

Article

A Low Redundancy Wavelet Entropy Edge Detection Algorithm

Yiting Tao ^{1,*}, Thomas Scully ^{2,†}, Asanka G. Perera ¹ , Andrew Lambert ²  and Javaan Chahl ^{1,3} 

¹ UniSA STEM, Mawson Lakes Campus, University of South Australia, Adelaide, SA 5095, Australia; asanka.perera@mymail.unisa.edu.au (A.G.P.); Javaan.Chahl@unisa.edu.au (J.C.)

² School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2610, Australia; tscully@fortinet.com (T.S.); a.lambert@adfa.edu.au (A.L.)

³ Joint and Operations Analysis Division, Defence Science and Technology Group, Melbourne, VIC 3207, Australia

* Correspondence: yiting.tao@mymail.unisa.edu.au

† These authors contributed equally to this work.

Abstract: Fast edge detection of images can be useful for many real-world applications. Edge detection is not an end application but often the first step of a computer vision application. Therefore, fast and simple edge detection techniques are important for efficient image processing. In this work, we propose a new edge detection algorithm using a combination of the wavelet transform, Shannon entropy and thresholding. The new algorithm is based on the concept that each Wavelet decomposition level has an assumed level of structure that enables the use of Shannon entropy as a measure of global image structure. The proposed algorithm is developed mathematically and compared to five popular edge detection algorithms. The results show that our solution is low redundancy, noise resilient, and well suited to real-time image processing applications.

Keywords: edge detection; wavelet decomposition; Shannon entropy



Citation: Tao, Y.; Scully, T.; Perera, A.G.; Lambert, A.; Chahl, J. A Low Redundancy Wavelet Entropy Edge Detection Algorithm. *J. Imaging* **2021**, *7*, 188. <https://doi.org/10.3390/jimaging7090188>

Academic Editors: Philippe Montesinos and Baptiste Magnier

Received: 7 July 2021

Accepted: 13 September 2021

Published: 17 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Edges of an image are considered to be a low-level image feature. They are undoubtedly one of the most important features in an image. Edge detection research spans several decades but still remains as an active research area due to its applicability to a wide range of computer vision tasks.

In the literature, a range of edge detection techniques have been developed and used by the computer vision community. Edge detection techniques are extensively used in many application areas such as outlier detection in medical images [1], remote sensing [2], image-to-image translation [3], photo-sketching [4], optical character recognition [5], and robotic vision [6].

Edge detection is a fundamental step in image processing. Often, edge detection is the first step applied to an image field in preparation for further image processing operations. Edge detection allows an image field to be segmented to extract important features, reducing redundancy in an image field and enabling subsequent image processing task complexity to be significantly reduced. Robust and computationally efficient techniques are therefore central to many image processing applications.

An edge is defined as the change of intensity within a given image field $I(x, y)$. Edges within $I(x, y)$ are described using gradient magnitude. Whilst this definition of an edge is simplistic, identification and extraction of edges are a non-trivial task. The non-trivial nature of edge detection and the importance to subsequent image processing tasks has resulted in edge detection being an active area of research for decades [7–11]

Multiple edge detectors have been developed with no single edge detector algorithm providing the ultimate solution across all image scenarios. Prevalent edge detection methods such as Canny [12], Prewitt [13], Roberts [14], Sobel [15] and Zerocross [16] have

proven popular due to their efficiency and ease of implementation, making them useful for real world image processing applications [11,17,18].

In this paper, we propose a new edge detection method based on the coupling of wavelet and entropy based techniques. This new edge detection algorithm capitalizes on the algorithmic efficiency of wavelet based techniques while incorporating entropy techniques to select the most appropriate wavelet scale to analyze for edges. Wavelet decomposition is used to extract edge location and orientation data across multiple frequency domains. Entropy is then used to assess the multi-scale decomposition data and determine which wavelet decomposition level contains the greatest image structure. This process produces a low redundancy edge detection output image with a significant increase in computational efficiency. In doing so, the new method is well suited to implementation in real-time image processing applications.

This paper is organized as follows: closely related work has been discussed in Section 2. Section 3 describes the proposed algorithm in detail. Section 4 provides experimental results and analysis of our solution when compared to five widely used edge detection algorithms. The computational efficiency and the resilience of the approach to noise was analyzed. Section 5 concludes the study.

2. Related Work

Several works have been published in the edge detection area. A complete list of studies can be found in these surveys [7,19]. For ease of analysis, in the following subsections, we broadly categorize edge detection algorithms based on their design.

2.1. First Order

Prewitt, Roberts and Sobel edge detection techniques are first order differential image based filters. Therefore, each filter has strengths and weaknesses when detecting edges in a scene. Thus, no single first order filter provides a generalized edge detection solution for all image scenarios [20,21]. First order methods typically perform poorly when the image contains a high noise content [22]. Furthermore, these operators are typically computationally expensive as they require convolution to enable implementation.

2.2. Second Order

A number of second order differential methods [16,22–25] exist for the detection of edges. Second order differential methods detect the crossing of the zero gradient axis of $\nabla^2 I(x, y)$ to define the location of edges within an image. As a result, second order differential methods are sensitive to noise within the image, making them less robust [22]. To increase robustness, second order differential methods typically employ filtering, which in turn results in loss of edge data contained within the image.

The Canny Edge Detector is accepted as the industry standard for edge detection techniques [22]. John Canny [12] aimed to create an optimized edge detection technique that satisfied a comprehensive set of goals for the computation of optimal edge points. The Canny Edge Detector succeeded in creating a robust and computationally efficient edge detection technique; however, it relies heavily on image smoothing to reduce noise within images. Smoothing is typically achieved through Gaussian filtering of the image, resulting in degradation of edge data. As a result, the Canny Edge Detector can break down when the gradient magnitude is small in comparison to the noise present within the image [22]. An approximation of Canny edge detection has been developed by Deriche [26] and by Van Vleit et al. [27]. Deriche proposed an extension to the two-dimensional (2D) case (Canny's optimal edge detector), with the resulting filtering structures implemented as 2D infinite impulse response (IIR) digital filters.

2.3. Entropy Based

Entropy based methods have also been investigated as a more robust and computationally efficient method for edge detection [28–36]. Entropy based methods apply the

concept of information entropy [37] to determine whether structured edges are present. Entropy based methods typically calculate edges through thresholding based on evaluation of a global entropy value [28,29,38] or through comparison of localized entropy calculations with $I(x, y)$ [30,32,34,35]. Entropy based methods assume that the calculated entropy value will be representative of the amount of structure present within the image. Entropy based methods have proved to be more resilient to noise within an image field while being computationally efficient compared to gradient based methods such as Prewitt, Roberts, Sobel, and Zerocross [22,32,36].

2.4. Wavelet Based

Wavelet based methods have been investigated as an efficient edge detection algorithm [39–50]. First and second derivative methods are based on Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filter design which can be computationally expensive to implement digitally. Wavelet based methods offer a method known as sub-band coding to enable shorter filter designs paired with subsampling to achieve more efficient computation than traditional Fourier based methods [51]. Multiscale analysis of wavelet decomposition levels enable more advanced techniques to be applied in the analysis of edge components within a given image.

2.5. Deep Learning/Machine Learning Based

In the past few years, deep learning based edge detectors have been proposed and developed. Because some Deep Learning (DL) methods are capable of predicting edges with more efficiency and accuracy, it has become popular in this field. More recently, DL methods that use convolutional neural networks (CNNs) to achieve the boundary and contour detection tasks have become popular, including some well known CNN based methods like HED [10], RCF [17], DeepEdge [52], DeepContour [53], and BDCN [54]. For instance, the HED approach offers image-to-image training and prediction, which produces the predicted edge map image directly [10]. Richer Convolutional Features (RCF) train a network using multi-scale and multi-level information to achieve contour prediction [17]. While deep learning-based methods remain state-of-the-art in the edge detection area, they need an extensive amount of work and resources for training and running compared to classical edge detection algorithms.

3. Methodology

In this section, we present the development steps of the method. We aimed at developing our algorithm to be efficient, low redundancy, and noise resilient. The proposed algorithm (LRWEEDA) consists of three distinct steps as shown in Figure 1.

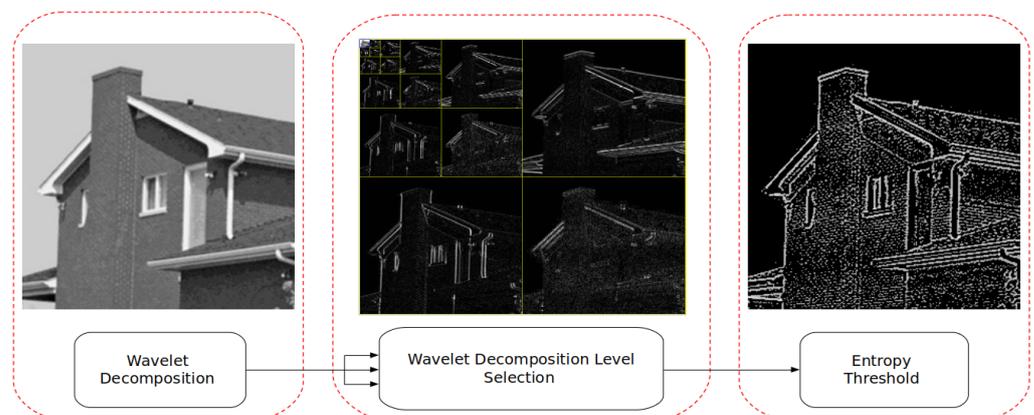


Figure 1. The overview of the proposed method.

