NavaJyoti, International Journal of Multi-DisciplinaryResearch
Volume 7, Issue 2, August 2023

A COMPARATIVE STUDY ON EDGE DETECTION TECHNIQUES

Dr. Senthil Vadivu M[1], **Smruthi.R**[2]

[1] Assistant Professor, Jyoti Nivas College Autonomous, Bangalore, India

[2] Student, Jyoti Nivas College Autonomous, Bangalore, India

Abstract: Edge detection is a core task in computer vision and image processing, with many applications such as object recognition, scene segmentation, and feature extraction. In this comparative study, two popular edge detection techniques are analyzed: the Canny edge detector and the HED (Holistically-Nested Edge Detection) algorithm. The comparison evaluates these algorithms on the BSDS500 dataset, which contains 500 natural images with manually annotated ground-truth edges. And compare the performance of the two algorithms using two objective metrics: the mean squared error (MSE) and the peak signal-to-noise ratio (PSNR).

Keywords: Canny edge detector, HED (Holistically-Nested Edge Detection) algorithm,MSE,PSNR.

1.INTRODUCTION

A crucial component of image processing is edge detection, which involves determining the boundaries between image elements. It is a crucial pre-processing step for many applications, such as object identification, segmentation, and tracking. The Canny and HED (Holistically-Nested Edge Detection) algorithms are two well-known edge detection techniques. The success of these two strategies will be evaluated in this project report using the mean squared error (MSE) and peak signal to noise ratio (PSNR) figures.

2. LITERATURE SURVEY

Mohd.Aquib Ansari, Diksha Kurchaniya, Manish Dixit,[1], To locate the edges of an image, they used edge detection operators like the Roberts, Prewitt, Sobel, and Canny edge detection algorithms. A comparison of the results of various edge detection techniques was then conducted using the PSNR (Peak Signal-to-Noise Ratio) and MSE (Mean Square Error) parameters. Canny was the one that ultimately proved to be the best after considering every other option.

Rui sun, Tao Lei, QiChen, Zexuan Wang,[2],Two kinds of edge detection algorithms—those that are based on conventional learning methods and those that are based on deep learning—have been put together by the authors.They discovered that in the realm of computer vision, edge detection is the area in which deep learning is most commonly utilised.

Jun Feng Jing, Weichung Zhang, Changming Sun,[3], The optimal dataset scale (OSD), optimal image scale (OSI), average precision (AP), F-measure PSNR (Peak Signal-to-Noise Ratio), and MSE (Mean Square Error) values are used in this study to compare the performance of existing hand-crafted edge detection methods such as Canny, SLEDGE, Robert, etc., and machine learning-based edge detection methods such as Deep Edge, COB, PiDiNet, etc. The evaluation's findings demonstrate that, when used on various photos in a data set, each approach has advantages and disadvantages of its own. It is finally chosen whether a hand-crafted or machine learning-based strategy is better for each particular image in the collection.

Imankadhim, Ajlan, AlaaAbdulhusseinDaleh, Al-magsoosi, HayderG.Murad [4], The authors reviewed edge detection methods based on discontinuity intensity levels, including Operation Sobel, Operation Roberts, and Operation Laplace. With varied colour images, multiple edge detection approaches are compared for performance. Peak Signal-to-Noise Ratio (PSNR) and MSE (Mean Square Error) figures are used to assess the approaches. The author discovered that Laplace produces the greatest results since it draws attention to the shape's outlines and that this approach is quicker than the other two. Robert's method, on the other hand, is comparably superior than Sobel's method in terms of performance and execution time.

Benjamin Kommey,Jhon Kwame Dunyo,Eric Tutu Tchao,Andrew Selasi Agbemenu,[5],The most popular edge detection algorithms, including Roberts, Prewitt, Sobel, and Canny, are compared. The best results are determined by analysing MSE (Mean Square Error), PSNR (Peak Signal-to-Noise Ratio), and Execution Time (Et). Out of these three metrics, Canny Edge Detection produces the best accuracy and least amount of noise (PSNR), but it takes slower to complete (MSE).

