

Received June 11, 2020, accepted June 23, 2020, date of publication June 25, 2020, date of current version July 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3005050

# Pixel Intensity-Based Contrast Algorithm (PICA) for Image Edges Extraction (IEE)

ABBAS M. AL-GHAILI<sup>1</sup>, HAIROLADENAN KASIM<sup>2</sup>, NAIF MOHAMMED AL-HADA<sup>3,4</sup>, MARINI OTHMAN<sup>1</sup>, MUNEEB AZIZ SALEH<sup>4</sup>, AND WANG JIHUA<sup>3</sup>

<sup>1</sup>Institute of Informatics and Computing in Energy (IICE), Universiti Tenaga Nasional (UNITEN), Kajang 43000, Malaysia

<sup>2</sup>College of Computing and Informatics (CCI), Universiti Tenaga Nasional (UNITEN), Kajang 43000, Malaysia

<sup>3</sup>Shandong Key Laboratory of Biophysics, Institute of Biophysics, Dezhou University, Dezhou 253023, China

<sup>4</sup>Faculty of Engineering, School of Chemical and Energy Engineering, Universiti Teknologi Malaysia (UTM), Johor Bahru 81310, Malaysia

Corresponding authors: Abbas M. Al-Ghaili (abbasghaili@yahoo.com) and Naif Mohammed Al-Hada (naifalhada@yahoo.com)


This work was supported in part by the Tenaga Nasional Berhad (TNB), and in part by the Universiti Tenaga Nasional (UNITEN) under Grant UNITEN/RMC(BOLD)/1/14/AL/2020/7 and Grant RJO10517919/iRMC/Publication, through Internal Research Opex.

**ABSTRACT** In this paper, images' pixels are exploited to extract objects' edges. This paper has proposed a Pixel Intensity based Contrast Algorithm (PICA) for Image Edges Extraction (IEE). This paper highlights three contributions. Firstly, IEE process is fast and PICA has less computation time when processing different images' sizes. Secondly, IEE is simple and uses a  $2 \times 4$  mask which is different from other masks where it doesn't require *while*-loop(s) during processing images. Instead, it has adopted an *if*-conditional procedure to reduce the code complexity and enhance computation time. That is, the reason why this design is faster than other designs and how it contributes to IEE will be explained. Thirdly, design and codes of IEE and its mask are available, made an open source, and in-detail presented and supported by an interactive file; it is simulated in a video motion design. One of the PICA's features and contributions is that PICA has adopted to use less *while*-loop(s) than traditional methods and that has contributed to the computation time and code complexity. Experiments have tested 526 samples with different images' conditions e.g., inclined, blurry, and complex-background images to evaluate PICA's performance in terms of computation time, enhancement rate for processing a single image, accuracy, and code complexity. By comparing PICA to other research works, PICA consumes 5.7 mS to process a single image which is faster and has less code complexity by  $u \times u$ . Results have shown that PICA can accurately detect edges under different images' conditions. Results have shown that PICA has enhanced computation time rate for processing a single image by 92.1% compared to other works. PICA has confirmed it is accurate and robust under different images' conditions. PICA can be used with several types of images e.g., medical images and useful for real-time applications.

**INDEX TERMS** Adaptive thresholding, edge extraction, pixel intensity, pixels contrast.

## I. INTRODUCTION

Images have many elements which include important details and information. One of these items is the pixel of which the digital image consists. Image's pixels are exploited in order to help find edges of objects and regions inside digital images. Pixels are considered as a very useful tool that helps discover the borders between regions by verifying their intensities, color, and/ or values' variation. A digital image has been involved in a wide variety of applications. It has been efficiently and largely exploited by many research studies

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Tang .

from different fields. As for example, Digital Image Processing (DIP) has been used to carry out a real-time response for security purposes due to its images high resolution(s) by exploiting pixel intensities and variations for a varied diversity of purposes some of which are extensively and technically described and reviewed in [1]–[14].

Usually, digital images include many items on which different image processes are applied. Each image's item or process contains a different degree of importance depending on feature(s) to be obtained from processing that item. One of the important considerations in regard to the pixel item is how to accurately process each pixel with consideration of 4 or 8 neighbors. In this article, image's pixels and their connected

neighbors are treated in order to accurately extract edges of objects and regions inside digital images. Meaning, pixel contrast and pixel intensities variations in binarized images by focusing on 1's and 0's values are exploited.

There is a number of correlated images' processes one of which is edge detection. Edge detection has been considered by many digital images processes and therefore many related fields and image-based applications have effectively and efficiently exploited such a process due to its importance and features it has. There are a lot of information and details an image's edges have. These details provide image processes with ability to apply further processes once edges have been detected and extracted. The way those edges are extracted is, the highly accurate and robust further processes are. Examples of applications which mainly might rely on an image's edges could be medical images [15], [16], machine vision [17], motion detection for tracking purposes [18], smart vehicles technologies [19]–[23], safety applications [11], [24], and so on. For those applications, different types of techniques have been used utilizing edges detection for example, vertical edges [25], text analysis [18], and many others.

There are several issues still need to be enhanced for those methods in order to increase performance quality. For example, computation time is important for several applications. Additionally, code complexity is a considerable factor to make the proposed method as simple as could. Accurately detected edges will contribute much to edge detection process and further image's processes. In this research work, computation time while processing an image is aimed to be short and edges' detection process is aimed to be accurate. Code complexity is usually influenced by the computation time for most cases and scenarios. This is evaluated thru Results and Discussion section.

A Pixel Intensity-based Contrast Algorithm (PICA) for Image Edges Extraction (IEE) is presented and in detail discussed in this paper. PICA aims to extract edges from complex-background images using a simple edges' detector with an accurate process.

Another important issue related to edges' detection is that some techniques which use global thresholding or those that do not efficiently consider regional and neighboring pixels well are applied. Thus, such an issue will affect further image's processes and the whole performance as well in case most of foreground and background have not been carefully considered. Therefore, in this research study, to detect and extract edges efficiently, it is proposed to process images using adaptive thresholding to binarize image [26], [27].

Current applied mask-based techniques e.g.,  $3 \times 3$  have been used [28]. In this research study, the PICA has proposed and applied a mask with two different sizes, which are: two-by-four designed to process vertical edges and four-by-two designed to process horizontal edges, respectively. In this type of mask-based technique, two pixels either vertically or horizontally will be processed at once each time the proposed mask is moving thru an image's pixels. The aim that

PICA is using such a technique is to enhance edge detection's computation time. Additionally, the related code complexity should have enhanced the simplicity of the code structure.

The organization of paper is described as follows: In the Section II, literature review is provided. Section III describes in detail the proposed Pixel Intensity-based Contrast Algorithm (PICA) for Image Edges Extraction (IEE) process. In Section, IV, Experimental Setup and Preparation are discussed. This is followed by Results and Discussion in Section V. After this, Conclusion is depicted in Section VI.

## II. LITERATURE REVIEW

Edge extraction (EE) is a key process for a lot of related applications and digital image processes in many fields and areas. Thus, EE has been widely and extensively exploited to contribute to many image processes needed depending on the purpose of image related application. EE has a lot of image's features from which significant information can be derived to feed to the respective application. Here, a brief review on those applications utilizing EE.

A proposed work presented in [29], [30] has used histogram equalization process in order to represent and recognize a series of speech. It has been also exploited by other techniques and applications such as mapping and geographic information systems to extract certain regions [31]. In robotics, DIP has been very widely exploited [32] to implement very crucial images-based tasks [31], [33]–[37] e.g., to do a 3D-based motion detection and multi-frame images recognition [38], to implement a vision-based tracking process for underwater vehicles [39], to enable an image sketching procedure from which a selected robot can benefit [32], or to enable the robot move to autonomously detect shadows of a region-of-interest (ROI) objects [40]. It has also been used by machine translation to measure the distance between an object and machine for a better focus calculation [41] or to propose a spectral-spatial classification method [42]. One of the attractive social network-applications is discussed in [43] which proposes a method helping to retrieve images from websites. In [44], a proposed method to carry out an analysis on certain images being transferred between social networks whereas this analysis has considered the level of pixel resolution in terms of color values and intensities for security purposes.

One interesting research field is image utilized social networks. It is with the help of image processes can do a varied band of useful services and implement important tasks for numerous applications. In [45], an approach that in parallel exploits textual and visual details to obtain the relation between tagged images is discussed.

One of the most important fields by which DIP is extensively exploited is medical imaging [46]–[52]. Different applications in literature have been reviewed. In [53], for example, a method to detect whether a fusion-based process can be applied on multimodal medical images or not is studied. A region-based segmentation process [54] was applied for certain objects to be highlighted from medical

images. Similarly, a detection method was designed in [55] to passively extract features from medical images taking into consideration pixels' intensities, region contrast, and inclined images. It tries to detect such tapering when compared to original digital images taken from the source after they have been subjectively evaluated. Sometimes, medical imaging applications require an additional supportive technique. Meaning, medical imaging applications are usually required to be combined with neural networks techniques in order to carry out complicated processes. For example, in [56], several image processes have been applied on medical images which, for example, include segmentation, features extraction, and classification. Then, trained neural network technique is applied on digital images to do further processes.

Recently, the topic of interactive image-based applications that require several successive processes has become of much interest. There have been many methods and approaches proposed for a wide range of research areas to provide a semi-optimal level of performance(s) taking into account accuracy, reliability, and robustness; e.g., an interactive image-based approach is discussed in [57]. In this research work, an image-based segmentation process has been applied on image's pixels in order to enhance the accuracy of segmentation using an updated interactively game theory for an optimized approach. It contributes to solve graph constraints caused by multi-layer combination. This also contributes to reduce the computation time caused by interactive layers of pixels for multi-frame images.

