e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:02/February-2022 Impact Factor- 6.752 www.irjmets.com

INTERFACE DEVELOPMENT FOR REAL TIME IMAGE EDGE DETECTION

Murlidhar Manjhi*1, Vinod K Patel*2, S K Pandey*3, Deepak Mishra*4
*1,2Department Of Computer Science, VNS Group Of Institution- Faculty Of Engineering, Bhopal, India.
*3,4Professor, Department Of Computer Science, VNS Group Of Institution-
Faculty Of Engineering, Bhopal, India.

ABSTRACT

Edge detection is one of the frequently used operation in image processing. Edge detection is a pre- processing step for improving the quality of the edges in noise contaminated image. Edge detection technique can preserve the edge details by removing unwanted information or noise from the image. In today's life edge detection becomes important in various fields such as in medical field, traffic control system in defense applications etc. Various edge detection techniques are available for this purpose. Deep Learning where neural network comes into picture also needs edges of input data to detect objects. Edge detection techniques can reduce the complexity of image processing by removing unusual data from the images. We have developed a user interface where user or researcher can easily find edges of any image including real time captured frames just by selecting available edge detection techniques. Apart from GUI, we have implemented edge detection techniques and also compared these techniques together. From the comparison we observed that the Canny edge detection technique is better than other technique as it can provide good results in noisy images as well.

Keywords: Image Processing, Edge Detection, Canny Edge Detection, Real Time, GUI.

I. INTRODUCTION

Edge detection is an important problem in image processing. Edge detection could be a fundamental technique to simplify high-level image processing work because the edges of an image contain important information about the images and the edges can also help to find the shape and size of the object in the image. Detecting the edges of the image separates the image into the object and the background [2] so it is easy to quickly find out the hidden features from the image. The edges of the image are widely used in feature extraction [4, 7], pattern recognition, visual perception (object recognition), image enhancement, image segmentation, etc. Edge detection techniques are useful in various different fields for this purpose. for example,

- Traffic monitoring
- Medical science
- Satellite imaging
- Fingerprint recognition etc.

The most important part of this work is to detect real time edges [1] of any images or frames of a video. Another important part of this work is to provide user interface for various edge detection techniques [3]. One can directly get the possible edges just by selecting any one technique. Here one possible question may arise- why edges of images are required ? So, it requires to enhance the quality of the image or extract useful information and features from the images. AI based model uses edge detection to identify and differentiate objects. This system is based on image processing and it is used as a dependent module, as it combines with other project and performs the major tasks, in different fields. If we talk about technical part of this work then, this system is implemented or simulated in Python using OpenCV, Matplotlib, Numpy and its performance is tested on real time videos and images or on stored images. In other word, some standard dataset as well as self-generated dataset has been tested. It is observed from the experiment that the developed application is perfectly detecting the edges of the image using different edge detection techniques. The literature includes several edge detection techniques such as Canny, Sobel, Roberts, Prewitt etc [5, 6]. Which proposes some operators to detect edges in the image. And these operators detects the edges in desired direction. During edge detection we faces many intercepts but noise is the major problem in the procedure of edge detection. And this problem was then overcame by Canny operator. Basic flow of any work is the essential part of particular project. In Figure 1, it can be understand that our model is reading colour image and then converting it into grey scale image. After getting grey scale image, edge detection technique play important role to generate all possible edges. It has been demonstrated in below section with example.

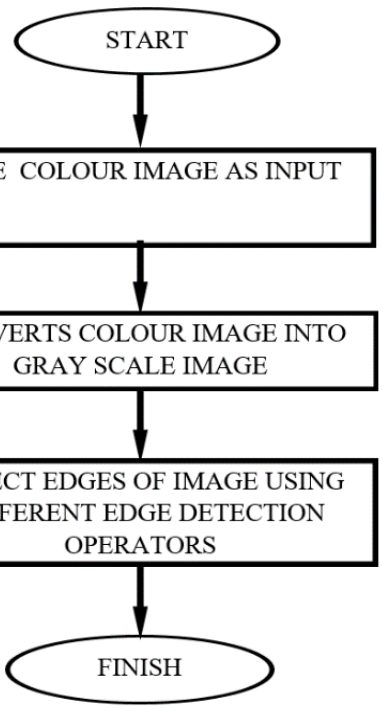
START
TAKE COLOUR IMAGE AS INPUT
CONVERTS COLOUR IMAGE INTO GRAY SCALE IMAGE
DETECT EDGES OF IMAGE USING DIFFERENT EDGE DETECTION OPERATORS
FINISH

Figure 1: Basic Flow Chart For Edge Detection.

