International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V71I2P208
Volume 71 Issue 2, 61-70, February 2023

Original Article

Comparative Analysis of Edge Detection Methods using Deep Learning

Dipmala Salunke[1], Bhushan Dhamankar[2], Sairaj Chidrawar[3], Rohit Kangule[4], Shrikrushnakumar Sondge[5], Sumit Deshmukh[6]

[1,2,3,4,5,6]*Department of Information Technology, JSPM's Rajarshi Shahu College of Engineering, Pune, India*

[1]*Corresponding Author: dipmala.salunke@gmail.com*

Received: 28 November 2022 Revised: 04 February 2023 Accepted: 14 February 2023 Published: 25 February 2023

Abstract *- Computer vision is a subset of artificial intelligence (AI), which is used to extract meaningful data from images. It provides different features such as object detection, edge detection, image classification etc. Edge detection is very useful in industries like the civil industry, agriculture industry, autonomous vehicles, facial recognition, manufacturing, etc. Using opencv, we can use different edge detection operators to get object edges and detect objects. The main problem with dimension detection is the edges. Edges are one of the important characteristics of the image, which can provide us with very useful information about the object. Though edge detection is a very old topic, there is still no solid study to explain which edge detection method will work best for dimension detection. So here is a comparative analysis to find which edge detection algorithm performs best for dimension detection to locate excellent edges that will generate decent contours. All edge detection systems' effectiveness needs to be evaluated. The edges of the image may be extracted using a variety of edge detection algorithms, and the performance can be judged using metrics like signal-to-noise ratio (SNR), structural similarity index measure (SSIM), entropy, peak signal-to-noise ratio (PSNR), mean squared error (MSE). In this paper, in addition to first derivative operators like sobel, robert, and prewitt, gaussian-based algorithms, the laplacian of gaussian, and the canny edge detector have also been taken into consideration. It is experimentally observed that the sobel operator is performing better than others, with an average SNR value of 1.1730.*

Keywords *- Opencv, Edge detection, Dimension detection, Image segmentation, Prewitt, Sobel, Laplacian of gaussian, Canny, Robert.*

1. Introduction

One of the most effective techniques in the business for finding an object's dimensions in real-time is called "dimension detection" so that the industrial resource management cost can be reduced. This paper provides the necessary review of edge detection operators so that users can find the better-performing one to get object edges. This is a crucial subject in computer vision issues. As previously mentioned, this paper also demonstrates a method for instantly calculating measures from images. For detecting the dimensions of the object system must use an edge detector that would yield uniform and fully connected contours. However, there is a lack of research on which edge detector algorithm will perform best for contour detection. Other research and comparative studies show that the sobel operator performs better based on the number of edges detected when it comes to edge detection but does not perform well for contour detection. So here is a comparative analysis to find which edge detection algorithm performs best for dimension detection to locate excellent edges that will generate decent contours. In order to detect the item, it essentially employs a webcam. After finding the thing, it applies an edge detector operator to display its measurements in the given units in real-time. Utilising this approach has several benefits, including the fact that it is highly beneficial in the industrial sphere and simplifies human work. To determine each item's size, one must first determine the reference object. It is the Aruco marker in this instance. Following that, the dimensions of the items in relation to the reference are measured or computed, leading to the display of the object's size[5].

An image consists of many characteristics like the object's size, colour, orientation and contours. For dimension detection, the main aspect is the object's shape, which can be given by contours, but to find contours, the object's edges must be detected [24]. This study and analysis of various edge detection algorithms makes recommendations for the most effective one so that the edge detector can determine the image dimensions.

2. Literature Review

Dimension detection is a key concept when it comes to making the right or wrong understanding of whether parts or products have been worked or assembled according to desired specifications. The main practice of dimension detection is in civil corporations. Real-time dimension detection deals with image processing, enabling us to detect objects' dimensions in real-time.

F. Chen et al. offer a more effective technique for determining the local fractal dimensions. In place of the missing data, they have added 0 as an alternate value. When certain border pixel neighbours in the local windows are not available, then replayed a few values available. The main drawback of this method is that it cannot detect edges for the homogeneous textured surface[1].

Israni S. et al. worked on edge detection of license plates. They have used a sobel operator for edge detection and to locate picture edges, employing derivative approximation. The sobel operator offers the advantage of flattening the image's random noise. The drawback here is that the authors are only focussing on signals and not considering the noise associated with the operators [2].

Hanmin yeet et al. presented an optimised picture edge detection method. To increase the algorithm's performance against noise, they integrate bm3d and prewitt operator de-noising. The prewitt operator has multiple applications because of its simple calculation and the drawback being poor anti-noise performance [3].

Achal sharma et al. have done the analysis of sobel edge detection for face recognition, where the edges from digital images are used to identify faces. The Sobel edge detection method is based on the horizontal and vertical convolution of the image. Here they are unable to detect complex edges[4].

Shweta pardeshi et al. proposed an augmented technique for finding real-time measurements of objects and detecting objects. They employed a combination of AI and IoT technologies like opencv and webcam. Webcam and white paper backgrounds were used to detect images with a canny edge detector[5].

G. Xin et al. have suggested a more accurate clever edge-detecting method. The reason behind this idea was that traditional canny edges were unable to perform on color images. Authors have proposed an algorithm containing a quaternion weighted average filter, calculation of the vector sobel gradient, and interpolation-reliant non-maxima suppression. The main drawback is that the proposed method is very computationally expensive[6]. Theodora sanida et al. have implemented the 5 x 5 sobel kernel using opencl and compared the outputs of both 3 x 3 and 5 x 5 kernels for experimental analysis[7].

L. Cao et al. have used an adjustable median filter algorithm to remove the garment image noise. They employed a quick fuzzy edge-detection technique to identify the edges of clothing photos. They also used a freeman code-based approach to locate the corner points. They successfully eliminated impulsive and non-impulsive noise from pictures using an adaptive median filter. This method is computationally expensive[8].