Ehsan. Akbari Sakahravani, Mehdi Masoodi,Eduard Babulak,[6],In this paper,before applying the median filter, all edges that are identified often contain noise and are not continuous. As a result, the noise-reducing technique is crucial since standard Canny edge detection is extremely noise-sensitive. Using the suggested approach, edges in noisy pictures can be effectively recognised, and it performs edge and detail identification better than the conventional Canny algorithm. Similarly, increase the accuracy of edge detection.

Dipika Deshmukh, GajananKurundkar [7], In this paper, the method of video steganography has become essential for information security in the age of quick internet communication. More information may be hidden in video than in picture. In this Paper, various edge detection methods and their associated procedures are analyzed. Compared to other approaches like Prewitt, Sobel, Robert, etc., the canny edge detector provides more edge pixels for data hiding.

Li Lou,Shasha Zang[8], the authors provide a deeper convolution feature-based edge detection technique(HED). Edge detection requires learning extensive hierarchical representations since objects in natural photos have various sizes and aspect ratios. The model that this research can successfully extract the edges of the objects in a picture, enhance target edge detection's efficacy and integrity, and provide a solid framework for image segmentation.

Ashok Kumar,H. L. Mandoria,B. K. Pandey Subodh Prasad,[9], This paper compares several edge detection methods and gives a study on edge detection. Sobel's approach produces edges with greater accuracy, Canny produces the best results with the lowest error rate, and Prewitt is easily computed are conclusions from this paper

Saining Xie ,Zhuowen Tu [10], the authors, provide a novel convolutional neural network-based edge detection system that exhibits cutting-edge performance on real-world photos at a pace that is relevant to practise.

3.PROPOSED METHODOLOGY

In order to generally determine which edge detection technique is the basically the best for a pretty certain image in the dataset, I have compared the Canny Algorithm with the HED algorithm in a major way. The image definitely is initially obtained from a folder, the noise is reduced, and the image mostly is then really sent through algorithms to definitely identify edges in a subtle way. MSE (Mean Square Error) and PSNR are the two metrics that literally are used to evaluate the methods (Peak for all intents and purposes Signal-to-Noise Ratio). The very much the best algorithm for a definitely specific image is the one with the basically the highest PSNR and hardly the lowest MSE value in a very major way. HED really is said to literally be the best algorithm for edge detection when compared to canny since the majority of images exhibit the condition of HED in a major way.

3.1 Algorithm

3.1.1 Canny Algorithm

An effective approach for identifying edges in digital images is canny edge detection. It was created in 1986 by John F. Canny and is frequently employed in computer vision and image processing applications. There are multiple steps to the algorithm, which may be summed up as follows:

- **Gaussian blur:** The Canny technique begins by using a Gaussian filter to smooth out the picture. This is done to reduce image noise and small variations that might lead to spurious edges. The Gaussian kernel's size can be adjusted to control the level of blurring.
- **Gradient computation:** Calculating the image's gradient is the next step after blurring the original image. The gradient calculates how quickly the intensity of the image changes in various directions. The Sobel operator, a straightforward filter that approximates the gradient by calculating the intensity difference between adjacent pixels, is the most used method for computing the gradient.
- **Non-maximum suppression:** After that, non-maximum suppression is carried out. Suppressing all values other than local maxima requires scanning the gradient magnitude picture. In order to maintain just the strongest edges, the edges are thinned.
- **Double thresholding:** Following non-maximum suppression, this process is carried out. Setting a high threshold and a low threshold entail doing this. Strong edges are defined as all edges with a gradient magnitude greater than the high threshold, whereas weak edges are defined as edges with a gradient magnitude between the low and high thresholds. All other edges are muted. Double thresholding is used to get rid of noise and flimsy edges that aren't likely to be true edges.
- **Edge tracking by hysteresis:** The Canny algorithm performs edge tracking through hysteresis as its last step. In order to do this, the weak edges must be linked to the strong edges by tracing their pathways. This is done to close up edges that have gaps in them and create an edge map that is more continuous.