For an image $I(x, y)$, the edges within the image can be described using gradient magnitude $|\nabla I(x, y)|$ and gradient direction γ :

$$|\nabla I(x, y)| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \quad (1)$$

$$\gamma = \angle\left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right) \quad (2)$$

With regard to Equations (1) and (2), an edge is a collection of points with a similar gradient magnitude and direction clustered within a similar region. This implied definition means that edges within a given $I(x, y)$ can be identified using three measures: image frequency to identify gradient magnitude, feature orientation to identify gradient direction, and image structure to identify clustering. We use wavelet filtering in combination with Shannon entropy to enable the identification of image frequency, feature orientation and image structure. The combination of wavelet filtering with Shannon entropy provides a simple, efficient and low redundancy method for edge detection.

Generally, entropy measures are considered unsuitable for detecting image structure such as edges [28,55]. This is due to entropy being based on the statistical properties and does not take into account the spatial properties of pixels within the image. Attempts have been made by [28–32] to overcome this by using multiple techniques including calculating localized entropy with windowed image sections and the use of relative entropy measures [35]. As a result, these methods are typically computationally intensive and their performance is sensitive to the window size used to calculate the localized entropy. Wavelet filtering enables an image field to be decomposed into individual frequency and orientation layers. In doing so, the output wavelet decomposition makes an inherent assumption of structure. Shannon entropy can be applied to each of these wavelet decomposition levels to give a quantitative assessment of image structure, or in our case, a quantitative assessment of edge structure. The following sections will explore in detail each of the steps shown in Figure 1.

3.1. Wavelet Decomposition

Wavelets have received widespread interest over the past few decades with their ability to perform spatio-temporal analysis on signals of interest [55,56]. Their mathematical properties have enabled efficient implementations for high bandwidth applications using sub-band coding techniques. The one-dimensional, (1D) Continuous Wavelet Transform (CWT) is defined as

$$W_{\psi(s,\tau)} = \int_{-\infty}^{\infty} f(x)\psi_{s,\tau}(x)dx, \quad (3)$$

where

$$\psi_{s,\tau}(x) = \frac{1}{\sqrt{s}}\psi\left(\frac{x-\tau}{s}\right), \quad (4)$$

and s and τ are called scale and translation parameters. The scale parameter enables isolation of frequency components within a signal and τ enables the isolation of scale (frequency) components within time. For image processing applications, $I(x, y)$ is defined as a discrete two-dimensional (2D) signal. To do so, Equations (3) and (4) must be represented in their discrete form and applied to $I(x, y)$ as shown in Figure 2.

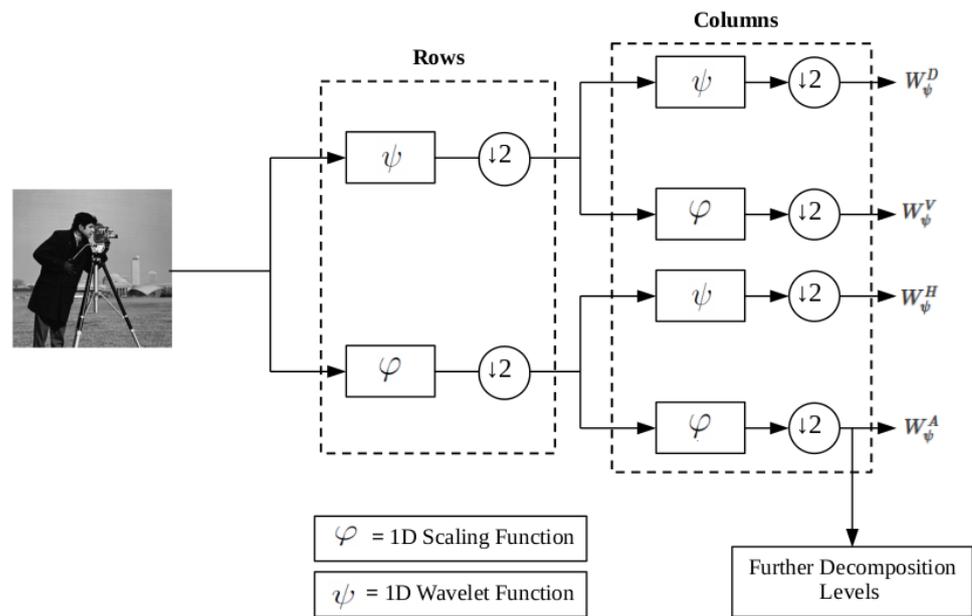


Figure 2. Application of wavelet decomposition to $I(x, y)$.

Figure 2 identifies that the 2D Discrete Wavelet Transform (DWT) is constructed through the application of cascaded 1D DWT filter banks applied to the rows and columns of $I(x, y)$, respectively. The product of the cascaded 1D DWT filter banks produces directionally sensitive wavelets that measure image gradient magnitude tuned to a particular size. Figure 2 can be described using the following equations. Here, $i = H, V, D$ and j is the decomposition level:

$$\psi^A(x, y) = \varphi(x)\varphi(y) \tag{5}$$

$$\psi^H(x, y) = \psi(x)\varphi(y) \tag{6}$$

$$\psi^V(x, y) = \varphi(x)\psi(y) \tag{7}$$

$$\psi^D(x, y) = \psi(x)\psi(y) \tag{8}$$

$$\varphi_{j,m,n}(x, y) = 2^{j/2}\varphi(2^jx - m, 2^jy - n) \tag{9}$$

$$\psi_{j,m,n}^i(x, y) = 2^{j/2}\psi^i(2^jx - m, 2^jy - n), \quad i = H, V, D \tag{10}$$

$$W_{\varphi(j_0,m,n)}^A = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)\varphi_{j_0,m,n}(x, y) \tag{11}$$

$$W_{\psi(j,m,n)}^i = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)\psi_{j,m,n}^i(x, y) \tag{12}$$

The wavelet decomposition process is well suited to real-time applications as it can be implemented in an efficient parallel way with the use of short cascaded FIR filter banks, making it computationally efficient [57].

The algorithm takes a grey scale image as the input (8 bit grey scale image with 512×512 pixels in size) and passes it through a 2D wavelet decomposition (six decomposition levels) using the Coiflet and Haar wavelet functions. In doing so, the image is broken down into individual frequency and orientation layers, isolating important image features according to their frequency and orientation within the image as shown in Figure 3.

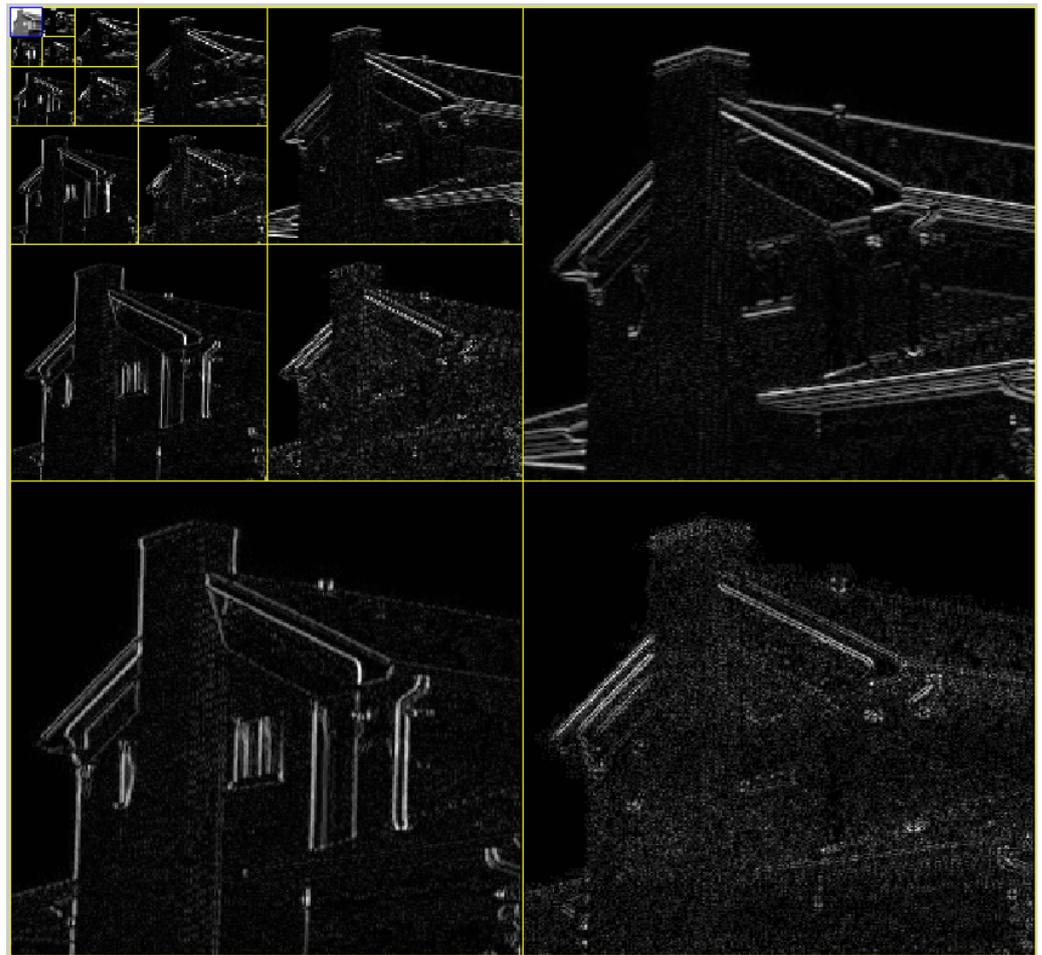


Figure 3. Wavelet decomposition applied to a test image. Six decomposition levels were applied.