Y. Zhang et al. utilised the laplacian of the gaussian operator, the canny operator, and the modified sobel operator to enhance edge detection results. Noise from the picture is removed using morphological smoothness and applied techniques for image fusion[9].

Jayshree deka et al. proposed fractional order derivatives based on sobel, prewitt and laplacian operators. They used a dataset of freshwater fish images and evaluated the performance of their system with the help of MSE, PSNR, SSIM and FSIM, the four metrics used in the image quality assessment (IQA) process[10]. Yolanda ferandji et al. proposed research regarding the best operators to identify the character of the word lontara in Sanskrit manuscripts. In this research, authors have used both qualities of images, good as well as poor. MSE parameter was used to measure the operator performance, but this study is only limited to a small use case[11].

P. Prathusha et al. morphological operation and masking to suggest an improved sobel, prewitt, robert operator. The crab photos are given thick images, and the useless edges are removed. Basically, they compared enhanced operators with older ones and found that enhanced ones are much better[12]. A. Jain et al. have researched numerous edge detection techniques and various operators. They have found that bio-inspired algorithms are better than traditional algorithms[13].

Md khurram et al. have presented techniques for extracting colour and form to detect fruits. Fruits are found and sorted using canny edge detection[14]. Hongli lu et al. have proposed an algorithm which improved canny obstacle edge detection. Using this algorithm, one can get a more accurate obstacle edge which is important for removing obstacles of wall cleaning robots. However, the selection of gradient templates largely depends on prior knowledge[15].

3. Methodology

Figure 1 shows the flow of the system used for comparative analysis of edge detection algorithms for real-time object detection.

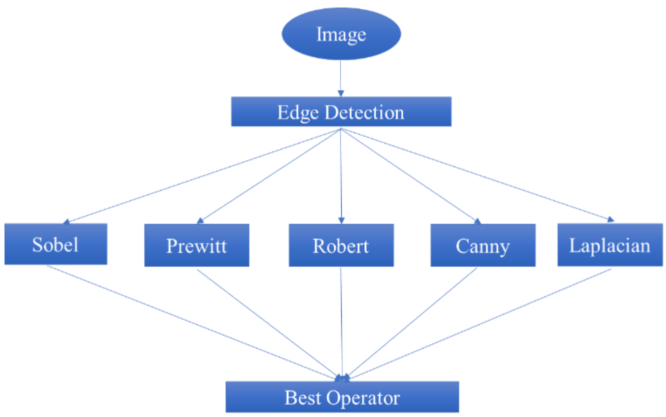


Fig. 1 Proposed Workflow

3.1. Sobel Operator

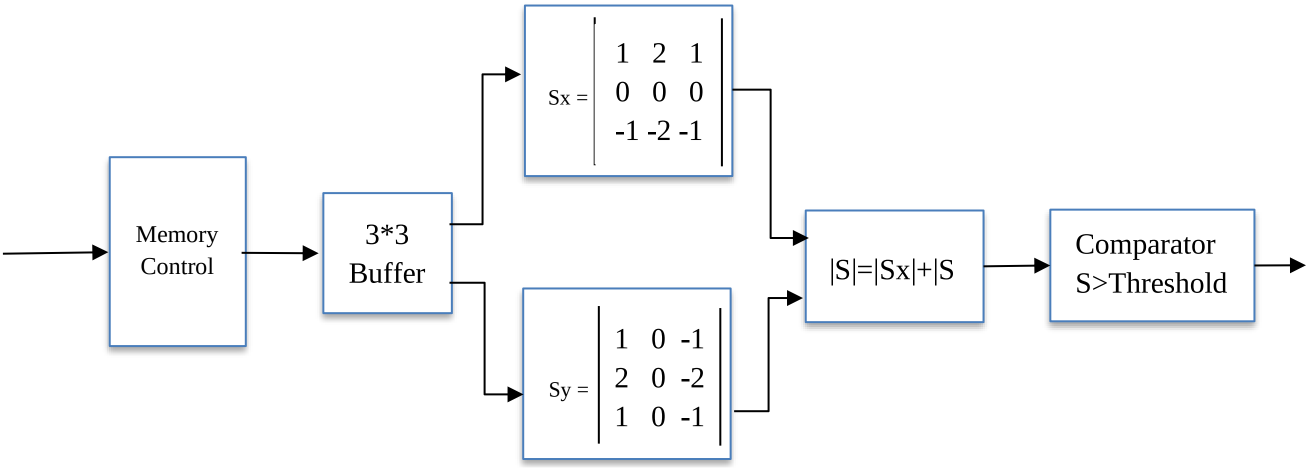


Fig. 2 Sobel Operator

Edges can be extracted by applying changes in pixel intensity. To mark the edges, we will observe the differences in the intensity of surrounding pixels[25]. Two masks make up sobel edge detection, one of them is helpful for vertical edge identification, and the second helps for identifying horizontal edge identification. Each mask works to identify and calculate the gradient effect horizontally and vertically in both directions. As with other processes, the first step is to read the image and generate the pixels accordingly, then apply filters by convolving the image [1][2][3]. After the process of masking, the image is convolved into the initial picture. Given gx and gy to the horizontal and vertical convolutions. One can specify the threshold values to be "T". Using this, calculate gradient ‘g’, which will be calculated using the formula given as {square root[(gx^2) + (gy^2)]}. So, please take a look at the first pixel, let's say m. If g is bigger than the t form, then go on to the next surrounding pixel and repeat the process[5]. The kernels which are used for sobel edge detection are:

Gx =

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

Gy =

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

- Using the vertical kernel of the types would result in edges which will be pointing towards the x-direction[5].
- Using the horizontal kernel of the types would result in edges which will be pointing towards the y-direction[5].
- Let gx and gy stand in for the x and y directional intensity gradients, respectively. If a and b denote the x and y kernels defined above: G_x = a*i and G_y = b*i … (‘*’ = convolution operator, i = input image)
- The final approximation of the gradient magnitude is calculable as

$$G = \sqrt{G_x^2 + G_y^2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

3.2. Prewitt Operator

Edge detection is achieved with this operator. Edges may be determined by calculating the difference between neighbouring pixels' brightness in a picture. It is the first derivative operator that uses the concept of derivative masking. As an image is being signalled to change in signal, that signal can be measured using differentiation. Due to which this is said to be the derivative operator or derivative mask [10][9].