Another example has been described in [58]. It has simply adopted a game theoretical approach to be applied on pixel-based for image noise removal algorithm. The interesting point in this approach is that it has mimicked a game player role with the relation to player's partners. The approach reviewed in [58] did the same scenario with pixel-domain neighbors and joint pixels in order to efficiently remove noise from neighborhoods with errors as low as could to obtain a highly accurate noise-removal image. One of bio-medical image processing applications which has been presented in [59] has replaced the vision-based tracking approach with the game theory. In [59], an image-guided game theory application has been proposed in order to track changes occurring during a surgery. Also, this application aims to utilize the game theory to acquire images by inferring information collected from several images. The process of image acquisition is aimed to help produce an accurate made decision. Similarly to the two research works reviewed in [58], [59], the denoising image process has been efficiently exploited in order to enhance the application of game theory as it has been explained in [60]. This research work has somehow utilized both image's pixels regions clustering which has been discussed in [58] and also re-use each pixel as a game's player mentioned in [59]. This has enhanced region-based segmentation and its boundaries' detection. Thus, it has reduced noise when considering a set of pixels' neighbors when the work proposed in [60] is compared to median filter based algorithms in terms of visual quality, as stated.

Numerous research works which consider to process pixel-by-pixel in order to reduce noise from images for dynamical behavior based image theory have been reviewed in literature e.g., [61], [62]. Each implemented algorithm has treated pixels as players in order to detect boundaries and edges or to segment criteria-based regions.

Image-based gaming applications are so extensive due to its feature, as for example, to detect accurately a location better than other location detection algorithms. In [63], this conception using pixel based manipulation has been used in order to localize a set of unequal objects inside the image for the augmented reality game.

Applications mentioned above have included many image processes such as edge detection, region extraction, contrast equalization, pixel-by-pixel manipulation and many other processes to do several image-based tasks. Amongst them, edge extraction is of key importance to perform subsequent image processes. Therefore, edges extraction process can mainly affect the whole performance of those application in terms of computation time and accuracy and detection rate. It is needed to make the edges extraction simple and accurate so that it can contribute much to many image processing related applications. In this paper, PICA has been proposed in which a simple mask designed to do an IEE process. The proposed PICA for IEE aims to achieve fast PICA's processing time with less code complexity and computation time.

### III. THE PROPOSED PIXEL INTENSITY-BASED CONTRAST ALGORITHM (PICA) FOR IMAGE EDGES EXTRACTION (IEE)

The proposed PICA flowchart contains four main processes as shown in Figure 1.

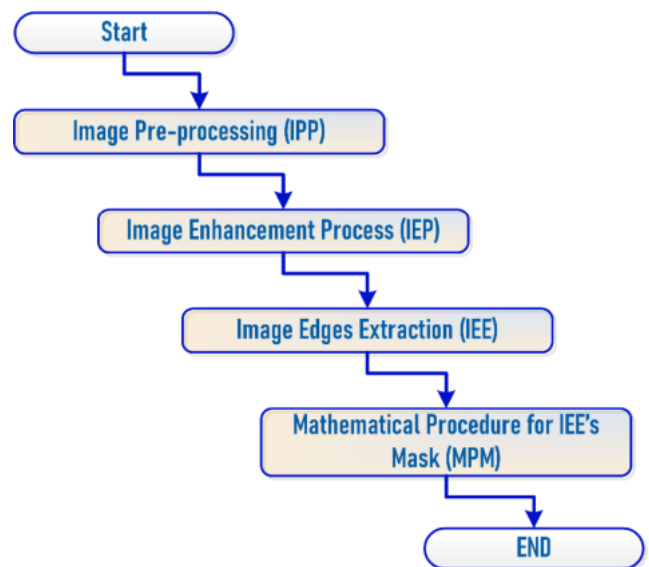


FIGURE 1. Proposed PICA flowchart.

In Figure 1, there have been four processes mentioned. In the first process, the image is pre-processed. In the second

one, the image is enhanced. The proposed pixel-based  $2 \times 4$  mask is applied on the resulted image. In the fourth one, the image's edges will be extracted.

### A. IPP

In this sub-section, several processes are proposed and applied on input image. Loading and initializing of the color input image and conversion of input image into a grayscale image are applied. After that, to acquire a black-and-white image from the grayscale one, a thresholding process is used. The proposed flowchart of image pre-processing is shown in Figure 2.

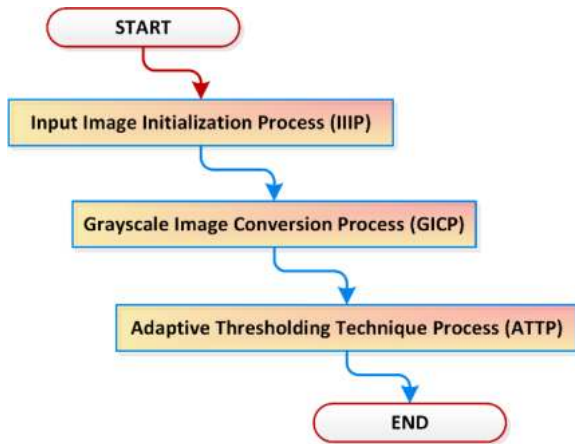


FIGURE 2. The proposed pre-processing flowchart.

#### 1) IIIP

The color input image will be loaded. The size of color input image will be tested according to the following condition: If image's height is 1024 pixels or below and the image's width is 768 pixels or below, then the image is processed.

#### 2) GICP

The input for this process is a color image. It is processed in order to compose the grayscale image. The following procedure has been applied on the color image shown in Figure 3(a).



FIGURE 3. A grayscale image conversion; (a) input image and (b) grayscale image of (a).

To convert the red, green, and blue (RGB) values in the color image into grayscale values; a sum of R, G, and B components will be weighted as formulated in (1):

$$I_{gs} = 0.299 \times R + 0.587 \times G + 0.114 \times B \quad (1)$$

where  $I_{gs}$  represents the obtained image representing grayscale values.

The proposed pseudocode implemented by the library Open Computer Vision (OpenCV) written using the programming language C++ and DEV-C++ Integrated Environment (IE) is provided in Algorithm 1.

### Algorithm 1 Color2Grayscale Image Conversion Process Using OpenCV and C++

```

// define and initialize image's variables
1: IplImage* cvFrame;
//load the color input image with file name "cImg"
2: cvFrame = cvLoadImage("cImg.jpg", 0);
//initialize and create a window to display output
3: cvNamedWindow("Gray scale image", 1);
//put the grayscale image into the window to display
4: cvShowImage("Gray scale image", cvFrame);
  
```

After this process is applied on the input image shown in Figure 3 (a), the output image is shown in Figure 3 (b).

#### 3) ATTP

Using an adaptive thresholding (AT) technique is needed due to that foreground's contents might be dismissed in case global thresholding technique has been used. The feature in ATs is that they consider all pixels of regions and objects exist in neighboring pixels of the currently processed pixels. When the mask is moving, neighboring pixels need to be carefully considered while processing. In this process, an AT technique has been used [26], [27]. The proposed pseudocode of this process is provided in Algorithm 2.

In Algorithm 2, the current pixel of input image (i.e., a grayscale image) will be compared to the locally neighboring pixels, as mentioned in line 13 of Algorithm 2.

The condition applied in order to produce the binarized value is mathematically represented in (2):

$$p_{th}(i, j) = \begin{cases} 0, & p_{in}(i, j) \times s^2 < op_s \times c \\ 255, & otherwise \end{cases} \quad (2)$$

where,

- $p_{th}(i, j)$  is the output thresholded assigned value to the pixel inside a thresholded image
- $p_{in}(i, j)$  is the currently processed pixel at location in the input grayscale image
- $s^2$  is the squared local region to which the  $p_{in}(i, j)$  belongs
- $op_s$  is a summation operator of a group of pixels per each row and this process is performed using (3)
- $c$  is a constant by which the binarization process is adjusted and it can be computed using the formula given in (4):

$$op_s = \sum_{j=0}^{w_{in}} p_{in}(i, j) |_{i^{th}} \quad (3)$$

$$c = 1 - T \quad (4)$$



**Algorithm 2** ATTP Pseudocode

```

//implement AT process
1: for(i = 0; i < hIin; i++)
2: {
3:     for(j = 0; j < wIin; j++)
4:     {
5:         //define a squared local region
6:         s = wIin/16;
7:         //define borders of the s region
8:         dx1 = j - s;   dx2 = j + s;
9:         dy1 = i - s;   dy2 = i + s;
10:        //check borders
11:        If(dx1 < 0) dx1 = 0;
12:        If(dy1 < 0) dy1 = 0;
13:        If(dx2 > wIin) dx2 = wIin - 1;
14:        If(dy2 > hIin) dy2 = hIin - 1;
15:        //perform a summation operation of grayscale
16:        values for each row subsequently
17:        ops = ops + pin(i,j); //ops is a summation
18:        operation
19:        //perform a criterion either to write a '0' or
20:        '255' value
21:        If(pin(i,j) × s2 < ops × c)
22:        pth(i,j) = 0;
23:        Else
24:        pth(i,j) = 255;
25:    }
26: }
    
```

where  $T$  is used to adjust the binarization process (a thresholded value) and  $0 < T < 1$ .

