A Modified Canny Edge Detection Algorithm for Fruit Detection & Classification

Md Khurram Monir Rabby,[1,*] Brinta Chowdhury,[1,2,†] and Jung H. Kim[2,‡]

[1]Department of Electrical & Electronic Engineering, Bangladesh University of Engineering & Technology (BUET), Bangladesh
[2]Department of Electrical and Computer Engineering, North Carolina A&T State University, USA

[*]khurram.rabby@gmail.com, [†]bchowdhury@aggies.ncat.edu, [‡]kim@ncat.edu

Abstract— **This study presents an image processing procedure to identify two different classes and types of fruits. The proposed method recognizes fruits by extracting two features (color and shape) based upon the training dataset analysis. In this study, an image processing method has been done using Canny Edge Detection (CED) algorithm to identify and sort the fruits. In addition to that modified Canny Edge Detection (MCED) algorithm is proposed to develop a fruit recognition method using color and shape of the fruits. In this work, only two different types of fruits (i.e. apples and oranges) are chosen for the experiment. At the end of this study, a comparative study has been shown to evaluate the performance of CED algorithm and MCED algorithm based on the training dataset.**

Index Terms— **Edge Detection, Canny Edge Detection (CED) Algorithm and Modified Canny Edge Detection (MCED) Algorithm.**

I. Introduction

The utility of image processing in the industry has expanded considerably in recent years due to its application in different fields like precision agriculture, automated guiding, crop monitoring quality control and sorting etc. The reasons behind its wide range of application are its capability to identify without direct contact of the products. In this regard, sorting different products depends on the size, shape, color, texture etc. which can be easily detected by the image processing. But the human error is relatively high in manual sorting process which has a long-term effect on the process industry. Therefore, automated sorting like the image processing in the food industry is the most demanding issue [1]. In this respect, research is continuously going on the edge detection procedure because of its huge applications in the image morphing, pattern recognition, image segmentation and image extraction in the image processing algorithms. It represents the contour of the image by connecting the outline of the object [2], [3]. In this perspective, the edge detection is considered an important property for good orientation of an object [4]. However, the required memory is less compared to the whole image even though it contains shape and orientation related information of the object. Having significance in edge detection for the image processing, many edge detection related algorithms are suggested by the researchers based on Gradient and Laplacian derivative. Considering the fact, the edge detection algorithms based upon the traditional approach are calculated by detecting either the maximum value of the first derivative or zero crossing of the second derivative. Even though the first order differential operators such as Roberts operator, Prewitt operator, Sobel operator, etc. and second-order differential operators such as Laplace operator, LOG operator, etc. have many advantages in a simple computation, and fast but easy implementation, they are more sensitive to the noise and their detection are not perfect from the engineering point of view [2],

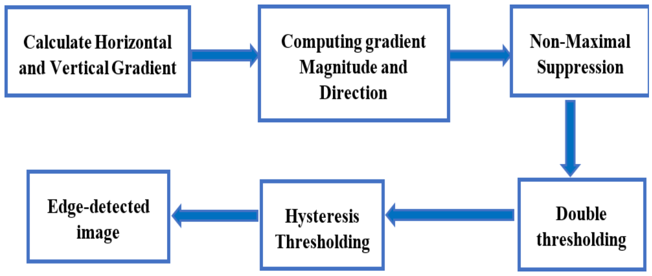

Fig. 1: Block Procedure of CED Algorithm

[4]–[6]. However, Canny Edge Detection (CED) is a standard edge detection algorithm for many years getting information about the outline of the object and local maxima and minima of the gradient of the intensity function. For this reason, the edge detection gains popularity not only in the computer vision but also in the automated industry [5], [6].

The automated sorting mostly works on a few basic features: color, intensity, shape, size, texture etc. The most common techniques used in a fruit recognition and classification in two-dimensional fruit images are either color based or shape based. Therefore, to increase the accuracy, features such as color and size have been combined. In this paper, CED and MCED algorithms are used to develop the fruit recognition method using the color and shape of the fruits. For this reason, two different types of fruit (Apple and Orange) images are processed for better edge detection. After extracting the features from the edge detected image, it has been trained from the labeled data to apply in an unknown system to detect the two different fruit types including their class also.

The rest of this paper is presented as follows. Section II introduces briefly a traditional CED algorithm. Section III of this paper describes the MCED algorithm. Proposed procedure of the image processing for a fruit detection has been analyzed in section IV. Simulation results with an analysis have been done in section V. Finally, a summarizing comment has been given in section VI.

II. Traditional Canny Edge Detection

Canny Edge Detection is known as the best edge detection algorithm because of maintaining low error rate, preserving useful information by filtering out the unwanted ones and keeping lower variation from the original image and removing multiple responses to close edge. Upon this criterion, Canny Edge Detector works as follows [5].