All derivative masks must contain below characteristics:

- Masks should include opposite signs.
- Both mask’s collective addition should be zero.
- As weight increases, edge detection gets better.

Generally, there are two masks provided by the prewitt operator. One is used to detect edges vertically, and one is used to detect edges in the horizontal direction.[8]

3.2.1. Vertical Direction

-1	0	1
-1	0	1
-1	0	1

Because the zeros column is positioned vertically, the above mask will detect edges in that direction. This mask will give you the vertical borders of a picture when you convolve it with one.[8]

3.2.2. Horizontal Direction

-1	-1	-1
0	0	0
1	1	1

Because the zeros column is oriented horizontally, the above mask will show edges in that direction. This mask would have obvious horizontal borders when it was superimposed over an image[7].

This mask will make an image's horizontal edges shine out. It determines the difference between pixel intensity across a certain edge and operates similarly to the above mask. However, since the middle row of the mask is entirely made up of zeros, it instead calculates the difference between the pixel intensities of an edge above and below edges rather than using the edge's original values from the original picture, boosting the important shift in intensities and increasing edge exposure. Both of the mentioned masks conform to the derivative mask principle. The opposite signs are present for both masks, and the sum of the two masks is zero. Given that both of the mentioned masks are standardised and that their values cannot be changed by us, the third criterion will not apply to this operator[9].

3.3. Robert Operator

Using robert cross operations, it is easy and efficient to compute two-dimensional measurements of gradients on the picture. That is why it can be said that robert will give insights into the greater spatial frequencies which correlate with the edges. Pixel values are visible, which will show the expected values of the spatial gradients of the input at a certain point.[11] theoretically, it has the convolution kernels of two, which is 2×2 (shown in figure 1). Robert and sobel operators have quite similar work.[10]

Gx =

1	0
0	-1

Gy =

0	1
-1	0

These kernels, one for each of the two perpendicular orientations, are made to react as much as possible to edges that run at a 45° angle to the pixel grid. The kernels can be individually used to measure the gradient component in each direction in the input image (call these gx and gy). The absolute size and direction of the gradient at each site may be calculated by combining these. Given by the gradient's magnitude is :[10]

$$|G| = \sqrt{G_x^2 + G_y^2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

The estimated magnitude is calculated by:

$$|G| = |G_x + G_y| \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3)$$

This is faster for computing.

In relation to the orientation of the pixel grid, the angle at which the edge producing the spatial gradient is oriented is given by:

$$Angle(\theta) = (\frac{G_x}{G_y}) - \frac{3\pi}{4} \ldots\ldots\ldots\ldots\ldots\ldots\ldots(4)$$

In this instance, orientation 0 is understood to signify that the direction of the image's highest contrast, from black to white, travels from left to right. Subsequent angles are then measured counterclockwise from this.[9]

Canny edge operator

This algorithm is divided into the following steps [9]:

3.3.1. Grayscale Conversion

The first and most important step in using this algorithm is to convert the input image into the grayscale format. The grayscale format makes images much smoother and more realistic to work on.

3.3.2. Noise Reduction using Gaussian Filter

As edge detection is susceptible to noise in the image. So, a gaussian filter must be used to eliminate the image's noise (gaussian blur). This noise can be supposed as edges due to sudden intensity change by the edge identifier. One of the solutions to clear the noise is by using gaussian kernels of certain sizes, i.e., 3×3, 5×5, 7×7 etc. A Gaussian kernel is directly proportional to a gaussian blur, which means blur will be minimum if the gaussian kernel is the smallest. This kernel ensures that the image gets blurred and noise gets removed. The gaussian filter kernel equation is

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(5)$$

3.3.3. Gradient Calculation

This step mainly focuses on edge direction as well as intensity. These gradients are identified by using a sobel filter, and edges come into the picture when an image's colour changes, so the image's intensity also changes. Both horizontal (x) and vertical (y) observations of this intensity variation are made.

The derivatives ix and iy with respect to x and y are computed when the image is smoothed. Convoluting it with the sobel kernel's horizontal kx and vertical ky achieves this.

Sobel kernels:
$K_x = [-1\ 0\ 1\ \ -2\ 0\ 2\ \ -1\ 0\ 1\,]$,
$K_y = [1\ 2\ 1\ \ \ 0\ 0\ 0\ \ -1-2-1\,]$

After using these kernels, the magnitude g and the slope θ of the gradient are calculated.

$$|G| = \sqrt{I_x^2 + I_y^2}\ , \ldots\ldots(6)$$

$$\theta(x, y) = (\frac{I_y}{I_x}) \ldots\ldots(7)$$

3.3.4. Non-maximum Suppression

Before this step, the extracted edges were much thicker, but the final image or output image should have thin edges. The edges are thinned out using non-maximum suppression. The pixels with the highest value in the edge directions are found by iterating over all the points on the gradient intensity matrix. It finds two neighbours for each pixel in positive as well as negative gradient directions.