In (2), if the logical operation is true, then it is decided to convert the corresponding value of the pixel intensity in the input image  $p_{in}(i,j)$  into a binarized (thresholded) value assigned to the pixel at the same location in the thresholded image, i.e.,  $p_{th}(i,j)$ .

Once the AT is applied to the image (Figure 3 (b)), the output result will be a thresholded image shown in Figure 4.

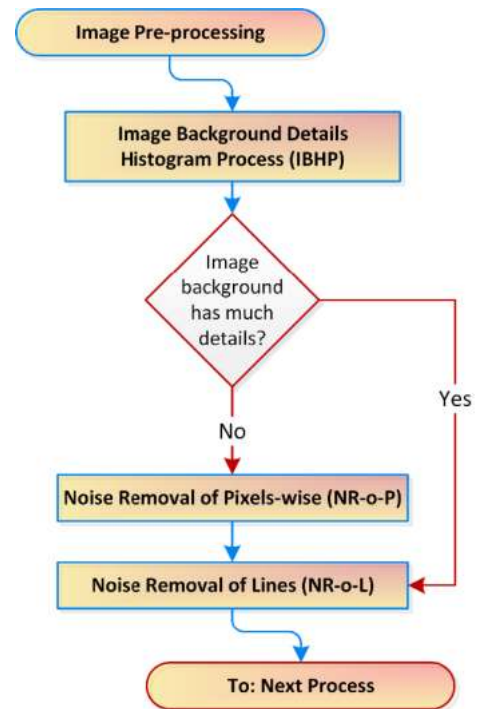


**FIGURE 4.** A thresholded image.

In Figure 4, foreground contents of regions and objects have been maintained after AT has been performed. This thresholding technique is able to process low contrast areas inside the gray scale image.

**B. IEP**

Image enhancement consists of several sub-processes applied to the output image resulted from the AT. Image enhancement process includes three proposed steps. One is a pre-process in which a statistical operation will be applied to the AT's image. The other two are post-processes in which on-pixel-wise processes are applied and performed to yield an enhanced image. An initialized pre-process to check whether the image being processed has many details or not. In order to make a decision, the Image Background details Histogram Process (IBHP) will be applied on the image. This pre-process is followed by two post-processes. Image enhancement process aims to remove unwanted details and is proposed to remove certain pixels. For noise removal, there are two post-processes which are Noise Removal of Pixels-wise (NR-o-P) and Noise Removal of Lines (NR-o-L) by processing unnecessary details existing in horizontal, vertical, and diagonal lines. The flowchart of Image Enhancement process is shown in Figure 5.



**FIGURE 5.** The proposed image enhancement flowchart.

1) IBHP

In this process, a very important procedure will be carefully applied on each region and/ or object as one unit. Meaning, this procedure is applied on each object separately whereas whole object's pixels are together treated. IBHP mainly performs histogram statistics for every region to make a decision either the currently processed region has huge number of pixels with almost similar intensity or not. Then, this process will go for the second region; starting to move from top to bottom and from left to right. After that, the third region is checked, then the fourth one, and so forth until whole

region and objects have been reached. The total number of foreground pixels will be judged whether the image has many details or not. If Yes, then the process will jump to the NR-o-L step; otherwise, the image will be enhanced using two subsequent steps which are NR-o-P and NR-o-L. The output image, after both NR-o-P and NR-o-L have been completely done, will be sent to the next process in order for the proposed mask to be applied on.

2) NR-O-P

This process will be applied on the thresholded image. NR-o-P processes pixels one-by-one in relation to its neighboring pixels. NR-o-P is going to consider four neighboring and eight neighboring pixels. Thus, the currently processed pixel will be compared if it has a potential relation to neighboring pixels or not. If Yes, it might be linked to an adjacent region in a four neighboring or eight neighboring relationship thru shared features in terms of pixel intensity. And in this case, the processed pixel will be considered a foreground one. Consequently, that pixel will be maintained. The Region of Interest (ROI) in this scenario is foreground details. Otherwise, it will be classified to belong to the image's background. In that case, it is unnecessary and would be removed.

3) NR-O-L

In order to remove unnecessary details that probably are noise or don't belong to objects inside images, multiple pixels gather in a way that they have no relations to neighboring pixels will be checked in order to decide to remove or keep. Therefore, the proposed PICA has applied an elimination algorithm dedicated for lines that don't belong to the ROI's regions, which has been proposed by [64], [65]. In [64], the proposed algorithm processes each pixel that exists in a cluster of pixels to make sure either it belongs to background of the image or not. If Yes, it is considered unnecessary and will be eliminated. Otherwise, that pixel would be unremoved. The proposed algorithm in [64] is able to process lines exist inside the image. Different directions can be processed even there are inclined objects because the proposed algorithm considered diagonal lines in addition to crossed lines to represent pixels being processed in 4- and 8- neighborhood similar to von Neumann and Moore neighborhood(s).

Both NR-o-P and NR-o-L processes have been used to enhance the thresholded image shown in Figure 6 (a) whereas the result is given in Figure 6 (b).

C. IEE

1) OVERVIEW

In this part, the enhanced thresholded image (Figure 6 (b)) is considered as an input for this process. the input image will be processed using the  $2 \times 4$  mask. Every pixel will be processed. To enhance the process in term of computation time, the proposed mask movement guarantees that each two neighboring pixels either they are horizontally or



FIGURE 6. Image enhancement process. (a) a thresholded image and (b) an enhanced binarized image of (a).

vertically adjacent, will be processed simultaneously at once. The proposed mask reduces the number of loops inside the coding stage. The proposed mask compares each two pixels to the locally neighboring region determined by eight pixels to increase the accuracy of an object's edges detection process.

2) PROPOSED DESIGN OF THE MASK

To implement the edge's detection process, the mask shown in Figure 7 will be proposed for this purpose.

	0	1	2	3
0	x,y-1	x,y	x,y+1	x,y+2
1	x+1,y-1	x+1,y	x+1,y+1	x+1,y+2

FIGURE 7. Mask used for the detection process of edges.

3) PROPOSED  $2 \times 4$  MASK FLOWCHART

A proposed flowchart explaining steps of the  $2 \times 4$  mask is shown in Figure 8.

The flowchart shown in Figure 8 briefly explains general steps of the conceptual idea of the proposed mask applied to extract edges of objects based on the contrast between pixels' intensities.

4) TECHNICAL PROCEDURE OF IEE

This process is dedicated to exploit the feature of pixel-based contrast. By utilizing pixels' intensities of the binarized image and also the abundance of contrast exists between white and black pixels, the  $2 \times 4$  mask is implemented.

The final step of PICA is to apply the proposed mask on the enhanced thresholded image shown in Figure 6 (b). when the 2 by 4 pixels related mask is applied to the selected image (i.e., thresholded), vertical edges will be highlighted and extracted. As for each region, left and right edges are extracted with a feature that the left edges have double thickness compared to those on the right edges. By applying the mask on the input, edges extraction and detection can be highlighted.

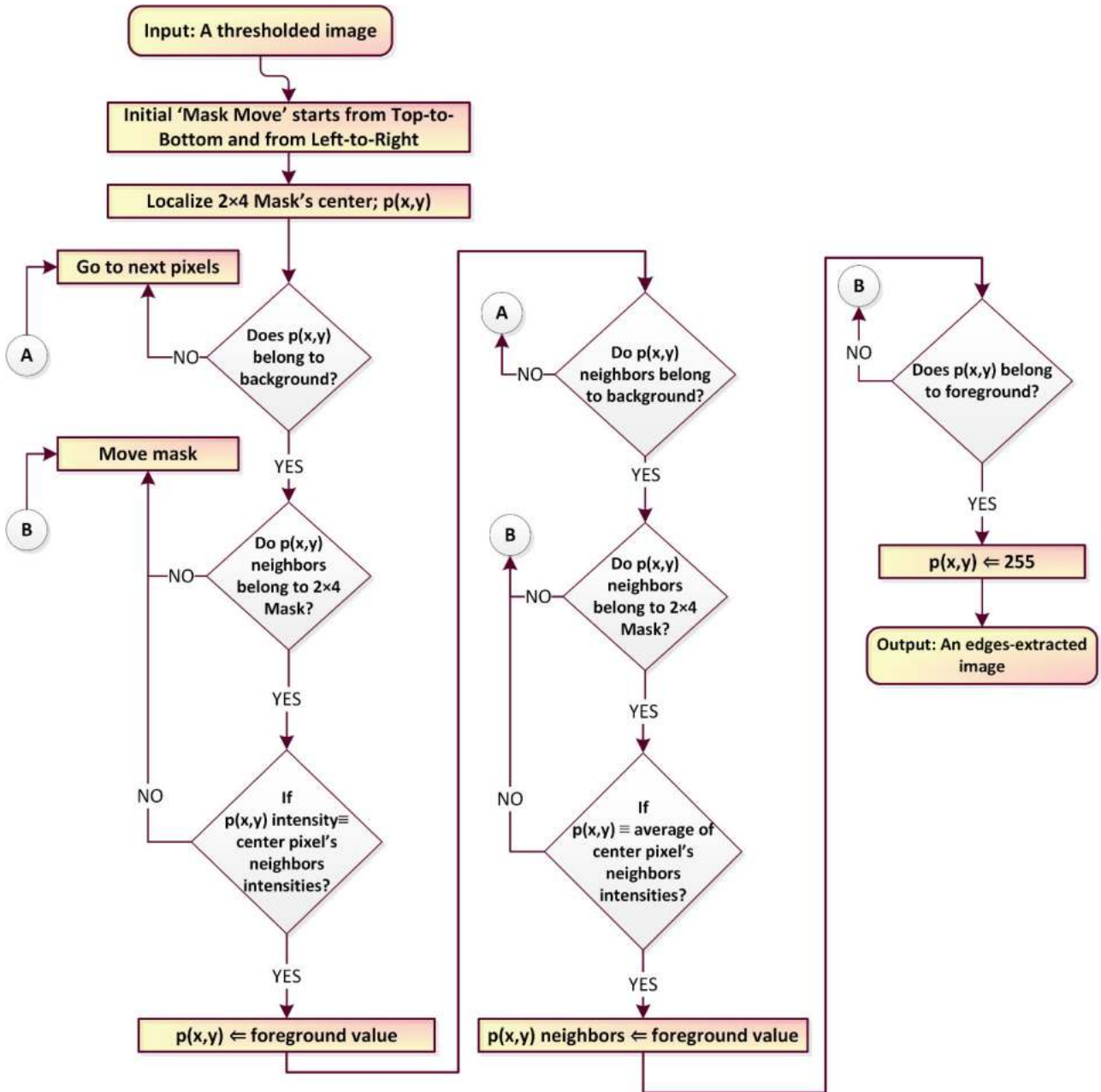


FIGURE 8. The flowchart of the proposed mask movement.

