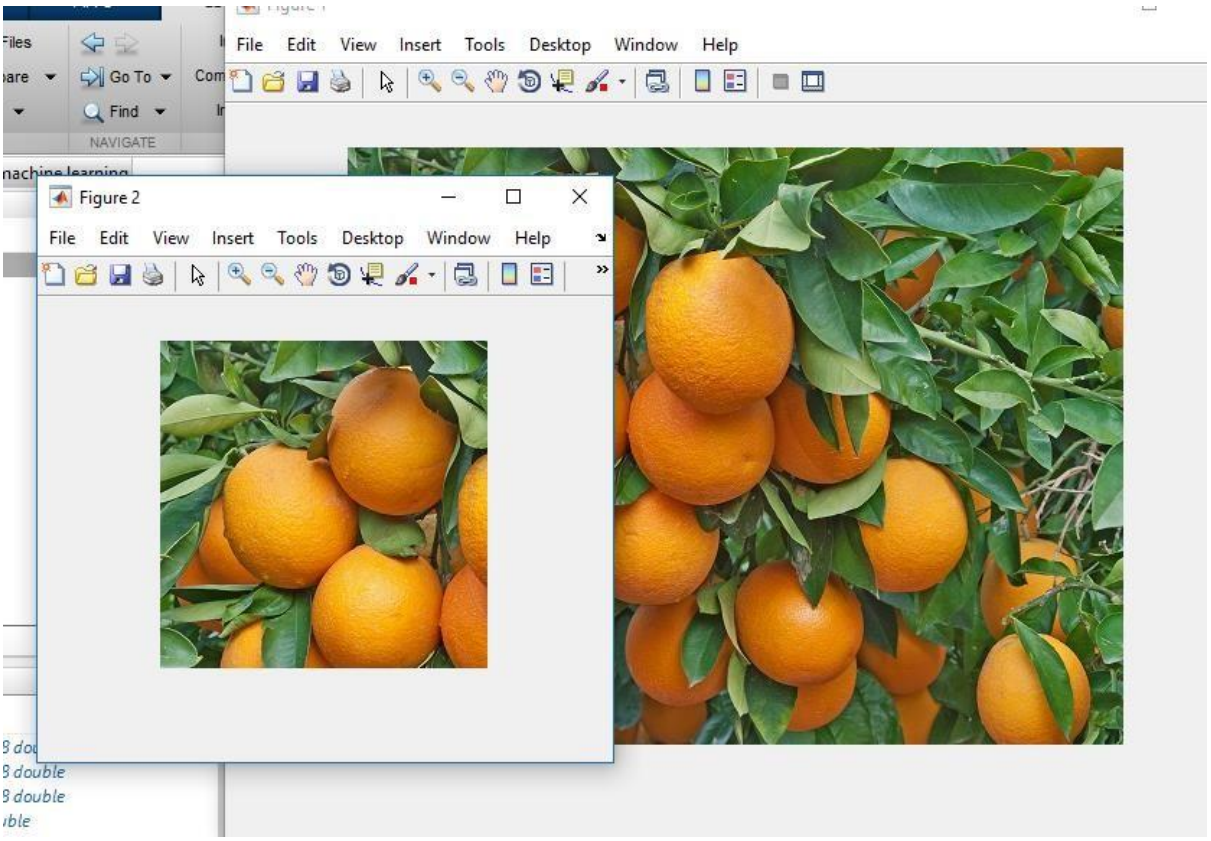


Fig. Input image for image recognition

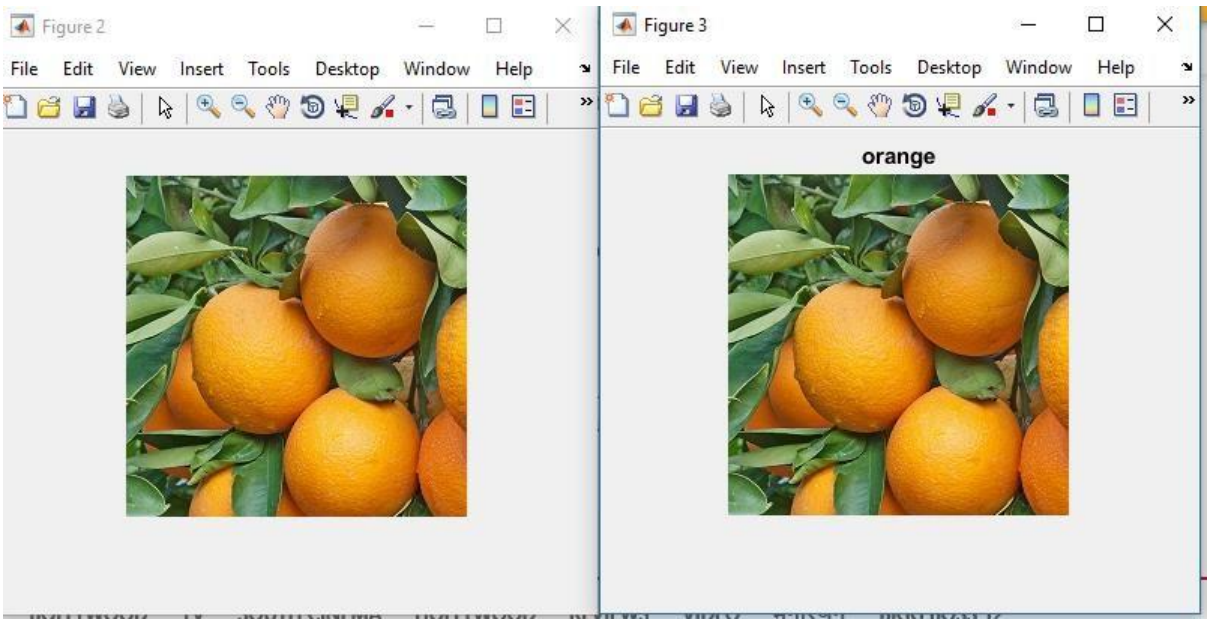


Fig.Output after AlexNet image labelling

We described a new and complex database of images with fruits. Also, we made some numerical experiments by using Alexnet library in order to classify the images according to their content. From our point of view one of the main objectives for the future is to improve the accuracy of the neural network. This involves further experimenting with the structure of the network. Various tweaks and changes to any layers as well as the introduction of new layers can provide completely different results. Another option is to replace all layers with convolutional layers. This has been shown to provide some improvement over the networks that have fully connected layers in their structure.

6. Conclusion

6.1 Conclusion

So, we applied canny edge technique to get the edge of given fruit. After getting edge for the sake of identification we used AlexNet database library. In this way, we increased the efficiency of detection as well as classification of Fruit. We also eliminated and suppressed speckle noise which usually affects medical images such as ultrasounds and Synthetic Aperture Radar (SAR) images.

6.2 Future Work

After applying canny edge technique to similar object. AlexNet or any image database find difficulty in distinguishing between them. For example, if it is lemon and orange the system may provide output either of the two but results may vary in different cases. In future its works will be useful because it requires less data consumption in comparison to existing method. This method will increase the time performance of the system. It is best for real time classification of the work.

7. References

- *Umbaugh, Scott E (2010). Digital image processing and analysis : human and computer vision applications with CVIPtools (2nd ed.). Boca Raton, FL: CRC Press. ISBN 978-1-4398-0205-2.*
- J. Canny (1986) "A computational approach to edge detection", IEEE Trans. Pattern Analysis and Machine Intelligence, vol 8, pages 679–714.
- Y. Sarig, "Mechanized fruit harvesting-Site Specific Solutions," Information and Technology for Sustainable Fruit and Vegetable Production, FRUTIC vol. OS, pp. 237-247,2005.
- https://www.mathworks.com/matlabcentral/fileexchange/46859-canny-edge-detection
- https://www.mathworks.com/matlabcentral/answers/277301-how-to-build-fruit-recognition-system-using-matlab-please-help-me
- http://www.academia.edu/download/34985080/09e415093b2da4eb22000000_1.pdf
- http://www.mathworks.com/discovery/edge-detection.html