3.3.5. Double Thresholding

Finding three categories of pixels—strong, weak, and irrelevant—is the main goal of this stage. Here, gradient magnitudes are compared with two threshold values. The first one is lower than the second. Strong pixels have high intensity and are considered for the final edge. Weak pixels do not have that much intensity to be considered strong ones but have that much intensity to not be considered non-relevant pixels and, other than this, are all non–relevant pixels. A higher threshold is considered to identify the strong pixels. The hysteresis approach (the next step) will enable us to determine which pixels are significant and which are irrelevant for all pixels with intensities between the upper and lower thresholds.

3.3.6. Edge Tracking using Hysteresis

It works on the result produced by threshold if and only if weak pixels are converted to strong pixels, and the pixel around that weak pixel is strong. After this, the pixels that would be considered for the final edge map shall be found.

3.3.7. Cleaning Up

In this final step, the remaining frail edges will be crossed by us and set to 0, which will show the final output.

Laplacian of Gaussian

Laplacians are the derivative filters. Derivative filters are affected by the noise, so it is important to smoothen the image using gaussian filters before using laplacian. This process is known as the laplacian of gaussian. It operates on an image's second derivative. If the image is uniform in nature, the laplacian of gaussian will give 0[10]. It responds negatively to the lighter side and positively to the darker side. The result will change if there is a sharp edge between any two regions given by laplacian gaussian as –

(i) response will be zero if it is away from the edge.
(ii) response will be positive if it is just to one side of it.
(iii) response will be negative if it is just to the other side of it.

The formula provides a definition of the gaussian function:

$$G(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \ldots\ldots(8)$$

Where, σ is the standard deviation.
And the laplacian of gaussian is calculated from

$$LoG = \frac{\partial^2}{\partial_{x^2}} G(x, y) + \frac{\partial^2}{\partial_{y^2}} G(x, y) = \frac{x^2+y^2-2\sigma^2}{\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \ldots\ldots(9)$$

4. Comparative Analysis

4.1. Parameters to Compare

This analysis uses five parameters on which the edge detection algorithms are compared upon entropy, PSNR, MSE, SNR and SSIM [16][17][18][22]

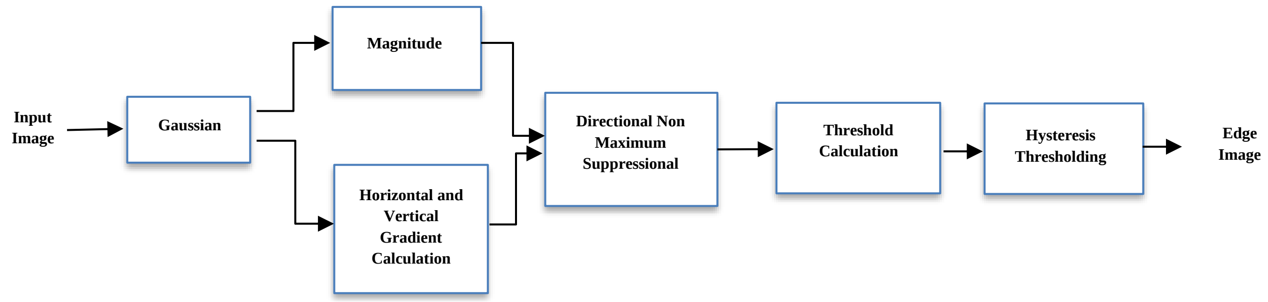


Fig. 3 Canny edge operator

Table 1. Advantages and disadvantages of operator

Edge detection method	Advantage	Disadvantage
Sobel	Simple to use and good noise suppression	Gives average output
Prewitt	Prewitt is less sensitive to noise	Results are sometimes noisier
Robert	Accurate positioning of images	Not much useful when noise is present
Laplacians	Detects good edges and their orientation	Sensitive to noise and nonrealistic contours
Canny	Better edge detection in the presence of noise	High complexity and time consuming

4.1.1. Entropy

Shannon's entropy is the amount of information gained from an image. Technically it is a measure of the degree of randomness in the image; entropy is very useful to get a measure of the amount of relevance the processed image is giving [17].

$$-\sum_{i=0}^{n-1} p_i p_i \text{ ………………….(10) [17]}$$

Where:
N = the number of gray levels (256 for 8-bit images)
Pi = the probability of a pixel having gray level i
B = the base of the logarithm function.

4.1.2. PSNR

To estimate the PSNR of an image, the comparison of that image with the real clean image of maximum possible power shall be made, peak signal noise ratio is an expression for the ratio of signal power to the noise power, and PSNR is expressed in decibels [16][18]. So

$$PSNR = 10\left(\frac{R^2}{MSE}\right) \text{ ………………….(11)[16]}$$

Where:
R= signal power
MSE = noise power

4.1.3. MSE

The distance between the actual and predicted values is the error made by the model [16]. Therefore, MSE is the mean of all the differences between actual and predicted values, which can be represented as

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \text{………(12)[16]}$$

Where:
N = the quantity of items,
Σ = summation notation,
Y_i = observed or original y-value,
$\hat{Y}_i$ = y-value from regression.