The result of the Canny algorithm is a binary edge map that indicates the locations of edges in the image. The algorithm is effective in detecting edges with a high degree of accuracy and is robust to noise and other artifacts in the image. However, the choice of the threshold values can be challenging, and the algorithm can be sensitive to the size and orientation of the edges.In OpenCV, Canny edge detection can be performed using the cv2.Canny() function.

3.1.2 HED Algorithm

The advanced edge detection technique HED (Holistically-Nested Edge Detection) creates high-quality edge maps using a deep neural network. Xie and Tu first introduced the approach in their 2015 publication titled "Holistically-Nested Edge Detection."

The foundation of the HED technique is a deep convolutional neural network that has been trained on a sizable dataset of annotated pictures to discover how to recognise edges at various dimensions and orientations. A hierarchy of layers in the network architecture capture data from the input picture at different sizes, and then a sequence of fusion layers combine these characteristics to create the final edge map.

To use the HED algorithm in Python with OpenCV, you need to have the HED model weights and the deploy.prototxt file, which contains the network architecture. These can be downloaded from the official HED GitHub repository.

Once you have downloaded the necessary files, you can load the HED model into OpenCV using the cv2.dnn.readNetFromCaffe() function. You can then run an input image through the model using the net.setInput() and net.forward() functions.

The HED algorithm involves the following steps:

- **Pre-processing:** A Gaussian filter is used to remove noise from the input image, and it is then converted to grayscale.
- **Multi-scale feature extraction:** To extract multi-scale features, the pre-processed picture is then put into a deep neural network that has already been trained. The picture is sent through numerous convolutional layers with various filter sizes and stride lengths to achieve this.
- **Holistic Nesting:** The features from each scale are then concatenated and used as input to a set of fully convolutional layers that predict the edge maps at each scale. The fully convolutional layers are called "holistic" because they operate on the concatenated features from all scales simultaneously.
- **Up sampling and merging:** The edge maps predicted at different scales are then up sampled to the original image size and merged to produce the final edge map. The merging process involves combining the edge maps at different scales using a weighted average.
- **Postprocessing:** The final edge map is postprocessed to remove false edges and smooth the remaining edges using a non-maximum suppression algorithm and a Gaussian filter, respectively.

The obtained edge map may be applied to a number of computer vision applications, including object detection, scene interpretation, and segmentation. Compared to conventional edge detection algorithms, the HED method has demonstrated enhanced performance in recognising complicated and fine-grained edges in real photos.

3.1.3 Evaluation Metrics

PSNR (Peak Signal-to-Noise Ratio)

Peak Signal-to-Noise Ratio, or PSNR, is a metric indicating the effectiveness of reconstructing a compressed or degraded picture. The ratio of the highest pixel value to the root mean squared error (RMSE) of the two pictures is obtained by comparing the original image to the reconstructed image. The quality of the reconstructed image is higher than the original image when the PSNR is higher.

The ratio between the largest potential value (peak signal) and the amount of noise present in the signal (measured as mean squared error, or MSE), is used to determine the quality of a picture or video. When comparing the effectiveness of various denoising algorithms, image enhancing techniques, or compression methods for images or videos, PSNR is frequently employed as a quantitative parameter. Higher numbers denote greater quality and are measured in decibels (dB). The PSNR calculation is:

$$PSNR = 20 * \log10(MAXp) - 10 * \log10(MSE)$$

where MAXp is the maximum possible pixel value (e.g., 255 for an 8-bit image) and MSE is the mean squared error between the original and processed images.

MSE (Mean Squared Error)

The average squared difference between two pictures is measured by MSE, or mean squared error. To compute it, subtract the pixel values from the two photos, square the result, and then average the resulting values over the whole image. The two photos are more comparable when the MSE is reduced.