II. LITERATURE REVIEW

EDGES

Edges are nothing but the sudden change of intensity in the image. It is a important feature of image. An edge may defined as a set of connected pixels that forms a boundary between two disjoint region. Most of the shape information of an image is enclosed in edges. And only edge detection make us able to understand the shape of the object in the image.

EDGE DETECTION

Edge detection is the fundamental tool in image processing or in computer vision, particularly in the field of feature detection and in feature extraction. Edge detection is the very first step in extracting information from images. Edge detection removes the unnecessary information from the image while finding the structure of the image and identifying points in a digital image at which the image brightness changes sharply.

Steps in edge detection

Edge detection is a technique in image processing, which is used for detecting boundaries [9, 10] of an object inside the image. It is mainly used for following purpose like image segmentation and for extracting features from the images in different areas such as in image processing, computer vision, or in machine vision etc.

Following are the main step in edge detection:-

Smoothing :- Smoothing is a technique in image processing which is used to reduce random variations of brightness and color in the image without changing the original form of the image.

Enhancement :- Enhancement is a basic techniques which is used for improving the quality of the image basically it sharpen the edges of the image.

Thresholding :- Basically thresholding decides which pixel of the edge should be discarded as a noise or which to kept. Thresholding means taking a fix value and separating the dark and light colored pixel of the edge.

Localization :- Localization is a technique which determine the exact location of the pixel of the edge or the most visible pixel of edge of object in the image.

III. METHODOLOGY

Convert RGB image into Gray scale image:

In every edge detection techniques, the first step is to convert RGB image into gray Scale image. This is because the RGB image can be viewed as three dimensional image basically a M*N*3 array of color pixel, where each pixel is triplet and corresponds to Red, Green, Blue. This increases the complexity so in order to minimize this complexity RGB images are converted into Gray scale image. As Gray scale images are one layered image that needs less information for each pixel.

Sobel Edge Detection Operator:

The Sobel Operator was developed by **Irwin Sobel** and **Gary Feldman**. Sobel filter is used in image processing or in computer vision specifically detect the edges of an image. Sobel operator [8] works by calculating the gradient (i.e. intensity or the colour change in image) of each pixel in image in X (horizontal) and Y (vertical) direction. Sobel operator is implemented as a sum of two directional edges.`

Sobel Operator uses two 3×3 Kernels. One kernel is used to calculate changes in horizontal direction, and the other kernel is used to calculate changes in vertical direction. These two kernels are convolved with original image(i.e. input image) in order to find the approximation of gradients. The gradient of each pixel is calculated in both the direction in image using:

-1	0	1
-2	0	2
-1	0	1

Gx

1	2	1
0	0	0
-1	-2	-1

Gy

Figure 2: Horizontal & Vertical Masks Of Sobel Operator.

The gradient of that pixel in each pixel of the image is estimated by a combination of the above results:

$$G = \sqrt{G_x^2 + G_y^2}$$

Sometimes the below give equation is also used for calculation:

$$G = |G_x| + |G_y|$$

Canny Edge Detection Operator

The Canny operator was introduced by John F. Canny(1986). Canny operator is probably the most commonly used operator for detecting an edge in an image. The canny operator can be broken into five different phases.

Phase 1:Smoothing :- Smoothing is the first step to detect edges of an image using a Canny operator. During this phase the noise from the original image is removed by adjusting the contrast and brightness of the image. This step can blur the original image so noise is removed from the image and a 5×5 Gaussian filter is used to blur the image.

Mathematically, the smooth resulting mage is given by,

$$f(i,j) = G*I(i,j)$$

Phase 2: Calculate the gradient :- Gradient is a directional change in the intensity of the image. Gradient is a two dimensional vector direction and the magnitude. Direction defines the direction of largest possible intensity increase for each pixel and where as magnitude defines the highest strength of the edge.

Phase 3:Non-maximum suppression :-This step converts thick edges into thin and sharp edges and these edges are further used for object recognition. Even if we have a image with multiple objects, non-maximum suppression is able to select one entity or a object from among several small overlapping objects.

Phase 4: Double Threshold:- The Canny operator considers two threshold values(high threshold and low threshold). The threshold is marked as high threshold if the pixels of edge is strong and the threshold marked as low threshold if the pixels of edge is weak and if the pixels are between the threshold values than its threshold is depends on its neighbouring pixel.

Phase 5: Edge detection through hysteresis:- In this step the strong edges are included in the final output and the weak edges are omitted. But if a weak edge is associated with a strong edge, it should included in the final output.