In order for PICA to perform the IEE process, the proposed  $2 \times 4$  mask shown in Figure 7 is applied on the input image shown in Figure 6 (b). The related pseudocode is provided in Algorithm 3. Once Algorithm 3 has been applied on the input image, the output result is shown in Figure 9.

5) MASK'S DESIGN AND CODING

In regard to the IEE process, the proposed PICA has adopted a simple mask with a design that allows for processing two neighboring pixels at once as the center of mask. This design has reduced processing time needed to process a single image by  $u^2$ . The center of the proposed mask verifies the

pixel intensities in relation to the locally neighboring pixels (i.e., regions) in a parallel way.

6) BIG-O-NOTATION AND CODE COMPLEXITY

This subsection describes a simple analysis of the proposed IEE process. First, the code structure is explained. Second, the code complexity is analyzed using Big-O-Notation.

a: CODE STRUCTURE

The proposed PICA code depends on using *if*-statement based condition in order to fulfil two criteria which are: *if*-statement conditional procedure and parallel processing of pixels.

**Algorithm 3** IEE Process Pseudocode

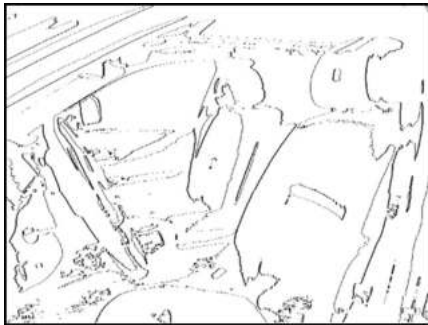
```

//define IEE-image as  $I_{EE}$ ; denotes for edges
//define a thresholded-image as  $I_{th}$ 

1: int  $r, l$ ;
2: for(int  $a = 0$ ;  $a < h_{I_{th}}$ ;  $a++$ )
3:   for (int  $b = 0$ ;  $b < w_{I_{th}}$ ;  $b++$ ) {
4:      $c = 1$ ;  $r = 1$ ;  $l = 1$ ;

5:     if ( $!p_{th}(a, b) \& !p_{th}(a, b+1) \& !p_{th}(a+1, b)$ 
6:        $\& !p_{th}(a+1, b+1)$ )
7:        $c = 0$ ;
8:       if ( $!(p_{th}(a, b+2) + p_{th}(a+1, b+2))$ 
9:          $r = 0$ ;
10:        if ( $!(p_{th}(a, b-1) + p_{th}(a+1, b-1))$ 
11:           $l = 0$ ;
12:          if ( $!c \& !r \& !l$ ){
13:             $p_{EE}(a, b) = 255$ ;
14:             $p_{EE}(a+1, b) = 255$ ;
15:          }
}

```

**FIGURE 9.** Process of extraction of edges.

That is, for the first criterion, it is important to use fewer number of while-loop(s) in coding. For the second criterion, two pixels are processed at once. Here, it is important not to check only one pixel while movement of the proposed mask but two pixels are checked for location  $(i, j)$  once. These two criteria are further explained for a more clarification.

**CRITERIA 1: IF-STATEMENT CONDITIONAL PROCEDURE (IF-CP)**

The proposed design of IEE's code has focused on how to reduce the use of while-loop. Instead, it has used if-statement conditions so that the time complexity is being reduced by  $u$  times, where  $u$  is the number of repetitions of loop inside the code. Since, there is a two-dimensional array to cover movement of mask from top-to-bottom and from left-to-right, there will be two while-loop(s) needed in the process. Hence, since they are replaced by if-statements, therefore, the code complexity is reduced by  $u \times u = u^2$ .

**CRITERIA 2: PIXELS PARALLEL PROCESSING PROCEDURE (4P)**

In this procedure, every pixel is needed to be checked as in lines 5, 7, and 9 of Algorithm 3. The proposed code parallelly uses a checking process. It checks two pixels at once rather than one-by-one pixel each time. This has reduced the code complexity by  $1/2 k$ . This is for processing pixels in the thresholded image. Similarly, this procedure is followed when the output of IEE are sent to the edges extracted image. Therefore, the code complexity in this case becomes  $1/4 k$ .

**b: BIG-O-NOTATION ANALYSIS**

As abovementioned discussed, the designed code has proposed two procedures be applied in order to reduce the code complexity while processing images. Here, the code complexity of IEE process is provided step-by-step in equations (5) - (9):

The code complexity before the if-CP is applied is formulated in (5):

$$CXY_{code}^b|_{if-CP} = O(M) \times O(N) \times O(u) \times O(u) \quad (5)$$

This equation can be simplified rewritten as in (6):

$$CXY_{code}^b|_{if-CP} = O(M) \times O(N) \times O(u^2) \quad (6)$$

Once the if-CP has been applied, the code complexity of IEE process is formulated in (7):

$$CXY_{code}^a|_{if-CP} = O(M) \times O(N) \quad (7)$$

As for the 4P, the code complexity before and after the 4P has been applied will be mathematically defined in equations (8) and (9):

$$CXY_{code}^b|_{4P} = O(M) \times O(N) \times O(k) \quad (8)$$

$$CXY_{code}^a|_{4P} = O(M) \times O(N) \times \frac{1}{4} O(k) \quad (9)$$

**D. MPM****1) DEFINITION OF FOREGROUND/BACKGROUND FUNCTIONS**

Let  $P_B$  be a function related to values of pixels intensities of background details, defined mathematically as in (10):

$$P_B = \{p(m, n) | p \text{ is a pixel element, } p \in \{0\}\} \quad (10)$$

where,

- $m$  represents number of image's row
- $n$  represents number of image's column
- $P_B$  represents a pixels' set, i.e.,  $\{P\} \subset \{0\}$  and in which (11) is fulfilled:

$$p(m, n) = c_1 \quad (11)$$

where  $c_1$  has a relation to the pixels set  $\{0\}$  and therefore,  $p(m, n)$  equals to values that fulfill the condition of 0's, as a '0' value is assigned in this case.



Similarly, let  $P_F$  be a function related to values of pixels intensities of foreground details, defined mathematically as in (12):

$$P_F = \{p(m, n) | p \text{ is a pixel element, } p \in \{1\}\} \quad (12)$$

where  $P_F$  represents a pixels' set, i.e.,  $\{P\} \subset \{1\}$  and in which (13) is fulfilled:

$$p(m, n) = c_2 \quad (13)$$

where  $c_2$  has a relation to the pixels set  $\{1\}$  and therefore,  $p(m, n)$  equals to values that fulfill the condition of 1's, as a '1' value is assigned in this case.

### 2) DEFINITION OF MASK'S FUNCTIONS

Let  $M$  be a mask, consisting of three blocks (parts) each of which consists of group-of-pixels. There are left, center, and right blocks (parts). Both left and right parts are designed in a way to put group-of-pixels in a vertical order, as shown in Figure 10 (a) and (c). The center part consists of a squared shape, as shown in Figure 10 (b).

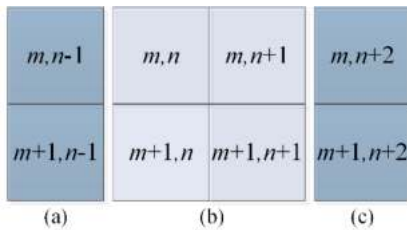


FIGURE 10. The proposed design of the mask M in three blocks; left (a), center (b), and right (c).

Let  $(m, n)$  be a location at an image's  $m^{th}$  row and  $n^{th}$  column for any pixel with an intensity represented by  $p(m, n)$ .

### 3) DEFINITION OF M MOVEMENT

Suppose that M movement will route from left-to-right and from top-to-bottom. To let M start moving, an initialization process is taken as mathematically represented in equations (14) and (15):

$$\dot{m} = f_{iz}^m \quad (14)$$

$$\dot{n} = f_{iz}^n \quad (15)$$

where,

- $\dot{m}$  is the first location of rows from which the pixel with the intensity  $p(m, n)$  starts movement
- $\dot{n}$  is the first location of columns from which the pixel with the intensity  $p(m, n)$  starts movement
- $f_{iz}^m$  and  $f_{iz}^n$  are initialized values that determine the location for first movement of M specified by  $m^{th}$  row and  $n^{th}$  column.