It is a quantitative way to compare two photographs, often the original and processed versions. By averaging the squared differences between matching pixels in the two pictures, MSE is determined. The performance of image processing methods, such as denoising, compression, or restoration procedures, is frequently assessed using the MSE value as a measure. An algorithm works better when the MSE value is lower since it means that the processed picture is more similar to the original image. The MSE equation is:

$$MSE = (1/n) * sum[(I(i,j) - K(i,j))^2]$$

where I(i,j) represents the pixel value of the original picture at position (i,j), K(i,j) represents the pixel value of the processed image at the same place, and n represents the total number of pixels in the image. To get the MSE value, the square of the pixel difference is added up over all pixels, and the average is then calculated.

3.2 Flow Diagram with explanation

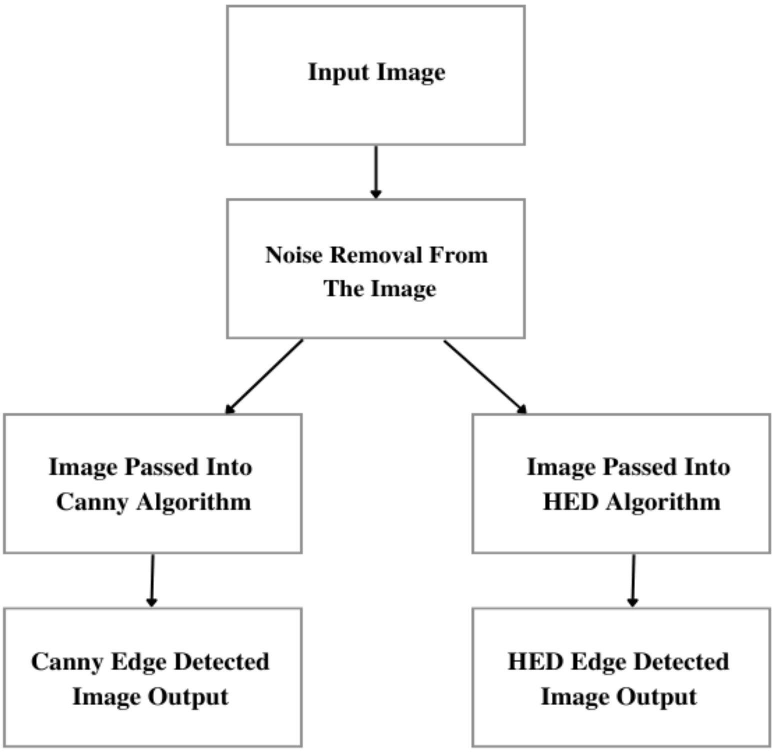


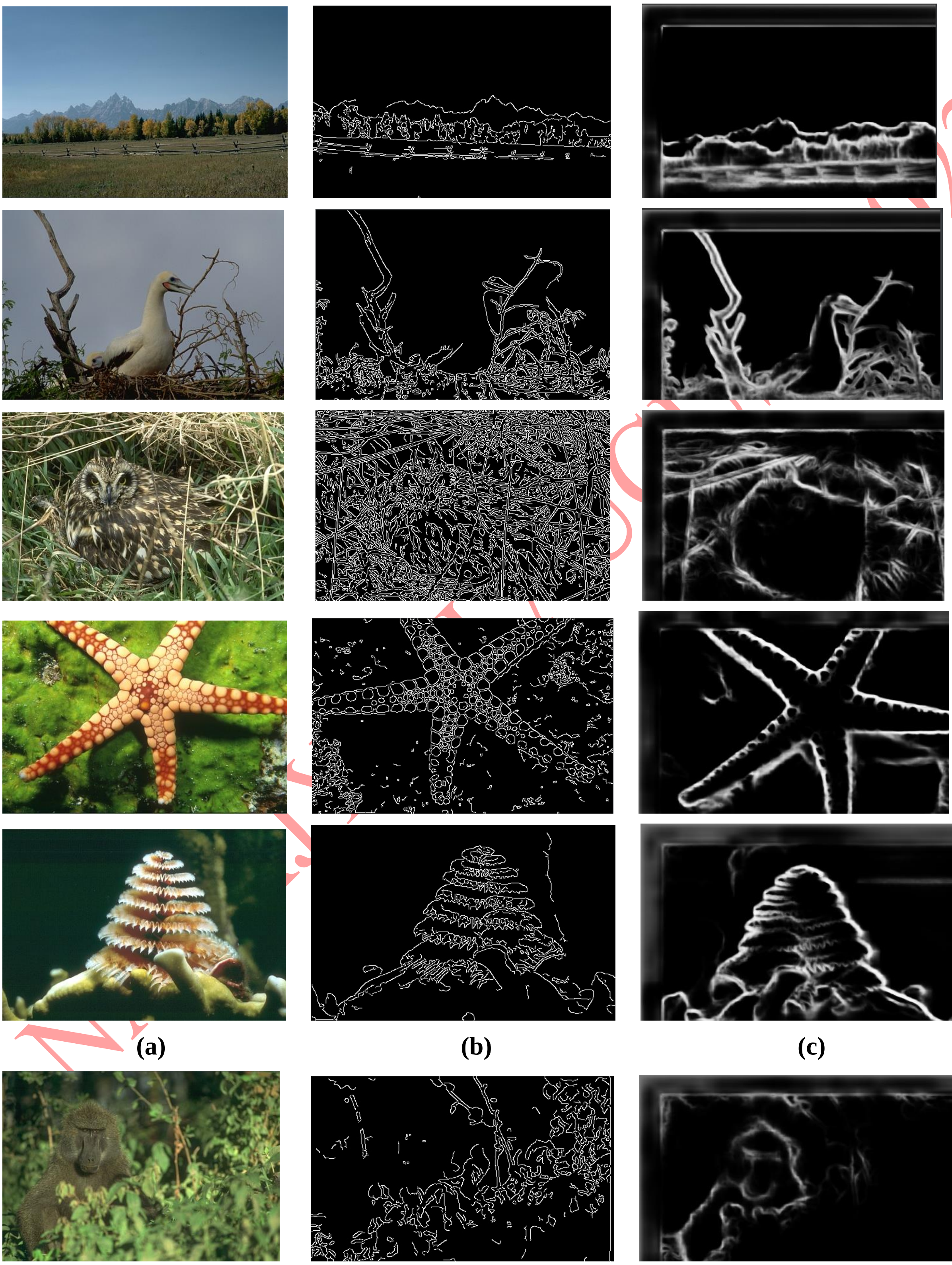
Figure 1: The Flowchart of the Methodology