4.1.4. SNR

The ratio of signal power to noise power is known as the signal-to-noise ratio and is calculated as [18]

$$\frac{S}{N} = \frac{P_s}{P_n}$$

Where,
P_s = signal power
P_n = noise power

4.1.5. SSIM

A statistic called the structural similarity index measure is utilised to determine the similarities between two photographs. It mainly extracts features like luminance, contrast and structure to make a comparison between the images. SSIM has a value between -1 to +1. +1 denotes that the given images are similar to each other or the same, whereas -1 denotes that they are different from each other.

$$SSIM(x,y) = [l(x,y)]^{\alpha} \times [c(x,y)]^{\beta} \times [s(x,y)]^{\gamma} \text{ ………………… (13) [22]}$$

Where,

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{{\mu_x}^2 + {\mu_y}^2 + C_1}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

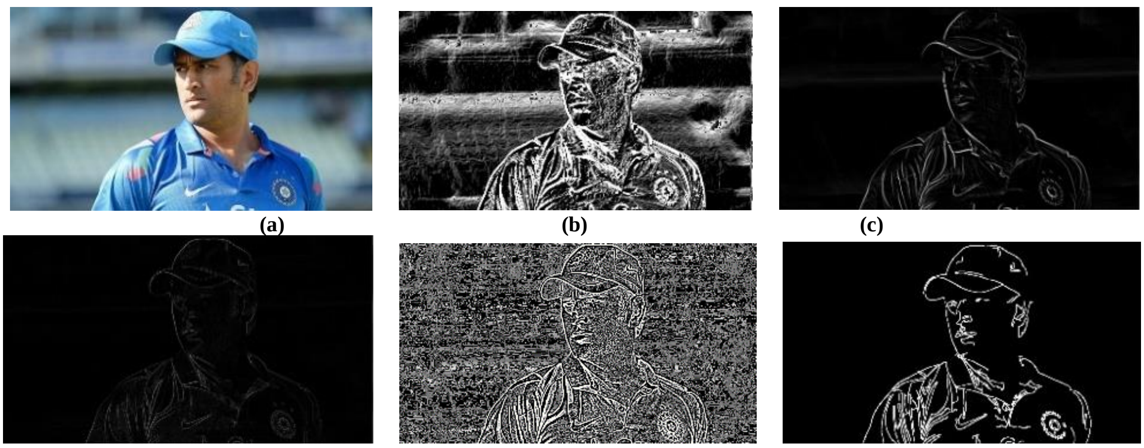
$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

Where $\mu_{x,}\mu_{y,}\sigma_x, \sigma_y$ *and* σ_{xy} are the local means, standard deviations and cross–covariance for images x, y.

If $\alpha = \beta = \gamma = 1$ and $C_3 = \frac{C_2}{2}$, then Index simplifies to:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \text{.(14) [22]}$$

4.2. Performance for Face Edge Detection

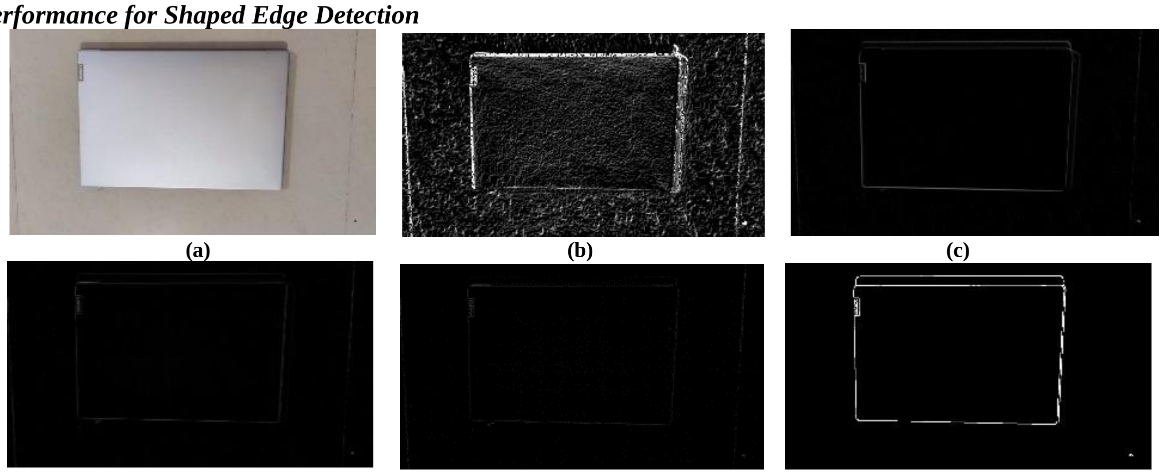


(a) **(b)** **(c)**
(d) **(e)** **(f)**

(a)Person(Original Image), (b)Sobel, (c)Prewitt, (d)Roberts, (e)Laplacian of Gaussian, (f)Canny

Fig. 4 Performance for face edge detection

4.3. Performance for Shaped Edge Detection



(a) (b) (c) (d) (e) (f)

(a)Laptop(Original Image), (b)Sobel, (c)Prewitt, (d)Roberts, (e)Laplacian of Gaussian, (f)Canny

Fig. 5 Performance for shaped edge detection

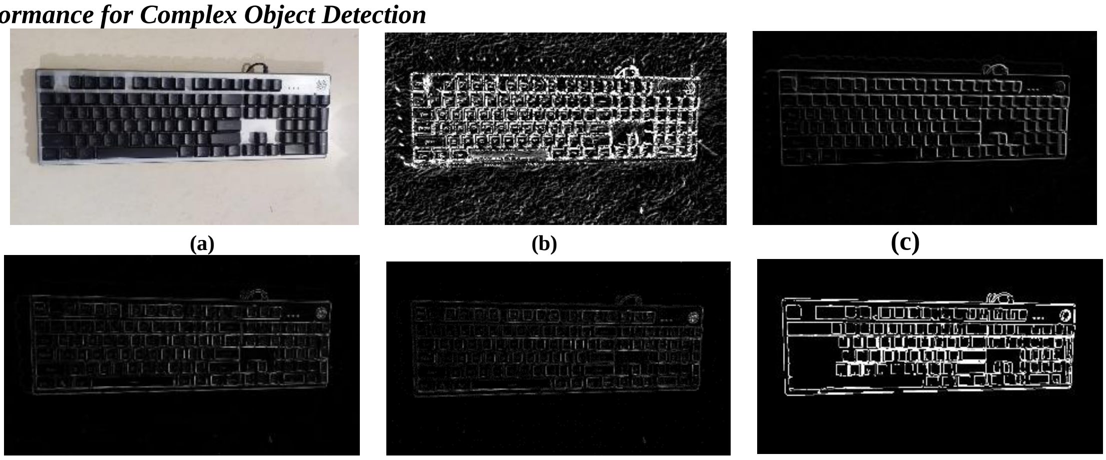
Table 2. Performance metrics for face edge detection