Figure 3 shows that the wavelet decomposition has identified edge features of differing gradient magnitudes aligned in three different orientations: vertically, horizontally and diagonally. For each wavelet decomposition level, vertical and horizontal components are combined and normalized between zero and one to define $\mathcal{U}(j, m, n)$,

$$\mathcal{U}(j, m, n) = \text{normalise}(W_{\psi}^H(j, m, n) + W_{\psi}^V(j, m, n)) \quad (13)$$

where j represents the decomposition level. Each decomposition level can be interpreted as an image frequency band, where lower levels represent higher frequencies and higher levels represent low frequencies. Diagonal features within each level are discarded as they typically contain noisy information that will reduce edge feature quality [55].

Upon calculation of Equation (13), the algorithm successfully isolates all gradient magnitudes and directions into individual image frequency and orientation layers. This means that each decomposition level shown in Figure 3 has an assumed level of image structure. Dependent on the image input, different decomposition levels will indicate greater structure than others. Decomposition levels with greater levels of structure will represent greater value in the identification of edges. Decomposition levels with less structure offer less value in the identification of edges and can be discarded to optimize the edge detection algorithm.

Ultimately, to identify all edges within an image, all decomposition levels would be combined. However, when viewed from an optimization and redundancy viewpoint, we may be able to ignore or not include some edge features contained within specific decomposition levels resulting in only minor degradation to the final edge detection solution. This is analogous to lossy image compression schemes or message redundancy

problems [55,58]. The algorithm assumes that the decomposition level with the most structure can be assumed to have the most useful edges within it and therefore should be used for further image processing applications. To quickly measure the amount of structure in each decomposition level, we use Shannon entropy for structure measurement.

3.2. Wavelet Decomposition Level Selection

After an image has been broken down into individual scale (frequency) and orientation layers using the wavelet decomposition step, the most suitable level must be selected for further processing. Shannon entropy is used to determine the most suitable decomposition level for the detection of edges. Shannon entropy is typically unsuitable as a measure of image structure [55] as it only accounts for the statistical distribution of pixel values, as described by

$$H(I(x, y)) = - \sum_{i=0}^L p(I_i) \log p(I_i) \quad (14)$$

where p is the probability of pixel value I_i occurring within $I(x, y)$, and L is the maximum pixel value within $I(x, y)$. Shannon entropy provides a global quantitative assessment, considering only global statistical information of $I(x, y)$ while discarding the spatial distribution information. This means that two images with the same Shannon entropy value (quantitative assessment) can be completely different when qualitatively assessed by human vision [28,55]. Thus, Shannon entropy cannot be used to infer image structure. However, when Shannon entropy is used in combination with wavelet decomposition, as shown in Equation (15), Shannon entropy values can be representative of image structure (in particular edge structure within an image). To combine Shannon entropy with wavelet decomposition, $I(x, y)$ is substituted for the combined and normalized wavelet data and the Shannon entropy value is calculated at each decomposition level (j). Thus, $I(x, y) = \mathcal{U}(j, m, n)$ resulting in

$$H(\mathcal{U}(j, m, n)) = - \sum_{i=0}^L p(\mathcal{U}_i) \log p(\mathcal{U}_i) \quad (15)$$

The decomposition step successfully isolates all gradient magnitudes and directions into individual image frequency and orientation layers. It is assumed that each decomposition level has an assumed level of image structure which can be measured globally using Shannon entropy. Lower values of entropy indicate a high level of redundancy within a given dataset. In the case of our wavelet decomposition levels, a low entropy value will indicate the presence of repetitive sequences and clusters within a decomposition level, indicating that a particular level has a high level of structure. By comparing the Shannon entropy value for each of the decomposition levels, we can form a quantitative assessment of which the decomposition level has the greatest image structure. The $H(\mathcal{U}(j, m, n))$ with the lowest Shannon entropy value is expected to have the most structure and therefore the most useful edge data as shown in Figure 4.

Figure 4 contains two images: a striped black and white image and the same image with the pixels randomly shuffled. Figure 4 identifies that the first structured image generates lower entropy values. The randomly shuffled image contains the same statistical distribution as the first image; however, it generates higher entropy indicating less image structure. Without applying wavelet decomposition before the Shannon entropy calculation, both images generate the same Shannon entropy value which provides no quantitative assessment of image structure.

Wavelet decomposition level selection discards edge data that contains a low level of structure (high entropy and thus low redundancy). The use of Shannon entropy to select which decomposition levels are kept or discarded means that we are identifying which level contains the highest level of redundancy, and therefore the highest level of edge structure. It is therefore assumed that discarded decomposition levels will only contain minimal edge structure that do not contribute significantly to the definition of edge structure. This helps

enable the algorithm to provide an efficient and low redundancy result. Figure 5 provides an example of Shannon entropy values for $\mathcal{U}(j, m, n)$ when the example image (“House”) from Figure 3 is used.

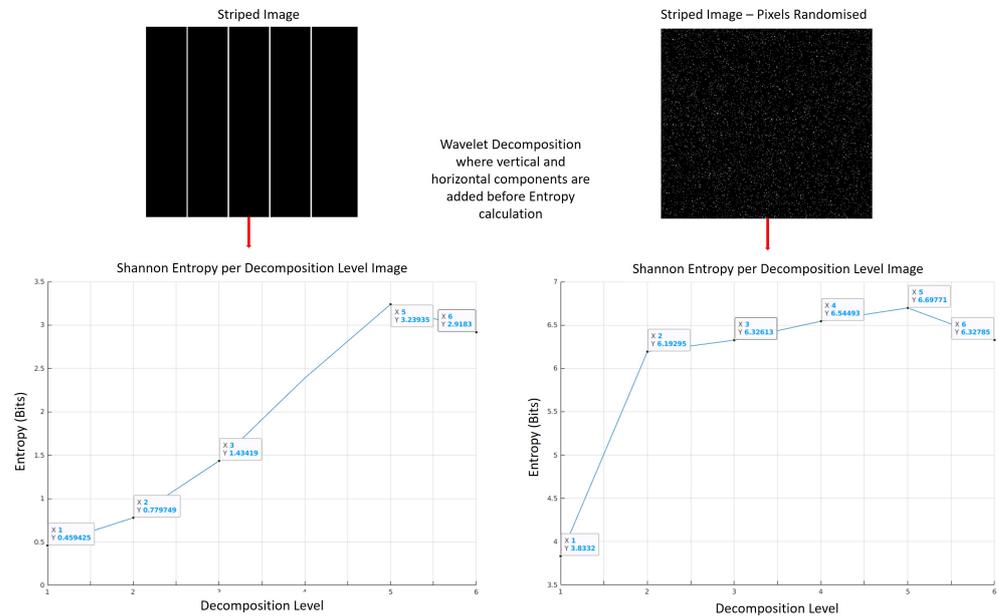


Figure 4. Global structure measurement using Shannon entropy. Both images are 512×512 pixels in size.

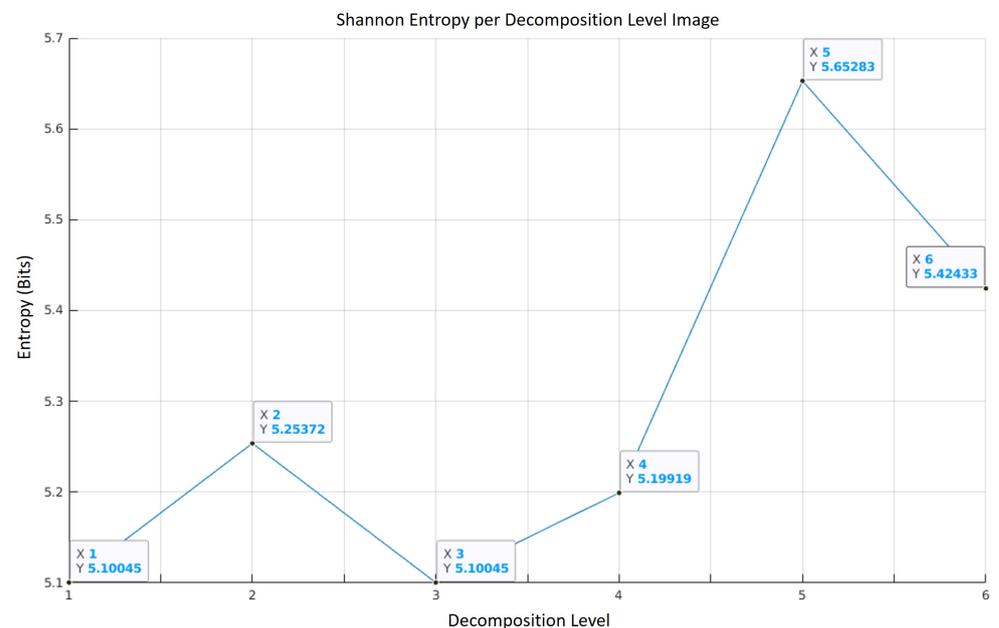


Figure 5. Entropy per decomposition level where vertical and horizontal components are added and normalised using the test image from Figure 3.