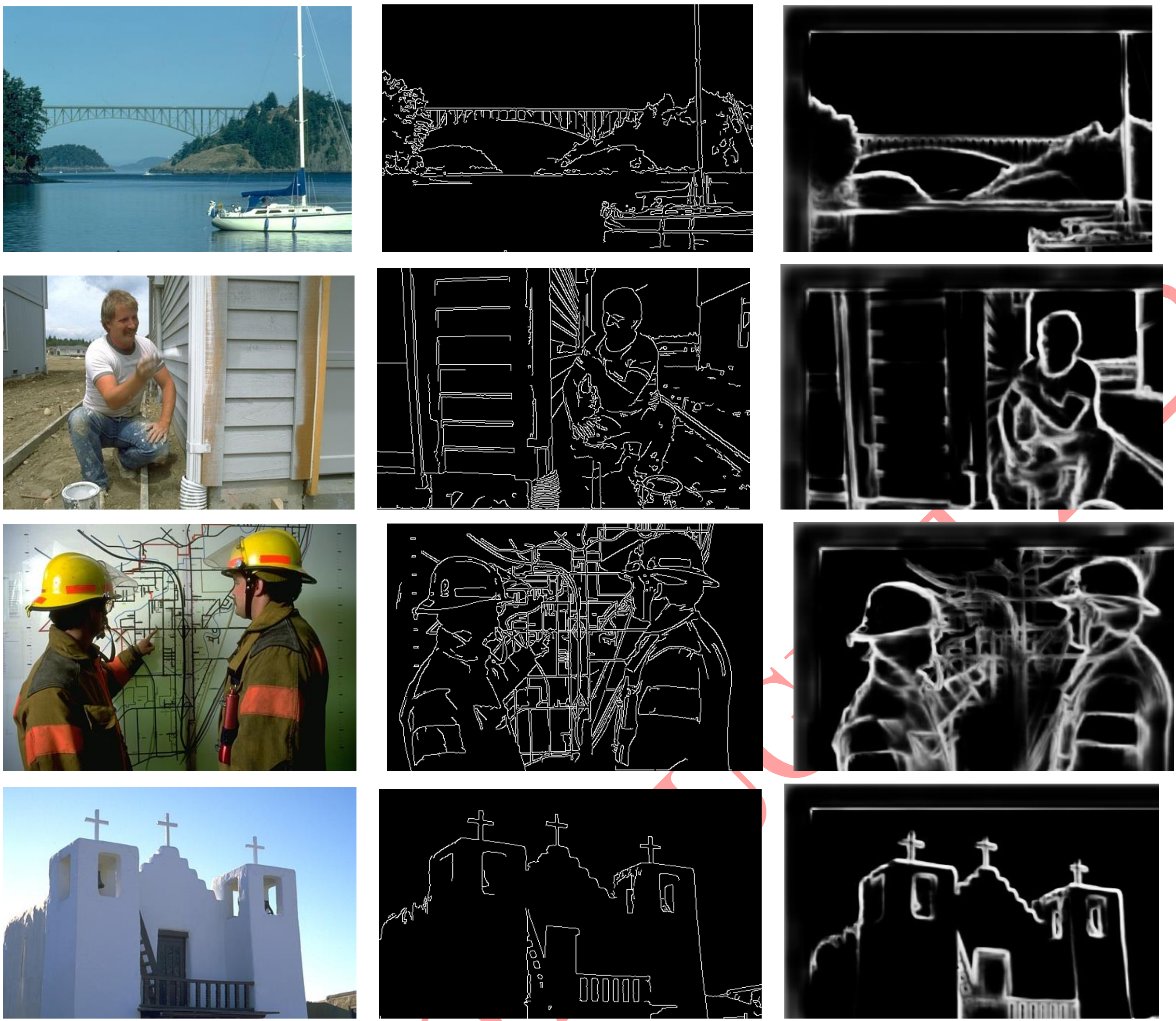
Flowchart explanation:

1. The code defines the input and output folders for the images.
2. It loads the HED and Canny models using the pre-trained weights.
3. It loops over all image files in the input folder.
4. For each image, it reads the input image and removes noise using Gaussian blur.
5. It converts the input image to grayscale and creates a blob for the HED model.
6. It runs the image through the HED model and resizes the output to the original image size.
7. It runs the image through the Canny edge detector.
8. It saves the output images to the output folder.
9. It calculates the MSE and PSNR values for both methods.
10. It prints the MSE and PSNR values for each method and determines the best method based on the highest PSNR value and lowest MSE value
11. It repeats the process for all images in the input folder.
12. Finally, the code prints "All images processed and saved" to indicate the end of the script.

4.RESULTS AND DICUSSION.

4.1 Discussion of the results





(a) (b) (c)

Figure 2 (a)Input Image (b)Canny Edge Detection (c)HED Algorithm

4.2 Performance Evaluation

Input Images	HED (Holistically-Nested Edge Detection)		Canny Edge Detection	
	(MSE)	(PSNR)	(MSE)	(PSNR)

2092.jpg	99.44	7.43	99.57	6.78
12003.jpg	106.95	6.92	107.22	6.49
12074.jpg	103 .62	9.38	103.67	8.95
15004.jpg	106.10	6.56	110.50	5.61
15088.jpg	112.41	4.88	115.01	4.65
16052.jpg	106.59	8.83	108.36	7.82
20008.jpg	112.94	6.76	115.01	6.06
22013.jpg	99.44	7.43	99.48	6.78

22090.jpg	100.63	6.18	102.39	5.39

Table 1: Performance Evaluation of Algorithms

HED (Holistically-Nested Edge Detection)