IV. RESULTS AND DISCUSSION

RGB image into Gray scale image:

In every edge detection technique, the first step is to convert RGB image into gray Scale image. This is because the RGB image can be viewed as three dimensional image basically a M*N*3 array of color pixel, where each pixel is triplet and corresponds to Red, Green, Blue. This increases the complexity so in order to minimize this complexity RGB images are converted into Gray scale image. As Gray scale images are one layered image that needs less information for each pixel. And finally edges can be detected using available methods. Figure 3, tells the process of getting grey scale image of RGB image and then possible all edges can be visualize in beside of grey image. Implementation steps are written below- as cvtColor( ) function is converting a color image into gray by passing argument ‘BGR2GRAY’. After that Gaussian filter has been applied to each pixel in the gray image and then Canny edge detection technique finds all possible edges.

```
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
img_gaussian = cv2.GaussianBlur(gray,(3,3),0)
img_canny = cv2.Canny(img,100,200)
```

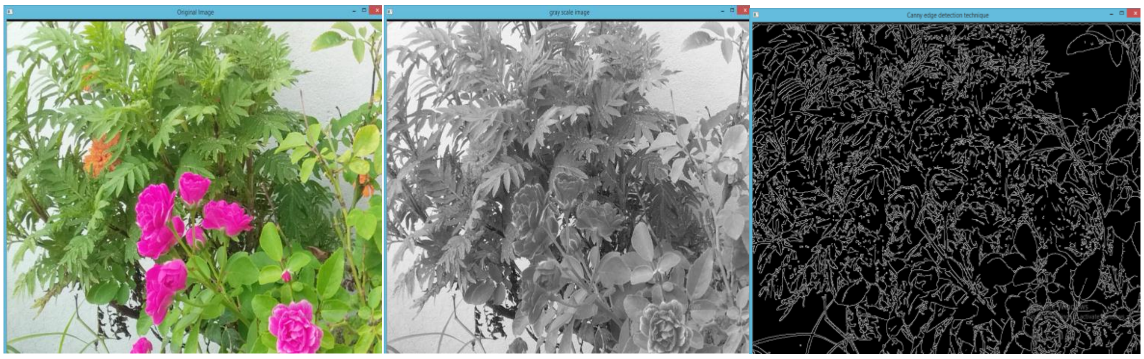

Figure 3: RGB Image -> Grey Image -> Edge Detected.

Live Edge Detection:

Live edge of frames or image can be also detected in real time. In general, a video has 30 FPS, and all the edges of all frames can be evaluated in real time using any edge detection approach. Steps for extracting frames from the video and finding its edges has been illustrated with following lines of code-

```
Cap = cv2.VideoCapture(0)
while (1):
  _, framei = cap.read()
  hsv = cv2.cvtColor(framei, cv2.COLOR_BGR2HSV)
  sobelx = cv2.Sobel(framei,cv2.CV_64F,1,0,ksize=5)
  sobely = cv2.Sobel(framei,cv2.CV_64F,0,1,ksize=5)
  laplacian = cv2.Laplacian(framei,cv2.CV_64F)
  gray = cv2.cvtColor(framei, cv2.COLOR_BGR2GRAY)
  edges = cv2.Canny(gray, 50, 50)
```

User interface:

GUI for edge detection can be visualize in Figure 4. All possible edge detection techniques are available on same window. Any user or students or researcher can find edges of image in single click by opting any of methods. All the techniques can be seen in left side of window and edges of hotel Taj has been detected by Canny edge detection technique. One can use any of techniques.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:02/February-2022 Impact Factor- 6.752 www.irjmets.com

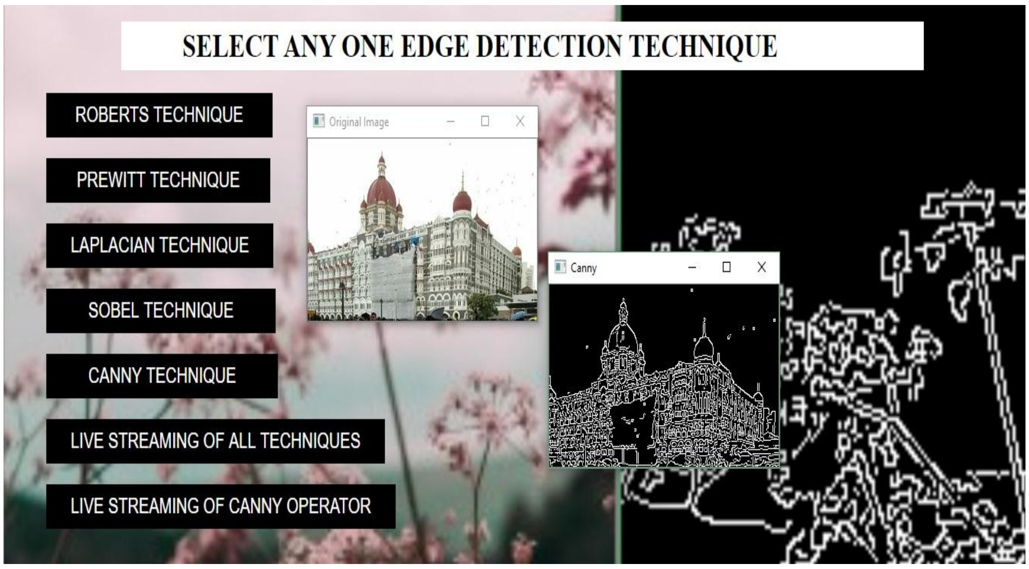
SELECT ANY ONE EDGE DETECTION TECHNIQUE
ROBERTS TECHNIQUE
PREWITT TECHNIQUE
LAPLACIAN TECHNIQUE
SOBEL TECHNIQUE
CANNY TECHNIQUE
LIVE STREAMING OF ALL TECHNIQUES
LIVE STREAMING OF CANNY OPERATOR
Original Image
Canny