Methods	SNR	SSIM	Entropy	PSNR	MSE
Sobel	1.46837	0.03298	5.93146	27.89180	105.65750
Prewitt	0.39299	0.05263	4.01110	27.89514	105.57612
Robert	0.48342	0.00223	6.36011	27.68969	110.69051
Canny	0.40859	0.00786	0.59221	27.91320	105.13809
Laplacian	0.41111	0.03137	3.15711	27.90541	105.32686

Table 3. Performance metrics for shaped edge detection

Methods	SNR	SSIM	Entropy	PSNR	MSE
Sobel	0.93146	0.02288	5.81567	27.98639	103.38100
Prewitt	0.27299	0.01784	2.64252	27.62377	112.38362
Robert	0.38728	0.00351	3.33073	35.47370	18.437832
Canny	0.15458	-0.001728	0.15980	27.57217	113.72687
Laplacian	0.47601	0.01271	2.33731	27.58827	113.30602

4.4. Performance for Complex Object Detection



(a) (b) (c) (d) (e) (f)

(a)Keyboard(Original Image), (b)Sobel, (c)Prewitt, (d)Roberts, (e)Laplacian of Gaussian, (f)Canny

Fig. 6 Performance for complex object detection

Table 4. Performance metrics for complex object detection

Methods	SNR	SSIM	Entropy	PSNR	MSE
Sobel	1.11922	0.01476	5.35781	27.85461	106.56599
Prewitt	0.33825	0.03872	3.74098	27.84017	106.92087
Robert	0.41092	0.00228	6.20855	22.47742	367.56842
Canny	0.44862	-6.21883	0.65205	27.65212	111.65238
Laplacian	0.42038	0.06040	3.73929	27.81936	107.43441

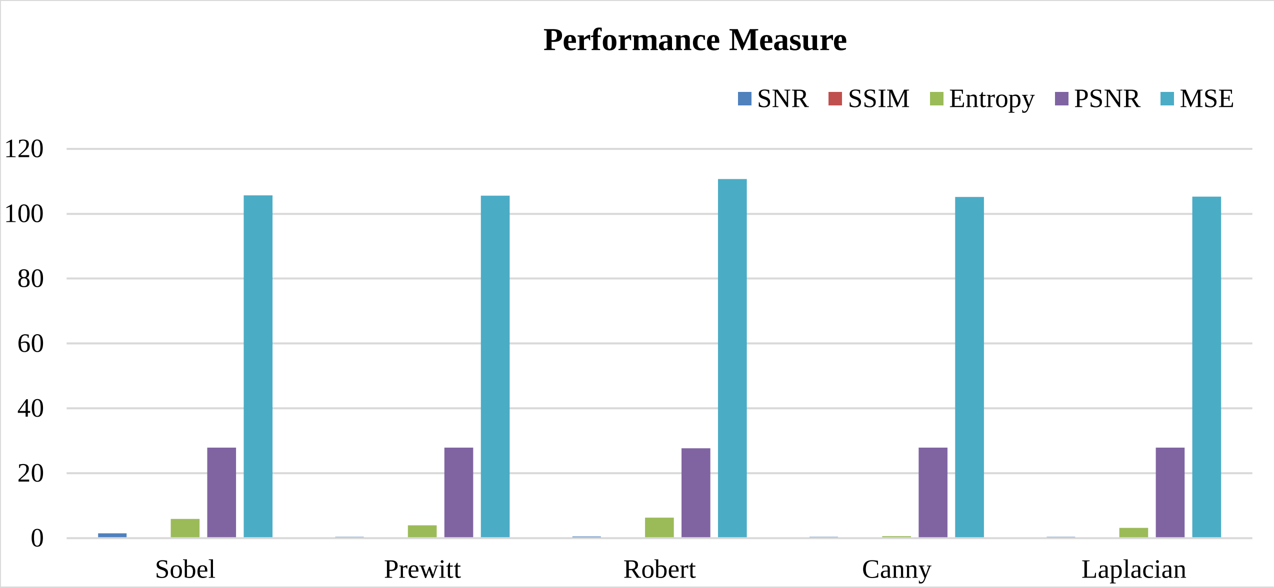
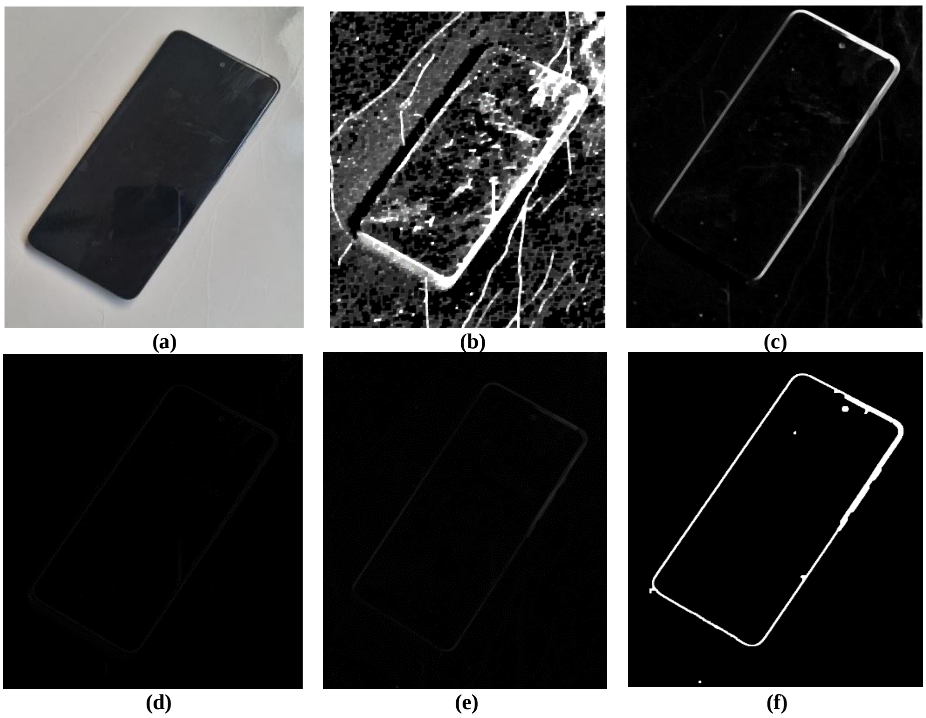