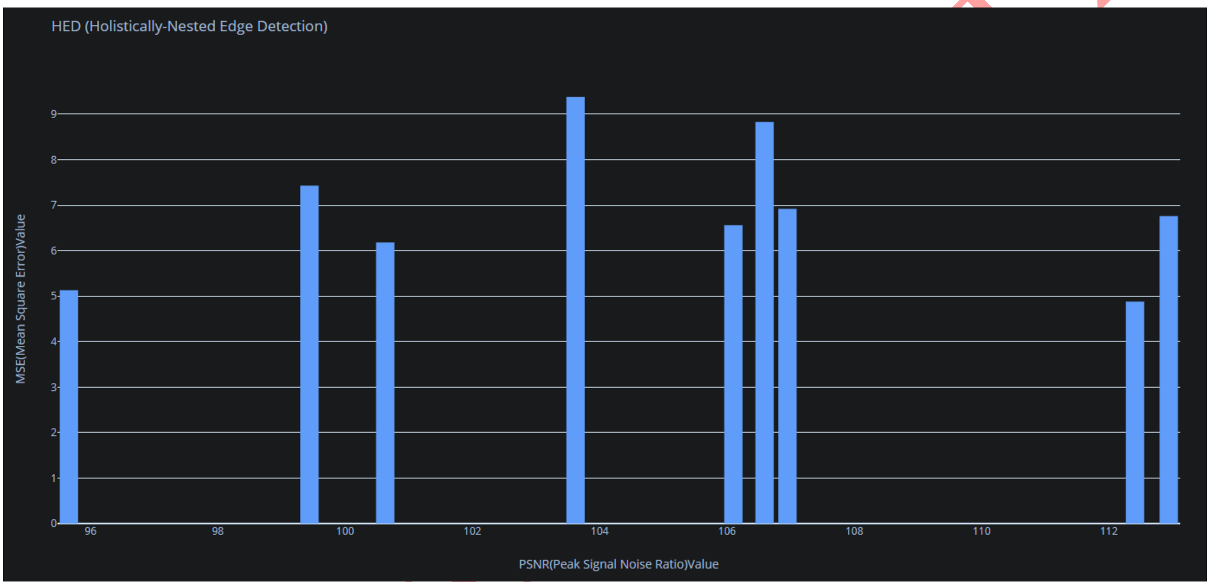


Figure 3: The graph is representing the performance evaluation of HED from table 1