Therefore, the first location at which the mask M starts to move will be a function with a relation to  $\dot{m}$  and  $\dot{n}$ , defined in (16) as follows:

$$L_M = f(\dot{m}, \dot{n}) \quad (16)$$

Simply,  $f(\dot{m}, \dot{n})$  will be valid if two values are set and initialized. Meaning,  $f(\dot{m}, \dot{n})$  will have a location for both  $\dot{m}^{th}$  and  $\dot{n}^{th}$  values, as formulated in (17):

$$f(\dot{m}, \dot{n}) = (\dot{m}, \dot{n}) \quad (17)$$

Therefore, (16) can be simplified in (18):

$$L_M = (\dot{m}, \dot{n}) \quad (18)$$

In (18), it is mentioned that  $L_M$  can be defined as a two-pairs value. Meaning, as discussed above,  $L_M = (0, 0)$ .

At each time, M moves, the location of M, denoted by  $L_M$ , will be updated according to the next rule and the initialization function will be updated as in (19):

$$\vec{L}_M = L_M + \vec{d} \quad (19)$$

where,

- $\vec{L}_M$  represents a sub-sequential location and not the initialized location which is  $L_M$
- $\vec{d}$  is a distance or a new shifted location defined in (20), as follows:

$$\vec{d} = (\dot{m} + \delta, \dot{n} + \varepsilon) \quad (20)$$

where  $\delta$  and  $\varepsilon$  are small numbers and they are increased by +1 depending on rows and columns variations, respectively.

Both  $\delta$  and  $\varepsilon$  have functions related to  $\dot{m}$  and  $\dot{n}$ , respectively, as formulated in Equations (21) and (22):

$$\delta = f(\dot{m}) \quad (21)$$

$$\varepsilon = f(\dot{n}) \quad (22)$$

For every changing in variations of a row,  $\delta$  is increased but  $\varepsilon$ , i.e.,  $\delta = \delta + 1$  and  $\varepsilon = \varepsilon$  and vice versa in case with columns changing. This is mathematically represented in equations (23) and (24), respectively.

$$\delta = \delta + 1 \quad (23)$$

$$\varepsilon = \varepsilon + 1 \quad (24)$$

Therefore, while the column at the second move varies and row does not, then (24) will be applied only. Therefore, the new location for M will be as defined in (25):

$$\vec{L}_M = (\dot{m}, \dot{n} + 1) \quad (25)$$

when M reaches *Max\_img\_width*, then M moves to a new row. Therefore, values of locations will be updated according to equations (23), i.e.,  $\delta = 0 + 1$  but for  $\varepsilon$ , it is reset. Then, (25) will be updated according to the  $\delta$  value as formulated in (26):

$$\vec{L}_M = (\dot{m} + 1, \dot{n}) \quad (26)$$

#### 4) DEFINITION OF M SIZE'S FUNCTIONS

Let  $S(M)$  be a function of M's size. Mathematically,  $S(M)$  is defined as formulated in (27):

$$S(M) = W_M \times H_M \quad (27)$$

where,

- $W_M$  represents M width
- $H_M$  represents M height.

Since M consists of three parts, (27) can be extended to have new formulas based on each part as formulated in equations (28) – (30).

$$S(M_l) = w(M_l) \times h(M_l) \quad (28)$$

$$S(M_c) = w(M_c) \times h(M_c) \quad (29)$$

$$S(M_r) = w(M_r) \times h(M_r) \quad (30)$$

where,

- $S(M_l)$ ,  $S(M_c)$ , and  $S(M_r)$  are obtained sizes' values for left, center, and right parts exist in M, respectively
- $w(M_l)$ ,  $w(M_c)$ , and  $w(M_r)$  are width(s) related values of left, center, and right parts of M, respectively
- $h(M_l)$ ,  $h(M_c)$ , and  $h(M_r)$  are height(s) related values of left, center, and right parts of M, respectively.

#### 5) DEFINITION OF EE FUNCTIONS

At every time M moves, every pixel  $p(m, n)$  will be checked either it belongs to the left, center, or right part of M, as shown in Figure 11. Once the determined  $p(m, n)$  is located at center of M, i.e.,  $M_c$ , it is checked if the  $p(m, n)$  is at center of  $M_c$  or not. If YES, the currently selected pixel will be the center point of M.

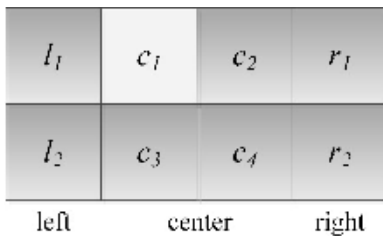


FIGURE 11. The design of center point of M at  $c_1$ .

When  $c_1$  is assigned a pixel value with which the M is currently located (as formulated in (31) expresses this as follows:

$$c_1 = p(m, n) \quad (31)$$

It is pre-defined that  $c_1$  value will be the center value of M. that is defined mathematically in (32):

$$c_p(M) = c_1 \quad (32)$$

where  $c_p(M)$  is the center point of M and  $c_1$  is the center of  $M_c$ .

$\therefore c_1 = p(m, n)$  and  $c_p(M) = c_1$ , this is updated in (33):

$$\therefore c_p(M) = p(m, n) \quad (33)$$

#### 6) DEFINITION OF LOGICAL OPERATION-BASED EE PROCESS

If  $M_c$  with its whole pixels are positioned as a region, a verification procedure is taken to know whether there is an edge or not. If it is a region only, and there is no an edge, it returns a '0' value, stored in a variable denoted by  $c$ . That is mathematically explained in (34):

$$c = \begin{cases} 0, & c_1 = 0, c_2 = 0, c_3 = 0, c_4 = 0 \\ 1, & otherwise \end{cases} \quad (34)$$

Similarly, this process is performed two times with left and right parts of M, i.e.,  $M_l$  and  $M_r$ , respectively, using equations (35) and (36):

$$l = \begin{cases} 0, & l_1 = 0, l_2 = 0 \\ 1, & otherwise \end{cases} \quad (35)$$

$$r = \begin{cases} 0, & r_1 = 0, r_2 = 0 \\ 1, & otherwise \end{cases} \quad (36)$$

Once each equation of equations (34) – (36) is equal to [0], i.e., they are black, the first column of  $M_c$  is converted to white i.e., [255]. That is mathematically formulated in equations (37) and (38) as follows:

$$c_1 = \begin{cases} 255, & l = 0, c = 0, r = 0 \\ 0, & otherwise \end{cases} \quad (37)$$

$$c_3 = \begin{cases} 255, & l = 0, r = 0 \\ 0, & otherwise \end{cases} \quad (38)$$

In equations (37) and (38),  $c_1$  and  $c_3$  represent the first column of  $M_c$ . This process aims to reduce details by removing unnecessary regions. This process also highlights certain parts, specifically those which are located between regions. This process will be applied on the thresholded image (see Figure 6 (b)). Once equations (37) and (38) have been applied on Figure 6 (b), edges are extracted as shown in Figure (9).

## IV. EXPERIMENTAL SETUP AND PREPARATION

### A. OVERVIEW

This section presents experimental conditions applied to images such as images sizes, types of image in term of background complexity, and the total number of images assigned to each size. Software platform and demonstration of implementation are also in detail described and explained in this section in order to give readers a further clarification on how such processes are implemented using the software and coding write-up.

### B. QUANTITATIVE CRITERIA AND CONDITIONS

In this experimental work, about 526 samples have been used. They have different images related conditions. Criteria applied on these images contain, as for example, images sizes, image background details, and other related features. Further criteria are discussed in the following sub-sections.

**C. EXPERIMENTAL CONDITIONS**

Samples used in this experiment have been divided into three datasets based on image’s size criterion. A total of 526 samples has been used for all datasets. Each dataset consists of a number of samples as shown in Table 1.

**TABLE 1. Dataset vs. number of images based on images’ sizes.**

Dataset	Image’s size	No. images
1	352×288	167
2	640×480	219
3	726×544	140
<b>Total</b>		<b>526</b>

The three datasets have been also classified based on image conditions (e.g., has complex background; lots of details, low contrast) into three groups. Each group represents one corresponding dataset, i.e., Dataset 1 is represented by G1, Dataset 2 is represented by G2, and Dataset 3 is represented by G3. Per each group, there are a number of samples (images) based on images’ conditions mentioned above. The total number of images assigned to each group based on the image’s conditions is shown in Table 2.

**TABLE 2. Group vs. number of images based on image’s conditions.**

Group	Class			Total no. images
	C1	C2	C3	
G1	59	48	60	167
G2	64	73	82	219
G3	55	54	31	140
<b>Total</b>				<b>526</b>

Images conditions have been defined into three classes. There are three main classes to which images of each group have been assigned. Description of each class is provided in Table 3.