Fig. 7 Comparative analysis of different edge detection methods

5. Experiment Discussion

According to the edge detection performance of different operators, it was possible to find and analyse the working and results for different edge detection methods. The above experiment was performed using opencv and python. The edge detection has been performed on keyboard, laptop and face image, and for measuring the performance of the edge detectors, SNR, SSIM, entropy, PSNR and MSE parameters are used.

According to the order of derivatives, the sobel is outperforming other operators regarding SNR, which means that the sobel operator is extracting more useful information than the noise extracted with it, especially in the case of shaped objects. Prewitt operator has high SSIM as compared to other first-order derivatives, that mean prewitt provides high similarity of the edges with the original image, and as for entropy, the robert performs better than every other operator as it gives more information about the edges of the images as compared to the rest. When it comes to second-order derivatives, laplacian is performing better than canny, especially when it comes to entropy, but for face edge detection, canny is giving better PSNR as compared to laplacian.



(a) Object, (b) Sobel, (c) Prewitt, (d) Robert, (e) Laplacian of Gaussian, (f) Canny

Fig. 8 Edge Detection Comparison

Overall, it can be said that sobel is performing better than others. The sobel filter is a gradient-based technique that scans an image's first derivative for significant changes. Two 3*3 convolution masks are used by the sobel edge detector, one of which estimates the gradient in the x-direction and the other in the y-direction. It determines the direction and rate of change of the highest rise from light to dark, and it can be used with canny edge detection as canny offers us a lot of filters to work with a lot of customisations according to the use case.

As compared to other literature reviews for edge detection methods, authors have modified the edge detection algorithm to increase the performance and reduce the noise, but in the process of that, increased the complexity of operators. Also, the research and literature reviews show the study for edge detection that is not done specifically for dimension detection use cases or contour detection. Also, according to the above-observed metrics (PSNR), sobel and laplacian of gaussian are yielding a good amount of edges. However, the canny edge detection method performs better for contours or solid dimensions. Fig 8 shows the actual edge detection with noise reduction using gaussian blur and application on morphological operations like dilation and erosion on all the operators. The results clearly show that the canny edge detector outperforms other edge detection algorithms.

6. Conclusion

This paper shows the study and analysis of different types of edge detection methods/operators. According to the experiment, the second-order derivative, canny, is working better compared to first-order derivatives sobel, prewitt and robert. As the laplacian of the gaussian edge detection method is particularly noise-sensitive, thus it is not giving good enough results. That is why edge detection with canny edge detection, it may be argued, is a much superior technique for detecting edges of the object, compared to other operators (sobel, prewitt, roberts and laplacian of gaussian), and canny edge detector is less susceptible to noise. It is also observed that the sobel edge detector method is better than other operators when detecting the object's outer lines.

The edge detection algorithms have a number of problems that can be resolved by developing new filters or methodologies and by making appropriate improvements to currently used techniques. This would result in the better image output. In this experiment, It is found that while the sobel operator provides better image information gain, it is less accurate at recognising object shape, whereas the canny edge detection provides less information gain than the sobel edge detection but is more accurate at detecting shape and contour. The best results for shape detection can be obtained by combining sobel's information gain and canny edge detection precision.

References

[1] Chen Feng et al., "Image Edge Detection Based on Improved Local Fractal Dimension," *Fourth International Conference on Natural Computation,* pp. 640-643, 2008. *Crossref,* https://doi.org/10.1109/ICNC.2008.447

[2] Sheetal Israni, and Swapnil Jain, "Edge Detection of License Plate Using Sobel Operator," *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT),* pp. 3561-3563, 2016. *Crossref,* https://doi.org/10.1109/ICEEOT.2016.7755367

[3] Hanmin Ye, Bin Shen, and Shili Yan, "Prewitt Edge Detection Based on BM3D Image Denoising," *IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC),* pp. 1593-1597, 2018. *Crossref,* https://doi.org/10.1109/IAEAC.2018.8577643

[4] Miftahul Jannah, and Adli Abdillah Nababan, "Harfu Jar Detection System in Al-Quran Using Pierce Similarity Algorithm as a Basic Learning Media of Arabic Language," *3rd International Conference on Mechanical, Electronics, Computer, and Industrial Technology (MECnIT),* pp. 349-354, 2020. *Crossref,* https://doi.org/10.1109/MECnIT48290.2020.9166632

[5] Shweta Pardeshi, Pranali Pawar, and Nikhil Raj, "Real Time Object Measurement," *International Journal of Engineering Science and Computing,* vol. 11, no. 2, 2021.