Note that we denote the optimal wavelet decomposition level as $\mathcal{U}(\beta, m, n)$. The optimal wavelet decomposition Level will be the level that has the lowest entropy value. In the case of Figure 3, $\mathcal{U}(\beta, m, n) = \mathcal{U}(1, m, n)$.

3.3. Entropy Thresholding

Once wavelet decomposition level selection is completed, we are able to identify the optimal edge detection image. Depending on the wavelet chosen for decomposition,

artifacts may be generated in the edge detection image. The generation of artifacts depends on the properties of the wavelet chosen and how it convolves with the input image. To overcome these artifacts and enhance the quality of the edge detection image, a variable threshold Λ is applied to the image:

$$\forall m, n \in \mathcal{U}(\beta, m, n)$$

$$\mathcal{U}(\beta, m, n) = \begin{cases} \mathcal{U}(\beta, m, n) & \text{if } \mathcal{U}(\beta, m, n) \geq \Lambda \\ 0 & \text{if } \mathcal{U}(\beta, m, n) < \Lambda \end{cases} \quad (16)$$

where Λ is the threshold value and varies in the range of $0 \leq \Lambda \leq 1$.

A number of methods [28,33,55] exist for the selection of the most appropriate threshold value within an image to detect edges. Typically, these methods find a threshold based on the histogram of a grey scale image which is multi-modal. Due to the wavelet decomposition used in the selection of $\mathcal{U}(\beta, m, n)$, the image we are thresholding is not multi-modal. In our case $\mathcal{U}(\beta, m, n)$, selected during wavelet decomposition level selection, will have a Gaussian like histogram distribution as shown in Figure 6.

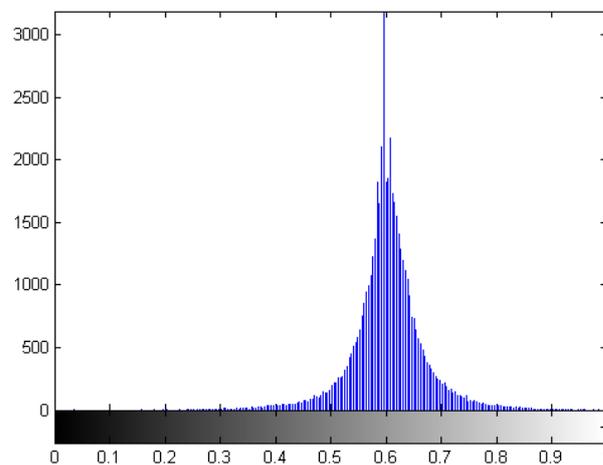


Figure 6. Pixel histogram for $\mathcal{U}(\beta, m, n)$ using the test image.

From Figure 6, we can see that pixel sequences from $I(x, y)$ that correlate well with $\psi_{\beta, m, n}^H$ and $\psi_{\beta, m, n}^V$ will result in a value closer to one. Likewise, pixel sequences from $I(x, y)$ that don't correlate well with $\psi_{\beta, m, n}^H$ and $\psi_{\beta, m, n}^V$ will result in a value closer to zero. To determine the optimal threshold, we generate a Shannon entropy curve for $\mathcal{U}(\beta, m, n)$ which is done by changing the threshold (Λ) from 0.1 to 1 in 0.1 increments and, for each increment, we use (14) to calculate the Shannon entropy value. Figure 7 shows the Shannon entropy curve for $\mathcal{U}(\beta, m, n)$ generated for the test image from the previous step.

Figure 7 shows that the optimal threshold value will exist somewhere between the inflection point ($\Lambda = 0.61$) and the minimum entropy value ($\Lambda = 1$). The inflection point provides a suitable image but will contain a number of artifacts. The artifacts reduce as the Shannon entropy value tends towards zero, however, so does the edge data. An application of the algorithm must compromise between edge detail and the number of artifacts shown in the output image. Figure 8 shows the final output using the "House" test image with a Λ of 0.635.

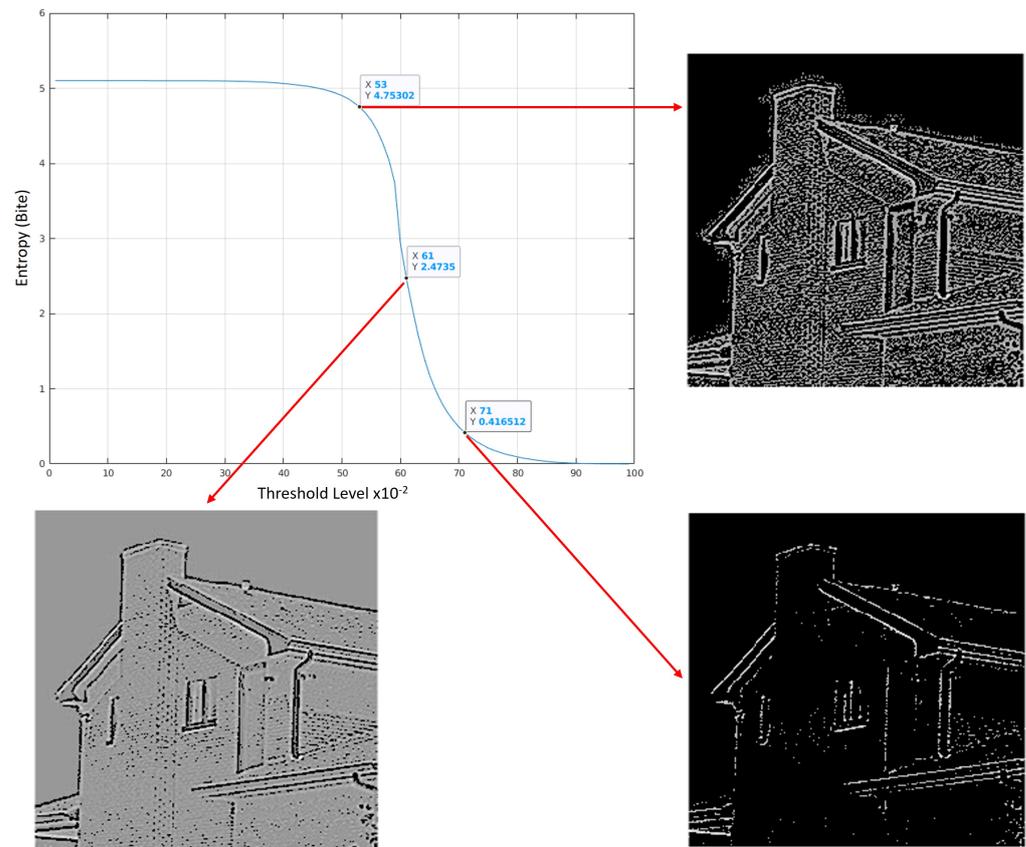


Figure 7. Shannon entropy curve for threshold selection and Coiflet wavelet. Output images are 258×258 pixels in size.



Figure 8. The resultant image (258×258 pixels) with Coiflet, $j = 1$, $\Lambda = 0.635$.

4. Results

The algorithm was tested on a small dataset of six images (Lena, House, Mandril, Peppers, Cameraman and Jetplane). The same six processing sets were used with the Canny, Prewitt, Roberts, Sobel and Zerocross edge detection algorithms. The results can be seen in Figure 9.