Step 1: Compute the horizontal (G_x) and vertical (G_y) gradient of each pixel in an image.
At each point convolve with 3×3 gradient template G_x and G_y respectively:

978-1-5386-7482-6/18/$31.00©2018 IEEE

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (1)$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2)$$

Step 2: Using the above information, magnitude and direction are computed as

$$M[u, \mathrm{v}] = \sqrt{P[u, \mathrm{v}]^2 + Q[u, \mathrm{v}]^2} \quad (3)$$

$$\theta[u, v] = \tan^{-1}(Q[u, v], P[u, v]) \quad (4)$$

Step 3: Candidate edge points are calculated by choosing the non-maximal value suppression method by

- Keeping the large gradient thin edges. Because it does not always at the location of an edge and there are many thick edges,
- Connecting thin broad ridges that are the only one pixel wide,
- Finding a local maximum by suppressing all values along the line of the gradient that are not peak values of the ridge,
- Reducing the angle of the gradient to one of 4 (four) sectors [0, 45, 90, 135],
- Checking 3×3 region of each magnitude M[u,v],
- Setting to 0 (zero) if the value at the center is not greater than the 2 (two) values along the gradient,
- Suppressing the magnitude that will contain many false edges caused by noise or fine texture.

Step 4: The candidate edge points have been calculated by double thresholding. It reduces the number of false edges by applying all values below the threshold, T is changed to 0 (zero).

To remove the limitation of false edge and edge softening problem, double thresholding is applied in the suppressed image in the following procedure.

Technique 1: Double Thresholding for Edge Detection

1. Upper Threshold = T_1 and Lower Threshold = T_2
2. Edge > T_1: Strong Edge Point
3. T_2 < Edge < T_1: Weak Edge Point
4. Edge < T_2: Delete

Step 5: Thresholding has been employed to establish a connection between the weak edges and the strong edges. Since zero padding is not included in this method to remove the false non-dominant edge, CED algorithm accuracy is not high in size related issue. Therefore, fruit size is often miscalculated which often leads to the wrong detection

III. Proposed Modified Canny Edge Detection

In this research work, color and size are used as fruit detection feature. To get the accurate size, the image has been segmented properly. In traditional CED, the presence of a pseudo edge causes an error in the determination of fruits' size. Therefore, the traditional algorithm has been modified by doing some additional works keeping rest of the algorithm is the same as follows.

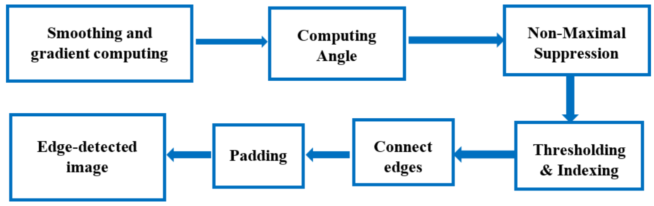

Fig. 2: Block Procedure of Proposed MCED Algorithm.

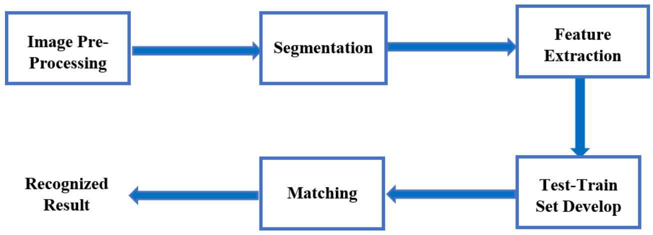

Fig. 3: Block Diagram of Image Processing.

Step 1: Input image is smoothing by computing the Gaussian kernel for horizontal (G_x) and vertical (G_y) gradient of each pixel. Here, the absolute value of the Gradient is calculated instead of Euclidean.

Step 2: Using the above information, the angular direction has been calculated.

Step 3: Candidate edge points have been calculated by choosing a non-maximal suppression method.

Step 4: The upper threshold is considered 1.562 times higher than the lower.

Step 5: Strong edges have been related to weak edges after grading.

Step 6: Zero padding is done to remove unwanted edges by keeping the unchanged strong edge.

IV. Proposed Fruit Detection & Classification Procedure

Since different fruits may not have identical feature, the classification more than two features at the same time decreases the efficiency and accuracy of the algorithm. In this research analysis, the following steps have been followed for image processing.

A. Pre-processing

Raw images are not usually suitable for object detection and classification because of the noisy environmental conditions, poor image resolution, and unwanted background etc. Therefore, the following steps of pre-processing are used:

1) Resize Image

All the images are resized into the same size to bring the uniformity in the selection process. Therefore, testing and training data set can be compared in the same field. In this work, bi-cubic interpolation for resizing, and an odd number (i.e 251 x 251) for getting a single center have been used for the post-processing.