**TABLE 3. Description of each class.**

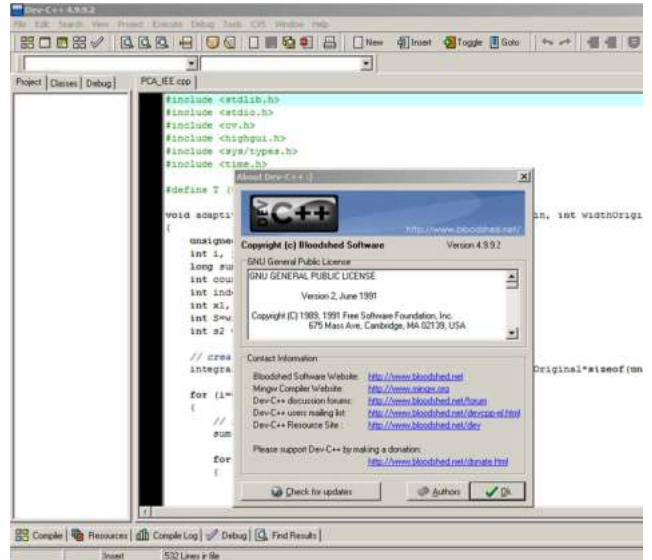
Class	Description
1	Blurry or low contrast
2	Complex background or lots of details
3	High similarity between regions or objects, (similarity in pixel intensities between edges and neighboring regions or objects)

**D. SOFTWARE PLATFORM**

In this experimental work, the C++ programming language has been used with the help of Open Computer Vision (OpenCV) library written in the DEV-C++ integrated environment (IE). Additionally, experiments have been tested using a MS-Windows 7 Professional run on a laptop with specifications as follows: CPU: Intel Core i3 2.26 GHz, RAM: 2GB. In Figure 12, a screenshot of DEV-C++ is provided.

**E. DEMONSTRATION**

In this subsection, a simple clarification is provided to show how experimental works are implemented in steps. Few examples are shown in figures 13-15.



**FIGURE 12. Software platform; C++ and OpenCV with the DEV-C++ IE.**



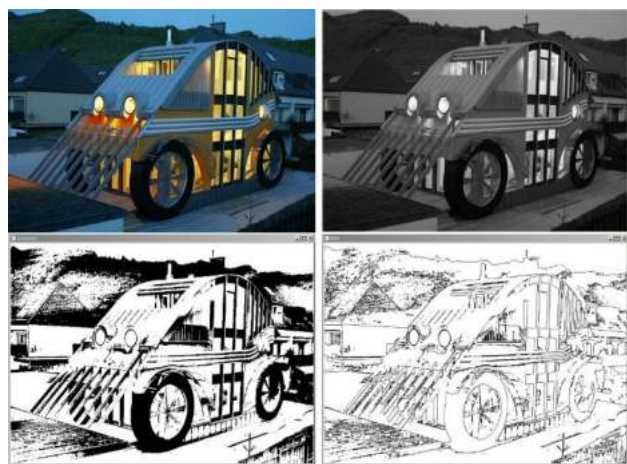
**FIGURE 13. Examples of PICA-IEE implementation on a “car house” sample image. (a) an input image (color), (b) a gray scale image, (c) an adaptive thresholded image, and (d) an edges’ extracted image (i.e., IEE).**

Examples shown in Figure 13-15 demonstrate that PICA is able to extract edges from different types of images inclusive varieties in colors, image background, similar-region areas, and day-or-night images. The threshold has been set to the value of 0.15 where it is the best threshold value for experiments. However, some regions that have high similarity with surrounding background might be removed from the edges detected image as shown in Figure 15 (d).

**F. INTERACTIVE MULTIMEDIA FILES DEMONSTRATION**

The proposed and real code for the part of IEE process is included with this paper and an open source. The implementation of this real code has been represented in a video motion form provided in a WMV file type. Besides, the real code of the PICA is also provided alongside this paper (IEE process part).





a b  
c d  
**FIGURE 14.** Examples of edges' extraction process by the proposed PICA-IEE of an image. (a) a color input image, (b) a gray scale image, (c) an adaptive thresholded image, and (d) PICA-IEE's output.

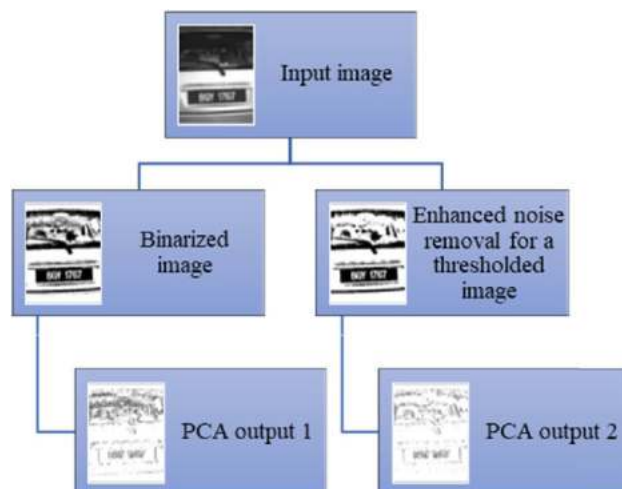


a b  
c d  
**FIGURE 15.** Examples of edges' extraction process by the proposed PICA-IEE applied on a "stilt house" image. (a) a color input image, (b) its gray scale image, (c) a binarized image, and (d) PICA-for-IEE's output (edges extraction image).

## V. RESULTS AND DISCUSSION

Here, results of PICA's evaluation are obtained and discussed. This section previews findings in five sub-sections, which are as follows:

1. In the first sub-section, the accuracy of PICA has been evaluated. Two processes have been used for a comparing two types of results. The first one shows the output results of PICA before a noise removal process of enhancement has been used while the second process shows obtained results of PICA for IEE after a noise removal has been applied to the thresholded image.
2. The second sub-section shows an evaluation of the proposed PICA in term of computation time for processing a single image taking into account the different images' sizes. The computation time for the three images' sizes (i.e., groups mentioned in Table 1) for each sample used in this experimental work with a total of 526 samples has been computed and graphically presented.



**FIGURE 16.** PICA's accuracy evaluation using noise removal process.

Additionally, an average computation time for whole samples based on groups of images (mentioned in Table 2) has been calculated and provided.

3. Performance of PICA in term of IEE has been evaluated. Different image's sizes have been also considered in this evaluation. Several samples representing images sizes (i.e., three classes mentioned in Table 2) have also been presented.
4. This sub-section evaluates the proposed PICA in term of robustness. Images classes mentioned previously in Table 3 are used in this experiment. Images conditions such as blurry, inclined, and complex background have been used in this experiment.
5. The fifth sub-section provides a comparative evaluation between the proposed PICA and a number of competitive research studies in terms of computation time, computation time-based enhancement rate for processing a single image with consideration of different sizes based on the three classes, accuracy, and code complexity.

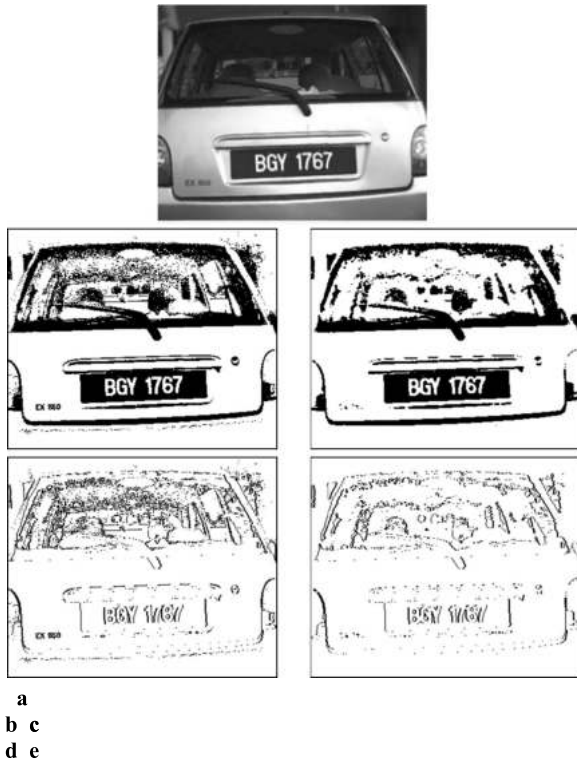
### A. ACCURACY EVALUATION OF THE PROPOSED PICA'S NOISE REMOVAL PROCESS

Two scenarios are used in this test to measure accuracy level of IEE of PICA. In the first scenario, this order of processes: input image → adaptive thresholding yielding a binarized image → the proposed  $2 \times 4$  mask of edges detection → output 1: edges' detected image, has been implemented. The second scenario has the following sequence: input image → an adaptive thresholding → an enhanced noise removal process → the proposed  $2 \times 4$  mask → output 2: edges' detected image. These two scenarios are illustrated in Figure 16.

As shown in Figure 16, the same input image will be tested using two processes. One of them includes an enhanced noise removal process applied on the thresholded image while the other process does not. Results obtained after both processes have been applied will be compared to each other in order to see which one has less noise or unnecessary details. In this



process, the PICA will be evaluated using a single input image as shown in Figure 17.



**FIGURE 17.** Applied noise removal process on PICA's accuracy evaluation; (a) input, (b) thresholded image, (c) edges' detected image of (b), (d) thresholded image after a noise removal has been applied, and (e) edges' detected image of (d).

The input image shown in Figure 17 (a) will go thru two processes. The first one applies an adaptive thresholding technique as shown in Figure 17 (b) while the second one applies two processes subsequently on input image (Figure 17 (a)) which are: an adaptive thresholding technique followed by a noise removal process and related result is shown in Figure 17 (c). After that, the same proposed mask process has been applied on both images shown in Figure 17 (b) and (c), in order to extract images' edges. Finally, corresponding obtained results are shown in Figure 17 (d) and (e), respectively. Obtained results shown in Figure 17 (e) compared to Figure 17 (d) is considered of accuracy. Additionally, most details are maintained with a less processing time is needed in the case of Figure 17 (e) compared to Figure 17 (d).