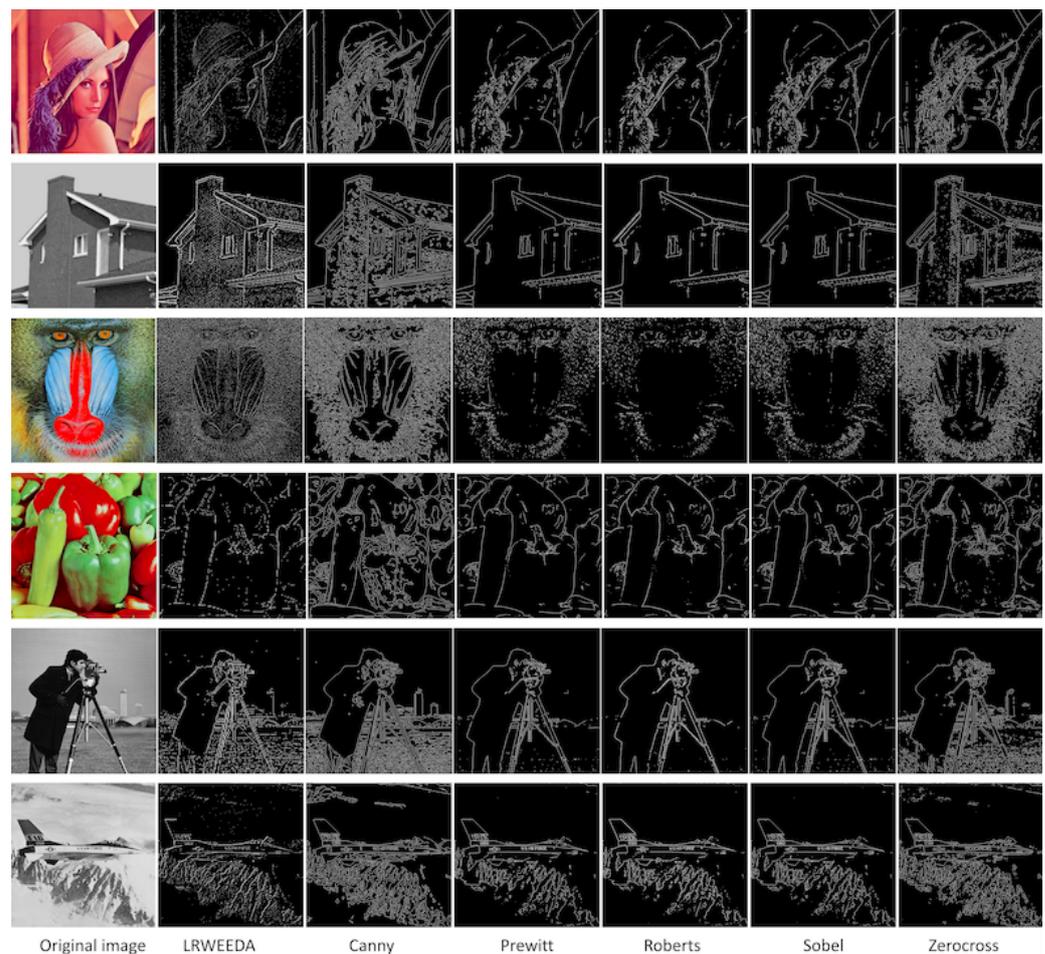


Figure 9. The original image is shown in the first column of the image grid. Other columns correspond to different edge detection algorithms, and the algorithm name is mentioned in the bottom of the column. For the LRWEEDA algorithm, the following parameters were used (from the first row to the last row): (1) “Lena” image: LRWEEDA (output image size of 256×256 pixels) using Haar, $j = 1$, $\Lambda = 0.480$; (2) “House” image: LRWEEDA (258×258 pixels) using Coiflet, $j = 1$, $\Lambda = 0.635$; (3) “Mandrill” image: LRWEEDA (258×258 pixels) using Coiflet, $j = 1$, $\Lambda = 0.520$; (4) “Peppers” image: LRWEEDA (131×131 pixels) using Coiflet, $j = 2$, $\Lambda = 0.480$; (5) “Cameraman” image: LRWEEDA (131×131 pixels) using Coiflet, $j = 2$, $\Lambda = 0.530$; and (6) “Jetplane” image: LRWEEDA (256×256 pixels) using Haar, $j = 1$, $\Lambda = 0.555$.

LRWEEDA achieves similar quality edge detection results to the Canny, Prewitt, Roberts, Sobel and Zerocross edge detection algorithms; however, it has achieved this with a significantly reduced pixel count due to the downsampling that occurs in the wavelet decomposition stage. It is important to note that, in each case, the results represent the theoretically optimal image quality that can be achieved with the lowest level of redundant pixels, due to the use of Shannon entropy to detect image structure. In each of the six images, the algorithm calculated the result in approximately 0.1 s and was only bettered in computational performance on the same hardware by the Zerocross method as shown in Figure 10.

Our algorithm provides greater performance in regions where there is a higher level of edge complexity. For instance, it is able to detect edge features within the brick work of the “House” image, the feather in the hat (“Lena” image) and the hairs on the face of the “Mandrill” image. The algorithm does not break down in areas of high edge complexity as the algorithm does not include a point clustering step like the Canny edge detection algorithm. This results in LRWEEDA generating false points within the image; however, it reduces the number of false edge artifacts similar to those seen in the brick work of the

Canny processed image (see second row, third column of Figure 9). Our algorithm was able to achieve this due to the use of Shannon entropy which enables a simplified threshold technique to be used to reduce false points and therefore noise.

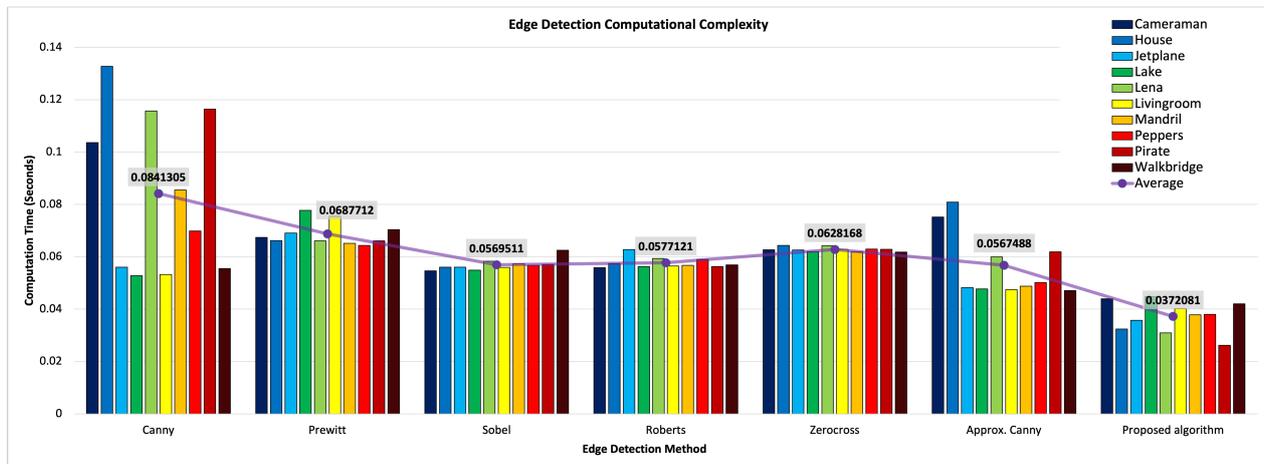


Figure 10. Average computation time for different edge detection algorithms. A dataset of ten images was used for the analysis.

4.1. Computational Efficiency

The computational efficiency was tested against five edge detection algorithms using a dataset of ten standard images (all images were unsigned 8 bit grey scale 512×512 pixel images). To test computational efficiency, the execution time for each edge detection algorithm was recorded. The proposed algorithm is shown in Algorithm 1.

Algorithm 1: LRWEEDA edge detection

Input: Image, $I_{in}(x, y)$
Output: Image, $I_{out}(x, y)$
Data: Testing set x

- 1 **forall** $I_{in}(x, y)$ **do**
- 2 Wavelet decomposition of Image down to level 6
- 3 Combine Vertical and Horizontal components
- 4 Normalize for each decomposition level
- 5 Calculate Shannon entropy for each of the combined decomposition levels
- 6 Find the combined decomposition level with the lowest Shannon entropy value
- 7 Find all values less than the threshold value within the decomposition level with the lowest Shannon entropy value and set to zero
- 8 **return** the results

Figure 10 shows that the proposed algorithm executes faster than existing existing edge detection techniques (i.e., Canny, Prewitt, Sobel, Roberts, Zerocross and approximate Canny). The Canny edge detector method takes approximately double the computational time required by the proposed method. The proposed algorithm is designed for scalability and parallel stream processing. Implementation of LRWEEDA on an ASIC, GPU or FPGA medium would result in significantly reduced computation times and low memory usage sufficient for real world applications. The edge detection output for each edge detection method is a 512×512 pixel image except for LRWEEDA, which produces a reduced pixel output dependent on the decomposition level selected (17). Edge detector algorithms are typically used as the first step in a larger image processing task. LRWEEDA’s ability to generate an edge detection image with a reduced pixel output means that processing

time for further image processing stages can be significantly reduced by applying pixel reduction

$$\kappa = \frac{N}{2^{2j}} \tag{17}$$

where N are the number of pixels in the original input image and j is the selected decomposition level.

4.2. Noise Resilience

The ability for the proposed method to quantitatively measure structure within the image means that it is inherently more resilient to the effects of noise. To determine the resilience of LRWEEDA to noise, White Gaussian Noise (WGN) with a Probability Density Function (PDF) of

$$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \tag{18}$$

where z represents the grey level, μ the mean value and σ the standard deviation was added to two images. WGN with a μ of 0 and variance (σ^2) of 0.00001, 0.0001, 0.001, 0.01, 0.1 and 1 was added to both images before being processed with the Canny and LRWEEDA algorithms. An artificial square image (first row, first column of Figure 11), a real world image (third row, first column of Figure 11), and two images from BSDS500 dataset [9] (fifth and seventh row, first column of Figure 11) were used as the four images. All images were 8 bit gray scale 1920×1920 pixel images, and the results can be found in Figure 11.

In all image test cases, as the noise levels increased (increasing σ), LRWEEDA used high decomposition levels (j) as $\mathcal{U}(\beta, m, n)$. This is expected as increasing noise levels result in an increase of noise away from the higher frequencies and towards the lower image frequencies. The increased noise at the lower image frequencies therefore begins to reduce the amount of structure now appearing at lower decomposition levels (lower j values) which raises their entropy value. Therefore, LRWEEDA can be used to find the image frequencies (using the Wavelet scale function) which contains the lowest level of noise corruption whilst containing the highest level of edge structure. This behaviour within the algorithm is closely linked to wavelet properties which have been heavily exploited in image processing for image denoising processes [59].