Fig. 4: Pre-Processing Basic Block Diagram.

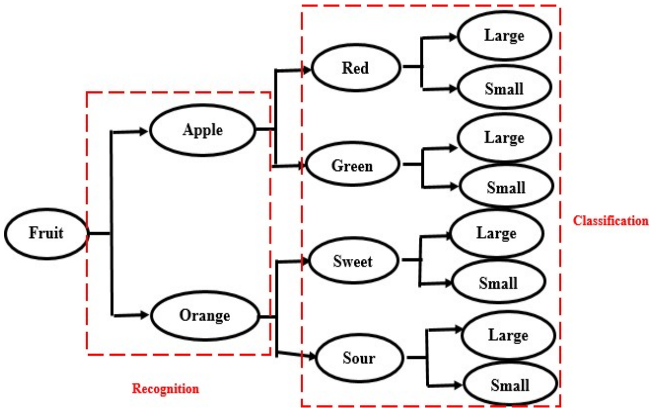

Fig. 5: Block Diagram of Recognition and Classification of Fruits.

2) Gray Image

To convert RGB image into grayscale, luminance is kept intact while eliminating the hue and saturation information.

B. Segmentation

The purpose of segmentation is to divide the image into meaningful regions. Since edge detection has been accepted as a fundamental step in image processing, it is necessary to find out true edges to get the best results from the matching process.

C. Proposed Feature Extraction

Feature extraction means to group the input data from the specification of the object. The extracted features have provided the relevant information to perform the feature matching with input data. Here, color intensity and size have been chosen as features.

1) Colour Intensity Calculation

The first step of color intensity calculation is to separate the red, green and blue components of the resized color image. After getting the values, mean of these intensity values is computed. Since selected fruits are related to red and green intensity, these two colors are normalized with respect to the summation of all three intensity values.

For example, if m_1, m_2. and m_3 are the mean values of red, green, blue intensity respectively, normalized red, green and blue intensity value are calculated as follows.

$$r_n = \frac{m_1}{m_1 + m_2 + m_3} \tag{5}$$

$$g_n = \frac{m_2}{m_1 + m_2 + m_3} \tag{6}$$

$$b_n = \frac{m_3}{m_1 + m_2 + m_3} \tag{7}$$

2) Fruit Size Calculation

Radius has been considered as a measurement of fruit size. First, the center of the edge detected image is determined. Using the Euclidean distance method average radius is calculated from the center to the edge point. If (x_i, y_i) is the edge point and (x_c, y_c) is the center of the image, the calculated average radius is:

$$D_{avg} = \frac{\sum \sqrt{(x_i - x_c)(x_i - x_c) + (y_i - y_c)(y_i - y_c)}}{\text{number of total radius drawn}} \tag{8}$$

D. Test Train Set Develop

A training set has been developed for matching after feature extraction. Different types of fruits (Orange and Apple) have been collected and intensity value and average distance from the center have been observed. Based on this observation, a threshold has been set for both color intensity, their difference and average radius.

E. Matching

Matching parts have been divided into two sections. First, identify which fruit it is and then classify them in different types of that fruit. Since orange and apple have been taken as an example, first, it has been identified whether it is orange or apple. After that, the classification has been done as follows.

Algorithm 1: Fruit Recognition

```
1.  If gn > rn
2.     If gn < Th_green intensity
3.          Detects Orange Class
4.     Else
5.          Detects Apple Class
6.  Else
7.     If Th_lower radius ⩽ Davg ⩽ Th_upper radius &&
            Th_lower intensity diff ⩽ (rn - gn) ⩽ Th_upper intensity diff
8.          Detects Orange Class
9.     Else
10.         Detects Apple Class
```

Here, g_n, r_n and D_{avg} are green and red intensity respectively and average radius for the selected fruit. The threshold for intensity difference and size have been computed from training set information. For differentiating between orange and apple, the green intensity has been calculated before the other one to make the computation easier.

Therefore, following algorithm for apple and orange classification has been applied.

Algorithm 2: Fruit Classification

```
1.  If gn > rn
2.     If Davg > Th_Apple Radius
3.        Detects Green Large Apple
4.     Else
5.        Detects Green Small Apple
6.  Else
7.     If Davg > Th_Apple Radius
8.        Detects Red Large Apple
9.     Else
10.       Detects Red Small Apple
11. If Davg > Th_Orange Radius
12.    If rn > Th_Orange Red Intensity
13.       Detects Large Sweet Orange
14.    Else
15.       Detects Large Sour Orange
16. Else
17.    If Davg > Th_Apple Radius
18.       Detects Small Sweet Orange
19.    Else
20. Detects Small Sweet Orange Apple
```

For classification of apple in the combination of green intensity and average radius, preference has been given in intensity due to its high value. On the contrary, in orange classification, average radius and red intensity combination are chosen.