**B. PICA'S COMPUTATION TIME**

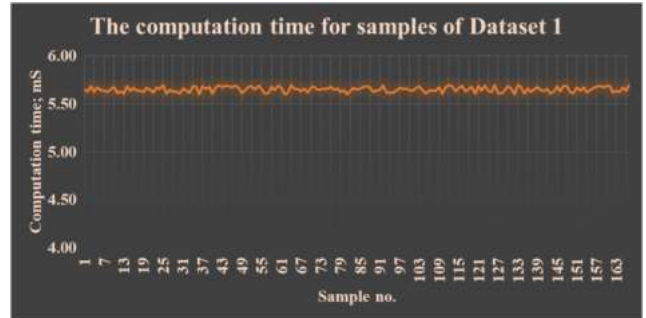
In this experiment, computation time for processing a single image has been provided. Besides, the average computation time per a group of samples (depending on images sizes) is also shown. The computed results are discussed as follows:

**1) TOTAL COMPUTATION FOR WHOLE SAMPLES OF EACH CATEGORY**

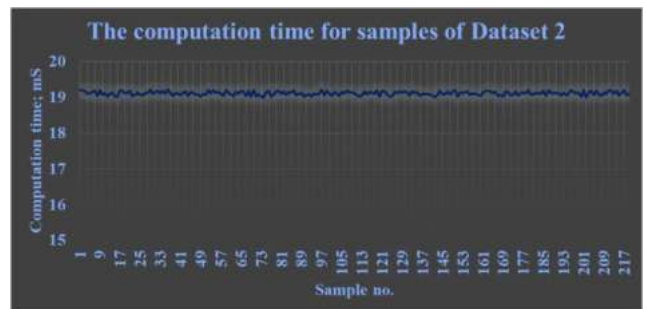
The obtained computation time after PICA has been applied on whole samples is recorded. Computation times for

processing a single image has been computed for each sample per each category.

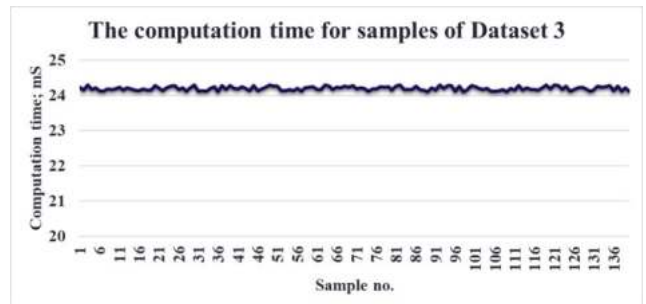
The computation times for the three datasets (i.e., groups, G1, G2, and G3) based on images' sizes will be shown in Figure 18, Figure 19, and Figure 20, respectively.



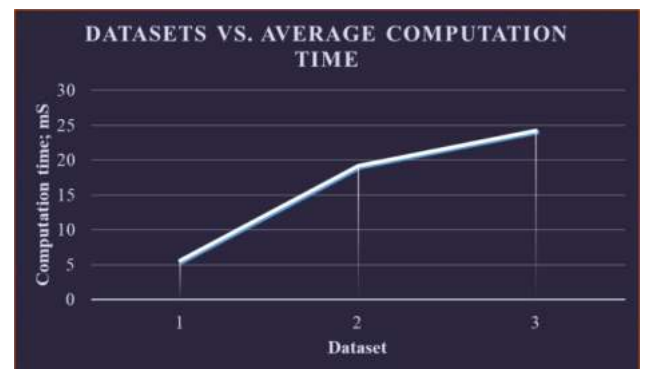
**FIGURE 18.** Computation times for whole samples of Dataset 1.



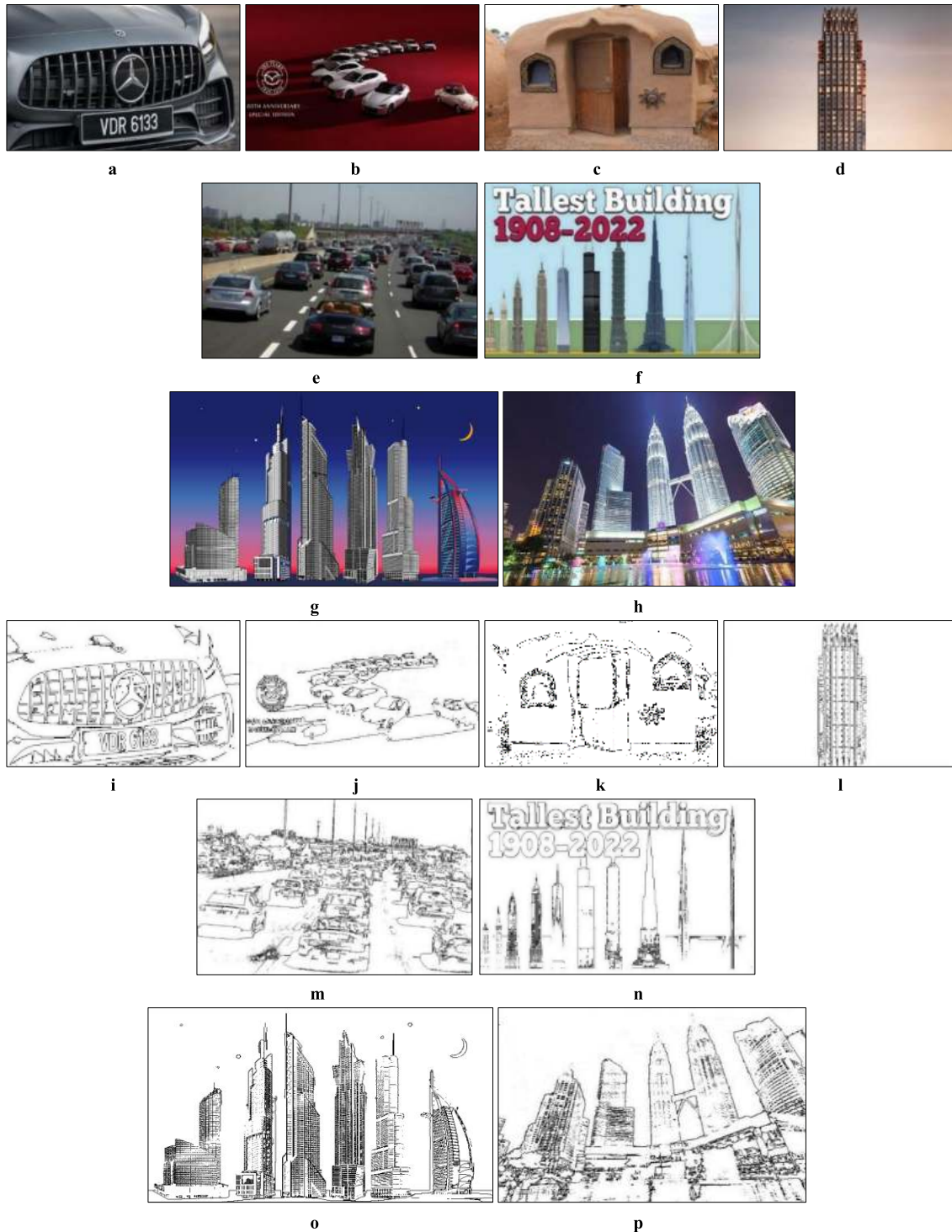
**FIGURE 19.** Computation times for whole samples of Dataset 2.



**FIGURE 20.** Computation times for whole samples of Dataset 3.



**FIGURE 21.** Average computation time for each dataset.



**FIGURE 22.** Examples of PICA's output. Input images are in (a)–(h) and their corresponding edges' extracted images in (i)–(p).

As shown in Figure 18, Figure 19, and Figure 20, computation times for the three datasets vary between 5.6 and 5.7, 19.1 and 19.2, and 24 mS, and 24.3 mS, respectively.

2) AVERAGE COMPUTATION TIME PER EACH DATASET  
 Averaged computation times are shown in Figure 21.

As shown in Figure 21, average computation times vary depending on the image size.

**C. PERFORMANCE EVALUATION OF IEE FOR EDGES DETECTION AND EXTRACTION PROCESS**

In this evaluation, the performance of the proposed PICA for IEE process will be evaluated. The evaluation is

quantitatively performed using different sizes of images and different types of classes (i.e., different conditions). Several samples of input images and their edges extracted output images will be shown in Figure 22.

As shown in Figure 22, edges of objects can be highlighted and detected using the proposed PICA. Samples shown in this figure are varied in terms of images' sizes and conditions. Samples include low contrast images (as mentioned in Table 3; Class 1) e.g., Figure 22 (c), (d), (f) in which some objects have low contrast with either background details or their neighbors. Some other samples include complex background and/ or lots of details (Class 2) e.g., Figure 22 (e). In Figure 22 (h), some objects or foreground details resemble with other objects' edges due to such excessive light.

**D. PICA'S ROBUSTNESS**

1) OVERVIEW

The PICA is evaluated in term of robustness against blurry, inclined, and complex background images.

2) EXPERIMENTS

This is to evaluate robustness of the proposed PICA. Firstly, the PICA is applied on varied types of images. Obtained results are evaluated. In this experiment, several categories of samples have been used. The selected samples of images might be inclined, low-contrast, and a huge portion of pixels images. Samples of images used in this experiment to be described as follows: The PICA is applied on a blurry and inclined image to extract its edges for the first experiment. In the second one, the PICA's robustness is evaluated taking into account a high similarity of pixels' intensities and also with low contrast between foreground details. It also shows a sample image in which a huge portion of pixels belong to both objects and foreground area is inclusive. Related input images and results obtained after the PICA has been applied can be found in Figure 23 (a) - (d).