Using the artificial square image (see first row, first column of Figure 11), LRWEEDA was able to successfully detect edge data for all values of σ^2 ; however, the quality of detected edge data decreased as σ^2 increased. In contrast, Canny showed degraded detection of edge data from $\sigma^2 = 0.00001$ to 0.001 and shows severely degraded edge data for $\sigma^2 > 0.001$. Whilst LRWEEDA did have degradation to the resolution of the edge image as σ^2 increased, unlike Canny, it did not have a significant increase in edge artifacts caused by the false detection of edges due to noise. This occurs due to LRWEEDA using higher decomposition levels (j) as $\mathcal{U}(\beta, m, n)$ which results in a low resolution edge image; however, LRWEEDA is able to determine which $\mathcal{U}(j, m, n)$ has the lowest level of noise but the highest level of structure (lowest entropy value) and use this as $\mathcal{U}(\beta, m, n)$. Canny, on the other hand, relies on its initial blurring function to remove and average out the noise within an image which is not as effective in the removal of WGN. Canny is further affected by the clustering of points to give edge continuity. As a result of the clustering component of the Canny algorithm, edge artifacts increase as false detection of edges caused by the noise are clustered together creating a further reduction in edge detection performance.

Applied to the other images, LRWEEDA was able to detect edge data for all values of σ^2 ; however, the number and quality of detected edge data significantly decreased as σ^2 increased. In contrast, Canny showed degraded detection of edge data from $\sigma^2 = 0.00001$ to 0.001 and shows severely degrade edge data for $\sigma^2 > 0.001$. LRWEEDA was once again able to determine which $\mathcal{U}(j, m, n)$ had the lowest level of noise but the highest level of structure (lowest entropy value) and used this as $\mathcal{U}(\beta, m, n)$. Canny, on the other hand, relies on its initial blurring function to remove and average out the noise within an image which is not as effective in the removal of WGN; furthermore, Canny was also hindered

by its edge clustering properties. It is interesting to note that, in the case of the least amount of noise ($\sigma^2 = 0.00001$) (third row, second column and fourth row, second column of Figure 11), LRWEEDA demonstrates a clear improvement in edge detection quality compared to Canny. In particular, LRWEEDA clearly identifies the building and tower structural features with minimal artifacts compared to the Canny algorithm.

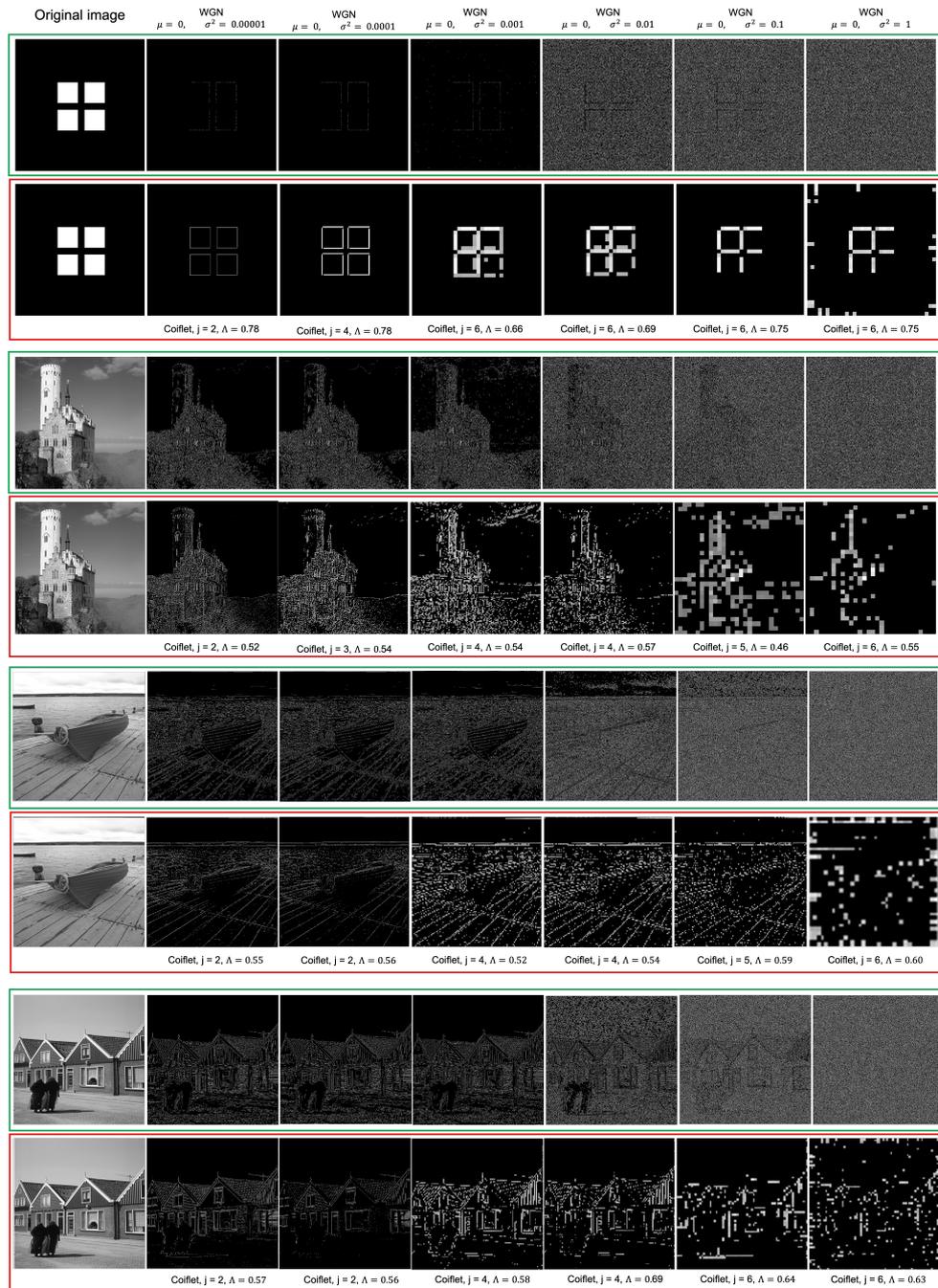


Figure 11. The original image is shown in the first column of the image grid. Other columns correspond to different noise levels. The first, third, fifth, and seventh rows correspond to Canny processed images (grouped in green) and the second, fourth, sixth and eighth rows correspond to LRWEEDA processed images (grouped in red). The corresponding LRWEEDA parameters are shown below the images of red groups. The LRWEEDA processed output images sizes are 483×483 pixels when $j = 2$; 244×244 pixels when $j = 3$; 124×124 pixels when $j = 4$; 64×64 pixels when $j = 5$ and 34×34 pixels when $j = 6$.

4.3. Performance against Standard Edge Detection Metrics

We calculated the Dice coefficient of the proposed algorithm using a synthetic image (see Figure 12).

- The Dice similarity coefficient (DSC), also called the F_1 score, is used to evaluate the similarity of two samples. It is calculated as the harmonic mean of the precision and the recall as

$$DSC = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{19}$$

or

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{20}$$

where TP is the true positive, FP is the false positive and FN is the false negative.

Dice coefficient returns a value between 0 and 1, where 1 is the highest similarity. To evaluate the edge detection performance, a higher Dice coefficient means more accuracy for the prediction of the edges compared to the ground truth images. The Dice coefficients were calculated for the resultant edge image of each algorithm.

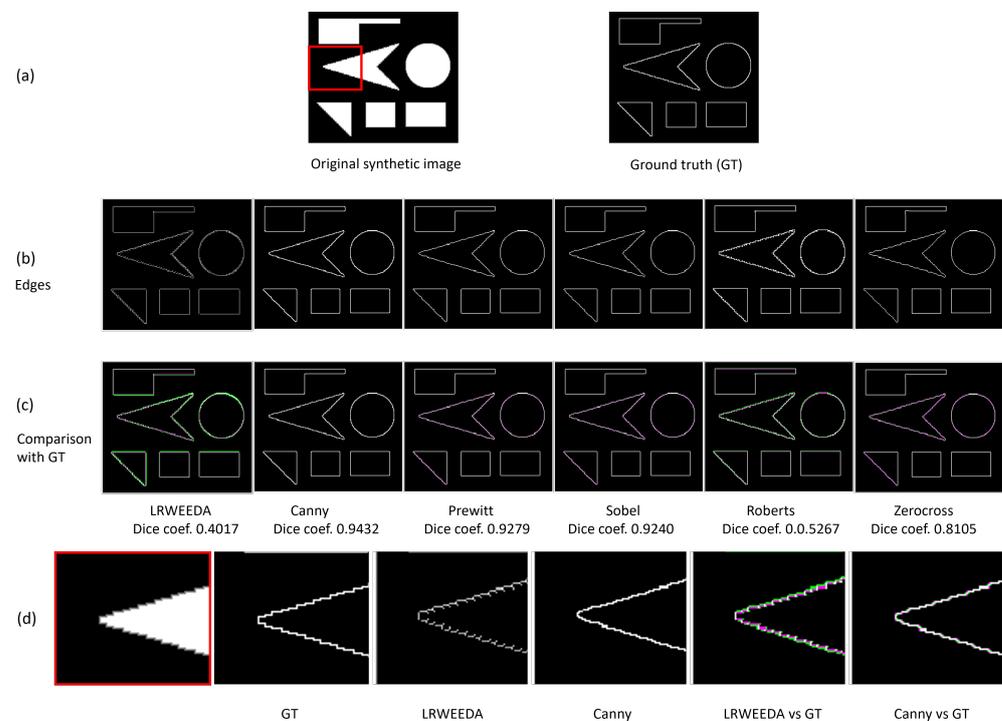


Figure 12. A synthetic image was used to calculate the Dice coefficient of the proposed algorithm. The original synthetic image (270×238 pixels) and the ground truth edges (270×238 pixels) are shown in (a). The edges obtained by LRWEEDA, Canny, Prewitt, Sobel, Roberts and Zero cross algorithms are shown from left to right in (b). In (b), all the images are 270×238 pixels. The edges were compared with the ground truth in (c). The edges calculated by each algorithm and the ground truth edges are shown in green and purple, respectively. The overlap between the calculated edges and the ground truth is shown in white. The Dice coefficient for each algorithm is shown under each image. The red color box in (a) is enlarged for the analysis purpose in (d).