[6] Geng Xin, Chen Ke, and Hu Xiaoguang, "An Improved Canny Edge Detection Algorithm for Color Image," *IEEE 10th International Conference on Industrial Informatics,* pp. 113-117, 2012. *Crossref,* https://doi.org/10.1109/INDIN.2012.6301061

[7] Theodora Sanida, Argyrios Sideris, and Minas Dasygenis, "A Heterogeneous Implementation of the Sobel Edge Detection Filter Using OpenCL," *2020 9th International Conference on Modern Circuits and Systems Technologies (MOCAST),* pp. 1-4, 2020 *Crossref,* https://doi.org/10.1109/MOCAST49295.2020.9200249

[8] Li Cao, Yi Jiang, and Mingfeng Jiang, "Automatic Measurement of Garment Dimensions Using Machine Vision," *International Conference on Computer Application and System Modeling (ICCASM),* pp. V9-30-V9-33, 2010. *Crossref,* https://doi.org/10.1109/ICCASM.2010.5623093

[9] Yi Zhang, "Edge Detection Algorithm of Image Fusion Based on Improved Sobel Operator," *IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC),* pp. 457-461, 2017. *Crossref,* https://doi.org/10.1109/ITOEC.2017.8122336

[10] Shaveta Malik, and Tapas Kumar, "Various Edge Detection Techniques on Different Categories of Fish," *International Journal of Computer Applications,* vol. 135, no. 7, pp. 6-11, 2016. *Crossref,* https://doi.org/10.10.5120/ijca2016908394

[11] Yolanda Gabyriela Ferandji, Diaraya, and Armin Lawi, "Performance Comparison of Image Edge Detection Operators for Lontara Sanskrit Scripts," *2nd East Indonesia Conference on Computer and Information Technology (EIConCIT),* pp. 241-244, 2018. *Crossref,* https://doi.org/10.1109/EIConCIT.2018.8878673

[12] P. Prathusha, S. Jyothi, and D. M. Mamatha, "Enhanced Image Edge Detection Methods for Crab Species Identification," *International Conference on Soft-computing and Network Security (ICSNS)*, pp. 1-7, 2018. *Crossref*, https://doi.org/10.1109/ICSNS.2018.8573629

[13] Akansha Jain et al., "Comparison of Edge Detectors," *International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom)*, pp. 289-294, 2014. *Crossref*, https://doi.org/10.1109/MedCom.2014.7006020

[14] Md Khurram Monir Rabby, Brinta Chowdhury, and Jung H. Kim, "A Modified Canny Edge Detection Algorithm for Fruit Detection & Classification," *10th International Conference on Electrical and Computer Engineering (ICECE)*, pp. 237-240, 2018. *Crossref*, https://doi.org/10.1109/ICECE.2018.8636811

[15] Hongli Lu, and Juan Yan, "Window Frame Obstacle Edge Detection Based on Improved Canny Operator," *3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*, pp. 493-496, 2019, *Crossref*, https://doi.org/10.1109/EITCE47263.2019.9095074

[16] Mohd. Aquib Ansari, Diksha Kurchaniya, and Manish Dixit, "A Comprehensive Analysis of Image Edge Detection Techniques," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 12, no. 11, pp. 1-12, 2017. *Crossref*, https://doi.org/10.14257/ijmue.2017.12.11.01

[17] Hui Zhang, Quanyin Zhu, and Xiang-feng Guan, "Probe into Image Segmentation Based on Sobel Operator and Maximum Entropy Algorithm," *International Conference on Computer Science and Service System*, pp. 238-241, 2012. *Crossref*, https://doi.org/10.1109/CSSS.2012.67

[18] Pinaki Pratim Acharjya, Ritaban Das, and Dibyendu Ghoshal, "Study and Comparison of Different Edge Detectors for Image Segmentation," *Global Journal of Computer Science and Technology*, vol. 12, no. 13, 2012.

[19] Manasa R, K Karibasappa, and J. Rajeshwari, "Autonomous Path Finder and Object Detection using an Intelligent Edge Detection Approach," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 9, no. 8, pp. 1-7, 2022. *Crossref*, https://doi.org/10.14445/23488379/IJEEE-V9I8P101

[20] Tasnuva Tasneem, and Zeenat Afroze, "A New Method of Improving Performance of Canny Edge Detection," *2nd International Conference on Innovation in Engineering and Technology (ICIET)*, pp. 1-5, 2019, *Crossref*, https://doi.org/10.1109/ICIET48527.2019.9290676

[21] Manish Yewange et al., "IJReal-Time Object Detection by Using Deep Learning: A Survey," *International Journal of Innovative Science And Research Technology*, vol. 7, no. 4, 2022. *Crossref*, https://doi.org/10.5281/zenodo.6568421

[22] Deepak T. Mane et al., "Pattern Classification using Supervised Hypersphere Neural Network," *International Journal of Emerging Technology and Advanced Engineering*, vol. 12, no. 8, 2022. *Crossref*, https://doi.org/10.46338/ijetae0822_25

[23] Liying Yuan, and Xue Xu, "Adaptive Image Edge Detection Algorithm Based on Canny Operator," *4th International Conference on Advanced Information Technology and Sensor Application (AITS)*, pp. 28-31, 2015, *Crossref*, https://doi.org/10.1109/AITS.2015.14

[24] Ashish Anand, Sanjaya Shankar Tripathy, and Sukesh Kumar, "An Improved Edge Detection Using Morphological Laplacian of Gaussian Operator," *2nd International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 532-536, 2015. *Crossref*, https://doi.org/10.1109/SPIN.2015.7095391

[25] Beixin Xia, Jianbin Cao, and Chen Wang, "SSIM-NET: Real-Time PCB Defect Detection Based on SSIM and MobileNet-V3," *2nd World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM)*, pp. 756-759, 2019, *Crossref*, https://doi.org/10.1109/WCMEIM48965.2019.00159

[26] Sunil Kumar et al., "Comparative Analysis for Edge Detection Techniques," *International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pp. 675-681, 2021. *Crossref*, https://doi.org/10.1109/ICCCIS51004.2021.9397225