Canny Edge Detection

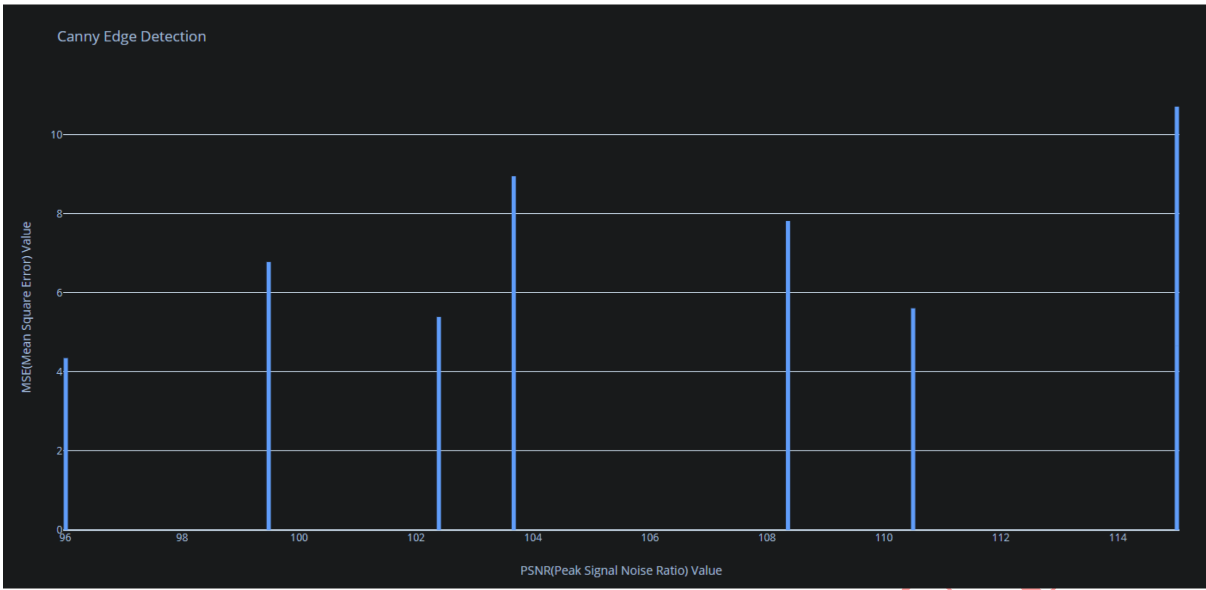


Figure 4: The graph is representing the performance evaluation of Canny from table 1

5.CONCULSION

The conclusion for this paper is that the two methods for edge detection are efficient after comparing edge detection using the Canny and HED algorithms. The Canny algorithm is a well-known edge detection technique that performs well in the majority of cases. It works well at identifying edges with minimal noise and is fast and simple to use. Whereas, the HED algorithm, on the other hand, is a more advanced and computationally demanding technique that is relatively new. It has demonstrated to generate more accurate results than the Canny algorithm in specific circumstances and employs a deep neural network to detect edges. Both methods achieved comparable results in terms of performance measures like PSNR and MSE, with minor differences depending on the image being analyzed.

REFERENCES

[1] Mohd.Aquib Ansari, Diksha Kurchaniya, Manish Dixit, “A Comprehensive Analysis of Image Edge detection Techniques”, International Journal of Multimedia and Ubiquitous Engineering vol.12, No.11(2017).

[2] Rui sun, Tao Lei, QiChen, Zexuan Wang, “Survey of Image Edge Detection”, Frontiers in Signal processing, Volume 2 article 826967,09 March 2022.

[3] Jun Feng Jing, Weichung Zhang, Changming Sun, “Recent advances on image edge detection: A comprehensive review”, Neurocomputing, vol.22, article September 2022

[4] Imankadhim, Ajlan, AlaaAbdulhusseinDaleh, Al-magsoosi, HayderG.Murad,"A compartive Study

of Edge Detection Techniques in Digital Image", Joural of Mechanical Engineering Research and Developments, vol.44, No.7, pp.109-119,2021

[5] Benjamin Kommey,Jhon Kwame Dunyo,Eric Tutu Tchao,Andrew Selasi Agbemenu,"A Comparative Study Of Common Edge Detection Operators In Digital Image Processing",JournalofResearch in Engineering and Applied Sciences,JREAS,Vol.06,Issue 04,Oct 2021

[6] Ehsan. Akbari Sakahravani, Mehdi Masoodi,Eduard Babulak,"Implementing canny Edge detection algorithm for noisy image",Bulletin of Electrical Engineering and Informatics Vol. 9, No. 4, August 2020, pp. 1404~1410 ISSN: 2302-9285, DOI: 10.11591/eei.v9i4.1837

[7] Dipika Deshmukh,GajananKurundkar"Comparative Study Of Edge Detection Techniques For Steganography",International Joural Of Advanced Research,Article DOI:10.21474,ISSN:2320-5407

[8] Li Lou,Shasha Zang,"Research on Edge Detection Method Based on Improved HED Network ",Journal of Physics: Conference Series,1607 (2020) 012068,IOP Publishing doi:10.1088/1742-6596/1607/1/012068

[9] Ashok Kumar,H. L. Mandoria,B. K. Pandey Subodh Prasad,"A Comparative Study of Different Edge Detection Techniques",International Research Journal of Engineering and Technology (IRJET),Volume: 07 Issue: 10 | Oct 2020

[10] Saining Xie ,Zhuowen Tu ,"Holistically-Nested Edge Detection ",arXiv:1504.06375v2 [cs.CV]