As shown in Figure 12c, LRWEEDA shows the lowest Dice coefficient among the tested algorithms. However, it detects all the edges with additional local image details. In Figure 12d, an enlarged section of the original synthetic image is analyzed. As can be seen from the red bordered image, the original image has saw tooth edges. The proposed algorithm is sensitive to minor local image details and hence shows more local details in the resultant image (LRWEEDA’s ability to detect complex local edges (Figure 9) was

discussed in the beginning of this section). When compared to the ground truth, these local edges are considered as noise as they were not represented in the ground truth. Therefore, in this analysis, LRWEEDA shows a lower Dice coefficient.

We further analyzed the performance of the proposed algorithm against four standard edge detection metrics. In addition to Dice coefficient, we tested our algorithm with a boundary F1 score (BFS) [60], Jaccard coefficient (JC) [60], and Pratt's figure of merit (FOM) [61] metrics to analyze the performance. The metrics are summarized below.

- The Boundary F1 score is defined as the harmonic mean (F1-measure) of the precision and recall values which measure the matching weight for the predicted boundary and the ground truth boundary, as

$$BFS = 2 \cdot \text{precision} \cdot \text{recall} / (\text{recall} + \text{precision}). \quad (21)$$

- The Jaccard coefficient for two sets is defined as the size of the intersection of the two sets divided by the size of their union as

$$JC = \frac{TP}{(TP + FP + FN)}. \quad (22)$$

- Pratt's FOM uses Euclidean distance to compare two edge images [60]. It multiplies a scale factor α to the Euclidean distance calculated between the two images to penalize displaced edges, as

$$\text{Pratt's FOM} = \frac{1}{\max(I_A, I_B)} \sum_{i=1}^{I_A} \frac{1}{1 + \alpha d_i^2} \quad (23)$$

where I_A , I_I , and d are, respectively, the detected edges, the ideal edges, the distance between the actual and the ideal edges.

A test dataset of 20 images were selected from the BSDS500 dataset. The performance metrics of the dataset are compared in Figure 13. As shown in Figure 13, LRWEEDA and Canny metrics are closer to each other. Overall, LRWEEDA, Canny and Zero-cross algorithms show roughly similar performance. LRWEEDA does not show superior performance over the other algorithms in terms of these performance metrics. However, its performance is within the range of the Canny and Zero-cross while representing complex local edges.

We ran all the experiments in MATLAB using a 2.60 GHz laptop computer. The MATLAB functions of 'canny', 'approx_canny', 'prewitt', 'sobel', 'roberts', and 'zerocross' were used for the comparison. Four experiments were conducted to validate the findings with the following image sets:

- Qualitative results of the proposed algorithm were obtained and compared with similar edge detection algorithms (Figure 9)
- Ten images were used to calculate the average processing times of the algorithms (Figure 10).
- Noise resilience of the proposed algorithm was analyzed by using four images and compared with Canny (Figure 9).
- Performance against standard edge detection metrics were calculated using a synthetic image and 20 images from BSDS500 dataset (Figures 12 and 13).

Overall, the performance analysis in Figures 10 and 13 show that LRWEEDA's edge detection performance is within the performance range of popular edge detection algorithms while its processing time is lower compared to other similar algorithms.

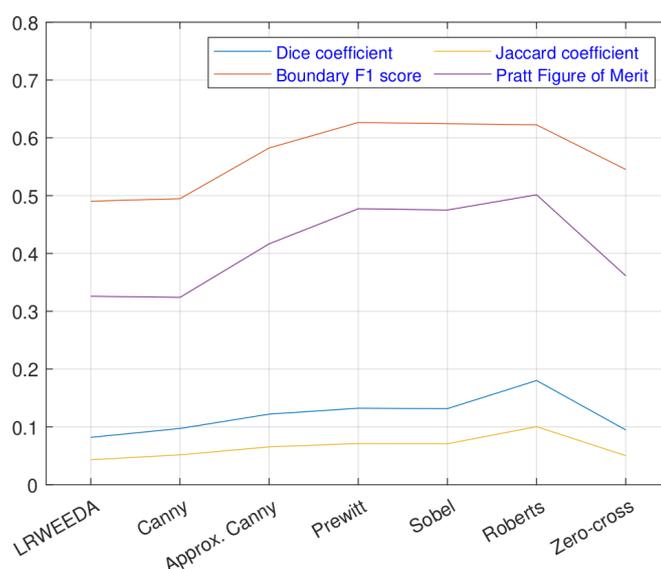


Figure 13. Comparison of the performance metrics.

5. Conclusions

We have developed a method for edge detection using a combination of the Wavelet transform, Shannon entropy and thresholding. The assumption of structure within a wavelet decomposition enabled the use of Shannon entropy as a measure of global image structure. Our work has resulted in a new edge detection method, known as the Low Redundancy Wavelet Entropy Edge Detection Algorithm that has been developed as an efficient, low redundancy, robust edge detection algorithm well suited to real-time image processing applications. Our analysis shows that LRWEEDA can generate similar or better edge detection results with significantly less output data being generated. Experiments were conducted on six standard image processing sets and we compared the performance of LRWEEDA to five popular edge detection algorithms. High resolution images with artificial and real world scenes were used to assess the noise resilience of the proposed algorithm.

For future work, we are focusing on further reducing the processing time of the algorithm and improving the threshold selection on the Shannon entropy curve. The low processing time of the proposed algorithm should be more appealing for high speed edge detection applications.