Figure 4: Graphical User Interface for Real Edge Detection.

V. CONCLUSION

All together We implemented several edge detection techniques (Robert's, Prewitt, Sobel, Laplacian of Gaussian and Canny) in this work, which are commonly used for edge detection technique. We applied these techniques on three different images such as flower, public place or on temple images and benchmark datasets. And from analyzing the output produced by these edge detection techniques it is observed that Laplacian of Gaussian and Canny edge detection technique gives good results as comparison to other edge detection techniques. But Laplacian of Gaussian is permeable to noise. So it does not provide better results than Canny edge detection technique and from the experimental result also proved that Canny edge detection technique is better than any other edge detection technique. Laplacian of Gaussian comes on second number as if we apply LOG edge detection technique on little noisy it can get much better results. Sobel edge detection technique comes on third place as it can give better results than Prewitt and Robert's edge detection technique. But Sobel operator is not able to detect continuous and thin edges of image. Prewitt edge detection comes on fourth number, because it can only detects that edges of object in image which have high gradient value. And at last comes Robert operator, it can not be able to detect more edges because Robert operator uses 2*2 Matrix. It is good for binary image. It is observed that the performance of edge detection technique is depends on the input image. Like if we apply these techniques on less noisy image it provides good results but if we provide noisy image the performance of these edge detection techniques is decreases. And at last not the least we developed GUI for edge detection by which any one can select edge detection techniques by choice and get real time edges of frames. The edge detection techniques can be applied in the area of surveillance system face recognition, parking system, feature extraction etc. There are still some things we can do in future like we will improve the stability of program and modify the code of function to let program more stable and it's a chance to get the better adaptive method for image edge detection or image segmentation.

VI. REFERENCES

[1] Yaman, Sertaç, Barış Karakaya, and Yavuz Erol. "Real time edge detection via IP-core based sobel filter on FPGA." 2019 International Conference on Applied Automation and Industrial Diagnostics (ICAAID). Vol. 1. IEEE, 2019.

[2] Liu, Yang, Zongwu Xie, and Hong Liu. "An adaptive and robust edge detection method based on edge proportion statistics." IEEE Transactions on Image Processing 29 (2020): 5206-5215.

[3] Menaka, R., Ramadoss Janarthanan, and K. Deeba. "FPGA implementation of low power and high speed image edge detection algorithm." Microprocessors and Microsystems 75 (2020): 103053.

[4] Ranjan Rajnish K., and Anupam Agrawal. "Video summary based on F-sift, Tamura textural and middle level semantic feature." Procedia Computer Science 89 (2016): 870-876.

[5] Marr, David, and Ellen Hildreth. "Theory of edge detection." Proceedings of the Royal Society of

London. Series B. Biological Sciences 207.1167 (1980): 187-217.

[6] Chaple, Girish N., R. D. Daruwala, and Manoj S. Gofane. "Comparisions of Robert, Prewitt, Sobel operator based edge detection methods for real time uses on FPGA." 2015 International Conference on Technologies for Sustainable Development (ICTSD). IEEE, 2015.Lk

[7] Ranjan, Rajnish K., Yachana Bhawsar, and Amrita Aman. "Video Summary Based on Visual and Mid-level Semantic Features." International Conference on Communication, Networks and Computing. Springer, Singapore, 2020.

[8] Canny, John. "A computational approach to edge detection." IEEE Transactions on pattern analysis and machine intelligence 6 (1986): 679-698.

[9] Maini, Raman, and Himanshu Aggarwal. "Study and comparison of various image edge detection techniques." International journal of image processing (IJIP) 3.1 (2009): 1-11.

[10] Ganesan, P., and G. Sajiv. "A comprehensive study of edge detection for image processing applications." 2017 international conference on innovations in information, embedded and communication systems (ICIIECS). IEEE, 2017.