V. Simulation & Result Analysis

Different images of orange and apple have been collected and simulated for image processing by the following method described in section IV. For simulation purpose, MATLAB has been used as a tool to detect and classify fruits as per their types.

A. Comparative Test Results

In Figure 6, large apple has been chosen to detect its type and class. It shows that both algorithms can successfully detect but the modified edge is clearer and more accurate than the traditional one. In Figure 7, for large sweet orange, both algorithms have successfully detected with variation in their edges. However, the modified edge has less detected pixels inside the boundary than the traditional one. In Figure 8, red small apple has been selected to detect its type and class. Although both algorithms can successfully detect, noisy image and Gaussian operator cause dominant pixels visible inside the boundary for both cases. In Figure 9, for red large apple, it has been showed that the MCED algorithm can detect it successfully. However, traditional CED algorithm cannot identify due to in-appropriate edge detection.

B. Performance Evaluation

After applying the apple and orange test train information, we have summarized the comparative performance of Canny and proposed modified Canny Edge detection algorithm in Table I and Table II respectively.

TABLE I
CED Algorithm Performance Evaluation

Fruit	Colour	Detection/Recognition	Classification	
			Small	Large
Apple	Red	11/13	4/5	2/2
	Green		1/2	4/4
Orange	Sweet	6/7	3/4	1/1
	Sour		2/2	N/A

TABLE III
MCED Algorithm Performance Evaluation

Fruit	Colour	Detection/Recognition	Classification	
			Small	Large
Apple	Red	13/13	5/5	2/2
	Green		2/2	4/4
Orange	Sweet	7/7	4/4	1/1
	Sour		2/2	N/A

From the above table information, it has been seen that fruit recognition and classification rate of modified one is better than the traditional approach. Traditional approach cannot detect edge successfully in presence of more noise and dark background. Therefore, it is difficult to set threshold data from training set information.

VI. Conclusions and Future Works

In this study, identification along with the classification of two different fruits has been performed using the image processing technique. Moreover, MATLAB has been used as a simulation tool to perform the experiment with accuracy. Following the proposed technical procedure for image processing, objects have been detected and classified using their color intensity and average diameter. After selecting threshold data from the trained information set, the procedure has been applied in a random set. Based on fruit color and size, it has been observed that proposed MCED algorithm's detection rate is quite higher than the traditional approach. In comparison to Neural Network and other computer vision approach, the proposed approach is computationally easy to implement within limited training data set. Though this research includes only orange and apple, it can be extended to different fruits and vegetables using the same technique.

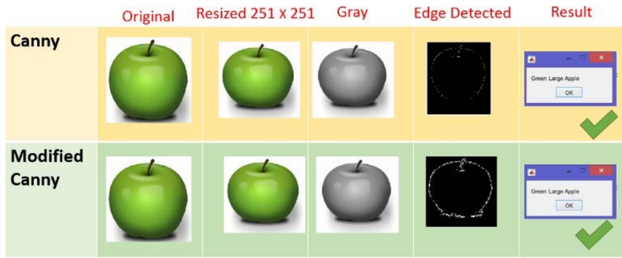

Fig. 6: Fruit Detection and Classification Type Test 1.

Fig. 7: Fruit Detection and Classification Type Test 2.

Fig. 8: Fruit Detection and Classification Type Test 3.

Fig. 9: Fruit Detection and Classification Type Test 4.

References

[1] R. R. Mhaski, P. B. Chopade, and M. P. Dale, "Determination of ripeness and grading of tomato using image analysis on raspberry pi," in *2015 Communication, Control and Intelligent Systems (CCIS)*, pp. 214–220, Nov 2015.

[2] R. C. Gonzalez and R. E. Woods, "Image processing," *Digital image processing*, vol. 2, 2007.

[3] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 679–698, Nov 1986.

[4] B. Kaur and A. Garg, "Mathematical morphological edge detection for remote sensing images," in *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, vol. 5, pp. 324–327, IEEE, 2011.

[5] A. L. Kabade and D. V. Sangam, "Canny edge detection algorithm," *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)*, vol. 5, no. 5, pp. 1292–1295, 2016.

[6] W. Rong, Z. Li, W. Zhang, and L. Sun, "An improved canny edge detection algorithm," in *Mechatronics and Automation (ICMA), 2014 IEEE International Conference on*, pp. 577–582, IEEE, 2014.