Author Contributions: Conceptualization, T.S., Y.T., A.L. and J.C.; methodology, Y.T. and T.S.; software, T.S., Y.T. and A.G.P.; validation, T.S., Y.T. and A.G.P.; formal analysis, T.S. and Y.T.; investigation, Y.T. and T.S.; writing—original draft preparation, T.S. and Y.T.; writing—review and editing, Y.T., T.S., A.G.P., A.L. and J.C.; visualization, T.S. and Y.T.; supervision, A.L. and J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kim, C.M.; Hong, E.J.; Park, R.C. Chest X-ray Outlier Detection Model using Dimension Reduction and Edge Detection. *IEEE Access* **2021**, *9*, 86096–86106. [[CrossRef](#)]
2. Dai, W.; Na, J.; Huang, N.; Hu, G.; Yang, X.; Tang, G.; Xiong, L.; Li, F. Integrated edge detection and terrain analysis for agricultural terrace delineation from remote sensing images. *Int. J. Geogr. Inf. Sci.* **2020**, *34*, 484–503. [[CrossRef](#)]
3. Gafton, P.; Maraz, E. 2D image relighting with image-to-image translation. *arXiv* **2020**, arXiv:2006.07816.
4. Li, M.; Lin, Z.; Mech, R.; Yumer, E.; Ramanan, D. Photo-sketching: Inferring contour drawings from images. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 7–11 January 2019; pp. 1403–1412.
5. Khan, N.H.; Adnan, A. Urdu optical character recognition systems: Present contributions and future directions. *IEEE Access* **2018**, *6*, 46019–46046. [[CrossRef](#)]
6. Hirz, M.; Walzel, B. Sensor and object recognition technologies for self-driving cars. *Comput.-Aided Des. Appl.* **2018**, *15*, 501–508. [[CrossRef](#)]
7. Ziou, D.; Tabbone, S.; others. Edge detection techniques-an overview. *Pattern Recognit. Image Anal. C/C Raspoznavaniye Obraz. Anal. Izobr.* **1998**, *8*, 537–559.
8. Nadernejad, E.; Sharifzadeh, S.; Hassanpour, H. Edge detection techniques: Evaluations and comparison. *Appl. Math. Sci.* **2008**, *2*, 1507–1520.
9. Arbelaez, P.; Maire, M.; Fowlkes, C.; Malik, J. Contour Detection and Hierarchical Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 898–916. [[CrossRef](#)]
10. Xie, S.; Tu, Z. Holistically-nested edge detection. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 1395–1403.
11. Orujov, F.; Maskeliūnas, R.; Damaševičius, R.; Wei, W. Fuzzy based image edge detection algorithm for blood vessel detection in retinal images. *Appl. Soft Comput.* **2020**, *94*, 106452. [[CrossRef](#)]
12. Canny, J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, PAMI-8, 679–698. [[CrossRef](#)]
13. Prewitt, J.M. Object enhancement and extraction. *Pict. Process. Psychopictorics* **1970**, *10*, 15–19.
14. Roberts, L.G. Machine Perception of Three-Dimensional Soups. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 1963.
15. Kanopoulos, N.; Vasanthavada, N.; Baker, R.L. Design of an image edge detection filter using the Sobel operator. *IEEE J. Solid-State Circuits* **1988**, *23*, 358–367. [[CrossRef](#)]
16. Basu, M. Gaussian-based edge-detection methods—a survey. *IEEE Trans. Syst. Man Cybern. Part C* **2002**, *32*, 252–260. [[CrossRef](#)]
17. Liu, Y.; Cheng, M.M.; Hu, X.; Wang, K.; Bai, X. Richer convolutional features for edge detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; pp. 3000–3009.
18. Poma, X.S.; Riba, E.; Sappa, A. Dense extreme inception network: Towards a robust cnn model for edge detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 1923–1932.
19. Gong, X.Y.; Su, H.; Xu, D.; Zhang, Z.T.; Shen, F.; Yang, H.B. An overview of contour detection approaches. *Int. J. Autom. Comput.* **2018**, *15*, 656–672. [[CrossRef](#)]
20. Siuzdak, J. A single filter for edge detection. *Pattern Recognit.* **1998**, *31*, 1681–1686. [[CrossRef](#)]
21. Peli, T.; Malah, D. A study of edge detection algorithms. *Comput. Graph. Image Process.* **1982**, *20*, 1–21. [[CrossRef](#)]
22. Maini, R.; Aggarwal, H. Study and comparison of various image edge detection techniques. *Int. J. Image Process. (IJIP)* **2009**, *3*, 1–11.
23. Waghule, D.R.; Ochawar, R.S. Overview on edge detection methods. In Proceedings of the 2014 International Conference on Electronic Systems, Signal Processing and Computing Technologies (ICESC), Nagpur, India, 9–11 January 2014; pp. 151–155.
24. Joshi, S.R.; Koju, R. Study and comparison of edge detection algorithms. In Proceedings of the 2012 Third Asian Himalayas International Conference on Internet (AH-ICI), Kathmandu, Nepal, 23–25 November 2012; pp. 1–5.
25. Torre, V.; Poggio, T.A. On Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, PAMI-8, 147–163. [[CrossRef](#)]
26. Deriche, R. Using Canny's criteria to derive a recursively implemented optimal edge detector. *Int. J. Comput. Vis.* **1987**, *1*, 167–187. [[CrossRef](#)]
27. Van Vliet, L.J.; Young, I.T.; Verbeek, P.W. Recursive Gaussian derivative filters. In Proceedings of the Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170), Brisbane, QLD, Australia, 20 August 1998; Volume 1, pp. 509–514.
28. Pal, N.R.; Pal, S.K. Entropic thresholding. *Signal Process.* **1989**, *16*, 97–108. [[CrossRef](#)]
29. Pun, T. Entropic thresholding, a new approach. *Comput. Graph. Image Process.* **1981**, *16*, 210–239. [[CrossRef](#)]
30. Shiozaki, A. Edge extraction using entropy operator. *Comput. Vision, Graph. Image Process.* **1986**, *36*, 1–9. [[CrossRef](#)]
31. Gull, S.; Skilling, J. Maximum entropy method in image processing. *IEE Proc. Commun. Radar Signal Process.* **1984**, *131*, 646–659. [[CrossRef](#)]
32. El-Sayed, M.A. A new algorithm based entropic threshold for edge detection in images. *arXiv* **2012**, arXiv:1211.2500.
33. Kapur, J.N.; Sahoo, P.K.; Wong, A.K. A new method for gray-level picture thresholding using the entropy of the histogram. *Comput. Vision Graph. Image Process.* **1985**, *29*, 273–285. [[CrossRef](#)]

34. Yang, C. A new operator for detecting edges in images based on modified Tsallis entropy. In Proceedings of the 2011 International Conference on Consumer Electronics, Communications and Networks (CECNet), Xianning, China, 16–18 April 2011; pp. 4671–4674. [\[CrossRef\]](#)
35. Chang, C.I.; Chen, K.; Wang, J.; Althouse, M.L. A relative entropy-based approach to image thresholding. *Pattern Recognit.* **1994**, *27*, 1275–1289. [\[CrossRef\]](#)
36. Medina-Carnicer, R.; Madrid-Cuevas, F.J.; Fernández-García, N.; Carmona-Poyato, A. Evaluation of global thresholding techniques in non-contextual edge detection. *Pattern Recognit. Lett.* **2005**, *26*, 1423–1434. [\[CrossRef\]](#)
37. Shannon, C.E. A mathematical theory of communication. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* **2001**, *5*, 3–55. [\[CrossRef\]](#)
38. Singh, B.; Singh, A.P. Edge Detection in Gray Level Images based on the Shannon Entropy. *J. Comput. Sci.* **2008**, *3*, 186–191. [\[CrossRef\]](#)
39. Li, J. A Wavelet Approach to Edge Detection. Master's Thesis, Sam Houston State University, Huntsville, TX, USA, 2003.
40. Liu, W.; Ma, Z. Wavelet image threshold denoising based on edge detection. In Proceedings of the IMACS Multiconference on Computational Engineering in Systems Applications, Beijing, China, 4–6 October 2006; Volume 1, pp. 72–78.
41. Schmeelk, J. Wavelet transforms and edge detectors on digital images. *Math. Comput. Model.* **2005**, *41*, 1469–1478. [\[CrossRef\]](#)
42. Shih, M.Y.; Tseng, D.C. A wavelet-based multiresolution edge detection and tracking. *Image Vis. Comput.* **2005**, *23*, 441–451. [\[CrossRef\]](#)
43. Siddique, J.; Barner, K.E. Wavelet-based multiresolution edge detection utilizing gray level edge maps. In Proceedings of the ICIP 98, 1998 International Conference on Image Processing, Chicago, IL, USA, 7 October 1998; Volume 2, pp. 550–554.
44. Zhang, L.; Bao, P. Edge detection by scale multiplication in wavelet domain. *Pattern Recognit. Lett.* **2002**, *23*, 1771–1784. [\[CrossRef\]](#)
45. Jiang, W.; Lam, K.M.; Shen, T.Z. Efficient edge detection using simplified Gabor wavelets. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2009**, *39*, 1036–1047. [\[CrossRef\]](#)
46. Hao, Y.; Changshun, L.; Lei, P. An improved method of image edge detection based on wavelet transform. In Proceedings of the 2011 IEEE International Conference on Computer Science and Automation Engineering (CSAE), Shanghai, China, 10–12 June 2011; Volume 3, pp. 678–681.
47. Elaraby, A.E.A.; Owny, E.; Ahmed, H.B.; Heshmat, M.; Hassaballah, M.; Rardy, A.S.A. A Novel Algorithm for Edge Detection of Noisy Medical Images. *Int. J. Signal Process. Image Process. Pattern Recognit.* **2013**, *6*, 365–374. [\[CrossRef\]](#)
48. Xishan, T. A Novel Image Edge Detection Algorithm based on Prewitt Operator and Wavelet Transform. *Int. J. Adv. Comput. Technol.* **2012**, *4*, 73–82.
49. Srivastava, G.; Verma, R.; Mahrishi, R.; Rajesh, S. A novel wavelet edge detection algorithm for noisy images. In Proceedings of the ICUMT '09, International Conference on Ultra Modern Telecommunications Workshops, St. Petersburg, Russia, 12–14 October 2009; pp. 1–8. [\[CrossRef\]](#)
50. Wu, Y.; He, Y.; Cai, H. Optimal threshold selection algorithm in edge detection based on wavelet transform. *Image Vis. Comput.* **2005**, *23*, 1159–1169. [\[CrossRef\]](#)
51. Vetterli, M.; Kovacevic, J. *Wavelets and Subband Coding*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1995.
52. Bertasius, G.; Shi, J.; Torresani, L. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; pp. 4380–4389.
53. Shen, W.; Wang, X.; Wang, Y.; Bai, X.; Zhang, Z. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; pp. 3982–3991.
54. He, J.; Zhang, S.; Yang, M.; Shan, Y.; Huang, T. Bi-directional cascade network for perceptual edge detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 3828–3837.
55. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 3rd ed.; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 2006.
56. Graps, A. An introduction to wavelets. *IEEE Comput. Sci. Eng.* **1995**, *2*, 50–61. [\[CrossRef\]](#)
57. Chakrabarti, C.; Vishwanath, M. Efficient realizations of the discrete and continuous wavelet transforms: From single chip implementations to mappings on SIMD array computers. *IEEE Trans. Signal Process.* **1995**, *43*, 759–771. [\[CrossRef\]](#)
58. Nelson, M.; Gailly, J.L. *The Data Compression Book*; M&T Books: New York, NY, USA, 1996; Volume 2.
59. Wei, D.; Rajashekar, U.; Bovik, A.C. 3.4—Wavelet Denoising for Image Enhancement. In *Handbook of Image and Video Processing*, 2nd ed.; Bovik, A., Ed.; Communications, Networking and Multimedia; Academic Press: Burlington, MA, USA, 2005; pp. 157–165. [\[CrossRef\]](#)
60. Csurka, G.; Larlus, D.; Perronnin, F.; Meylan, F. What is a good evaluation measure for semantic segmentation? *Bmvc* **2013**, *27*, 10–5244.
61. Abdou, I.E.; Pratt, W.K. Quantitative design and evaluation of enhancement/thresholding edge detectors. *Proc. IEEE* **1979**, *67*, 753–763. [\[CrossRef\]](#)