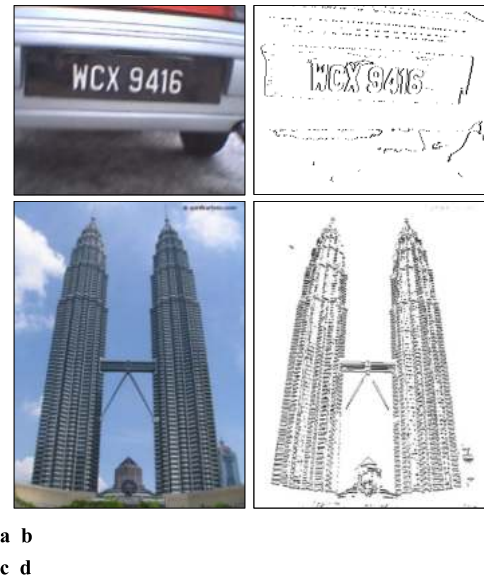
These examples show that the proposed PICA is able to perform an IEE with inclined and complex background images. The proposed PICA can accurately and efficiently detect edges associated with different objects inside an image and regions even their contrasts are low. The PICA has confirmed its robustness with inclined edges as shown in Figure 23 (a) and (b) and it has accurately extracted edges of car license plate. Additionally, as shown in Figure 23 (c), lots of edges exist whereas they have been efficiently detected as sown in Figure 23 (d).

**E. A COMPARISON BETWEEN PICA AND OTHER COMPETITIVE RESEARCH WORKS**

The proposed PICA for IEE will be compared to other competitive research works in terms of computation time, accuracy, and code complexity.

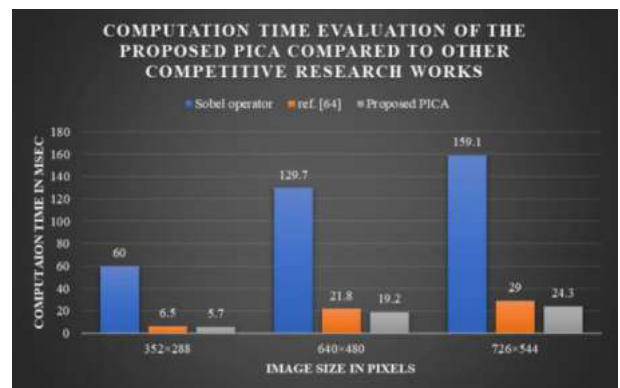
1) COMPUTATION TIME

In this experimental work, there have been different sizes of images used which are:  $352 \times 288$ ,  $640 \times 480$ , and  $726 \times 544$  representing images' width  $\times$  height. Computation time has



**FIGURE 23.** PICA evaluation in term of robustness; an input image shown in (a) and its corresponding vertical edges in (b); an input image in (c) and its horizontal edges shown in (d). Both (b) and (d) are extracted using the proposed PICA.

been considered in this experiment and calculated for three research studies inclusive the proposed PICA. The computation time consumed for processing a single image with a size of  $352 \times 288$  by using the proposed PICA is 5.7 milliseconds while the computation time needed to process the same image using the proposed method in [64] and Sobel operator have been found to be 6.5 and 60 milliseconds, respectively. The PICA's computation time for  $640 \times 480$  and  $726 \times 544$  have been found 19.2 and 24.3, respectively. Related results for other images' sizes will be shown in Figure 24.



**FIGURE 24.** Proposed PICA vs. other competitive works in term of computation time.

As shown in Figure 24, when comparing the computation time of processing one image using the proposed PICA to other competitive methods and operators, it looks that the proposed PICA outperforms others.



2) COMPUTATION TIME-BASED ENHANCEMENT PERCENTAGE COMPARED TO OTHER COMPETITIVE METHODS

The PICA computation time in a relation to total computation times for the three methods is calculated using (39):

$$R_{CT|PICA}\% = \frac{CT_{PICA}}{CT_{Sobel} + CT_{[64]} + CT_{PICA}}\% \quad (39)$$

where  $R_{CT|PICA}$  is rate of computation time of PICA.

By using (39),  $R_{CT|PICA}$  is  $\approx 7.9, 11.2,$  and  $11.4\%$  for images' sizes  $352 \times 288, 640 \times 480,$  and  $726 \times 544,$  respectively. From this, it is clear that rate(s) of computation time for both [64] and Sobel are 92.1, 88.8, and 88.5% for images' sizes  $352 \times 288, 640 \times 480,$  and  $726 \times 544,$  respectively. That is, a lot of computation time will be needed for both methods. To calculate the enhancement rate of PICA  $E_{CT|PICA}$  can be found using (40):

$$E_{CT|PICA} = (1 - R_{CT|PICA}) \times 100 \quad (40)$$

From (40), it is found that PICA has enhanced the edge detection and extraction process in term of computation time with a percentage of 92.1, 88.8, and 88.5% corresponding to images sizes  $352 \times 288, 640 \times 480,$  and  $726 \times 544.$  This rate varies depending on the image's size whereas  $E_{CT|PICA}$  has an inverse proportion with the image size.

3) ACCURACY

The performance of the proposed PICA's output is compared to other research works e.g., Sobel's in term of accuracy of edges extraction process. In Figure 25, a comparison between the proposed PICA and Sobel operator is performed in term of accuracy.

In this evaluation, the horizontal edges have been used. As shown in Figure 25, edges are obviously extracted for both Sobel operator and the PICA. For example, details and edges of textual contents might be acceptable and could be read. Thus, the PICA is able to extract small details specifically if the contrast between edges (foreground) and background details is high.

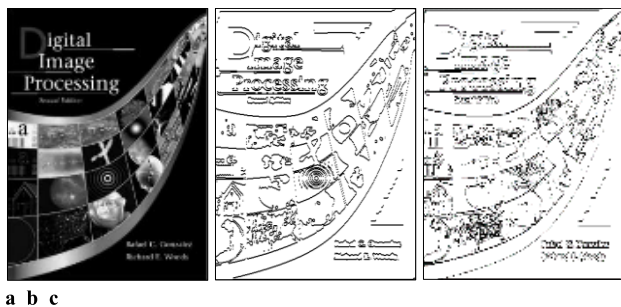


FIGURE 25. Image edges extraction (IEE) process of an image shown in (a) using Sobel operator in (b) and the proposed PICA in (c).

4) PICA'S CODE COMPLEXITY

In this subsection, a comparison between Sobel and PICA in term of code complexity is discussed.

The code complexity for Sobel is formulated in (41):

$$CXY_{code|S} = O(M) \times O(N) \times O(u^2) \quad (41)$$

As discussed in earlier, the code complexity for the proposed PICA can be obtained using a summation operation of equations (7) and (9) and the result is given in (42):

$$CXY_{code|IEE} = O(M) \times O(N) + \frac{1}{4}O(k) \quad (42)$$

where  $CXY_{code|IEE}$  is the code complexity of the process IEE.

It is simply re-formulated as in (43):

$$CXY_{code|P} = O(M) \times O(N) + \frac{1}{4}O(k) \quad (43)$$

In (43),  $\frac{1}{4}O(k)$  can be considered as a Big-O of a constant as formulated in (44):

$$CXY_{code|P} = O(M) \times O(N) + O(1) \quad (44)$$

It can be formulated in (45):

$$CXY_{code|P} = O(M) \times O(N) \quad (45)$$

By comparing (41) to (45), it can be concluded that the code complexity of PICA is less by  $u^2$  times.

F. QUALITATIVE AND QUANTITATIVE COMPARISONS BETWEEN PICA AND A NUMBER OF THE STATE-OF-THE-ART RESEARCH WORKS

In this subsection, the proposed PICA algorithm is compared to a number of the state-of-the-art methods in terms of quantitative and qualitative evaluation(s).

1) COMPUTATION TIME BASED PICA VS. SEVERAL COMPETITIVE RESEARCH WORKS

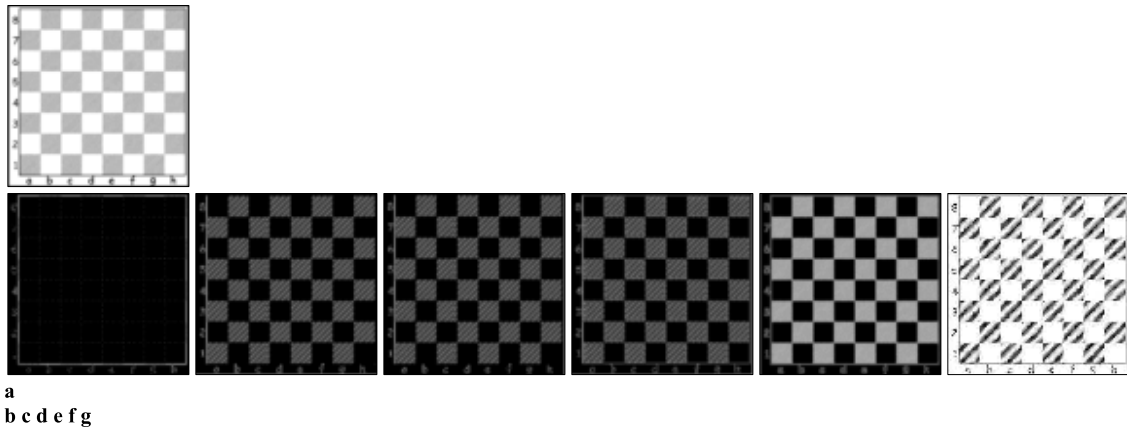
The PICA is evaluated using a computation time parameter and it is compared to a number of the state-of-the-art methods. In Table 4, PICA is compared to several edge detectors in term of computation time.

TABLE 4. Computation time based PICA vs. state-of-the-art methods.

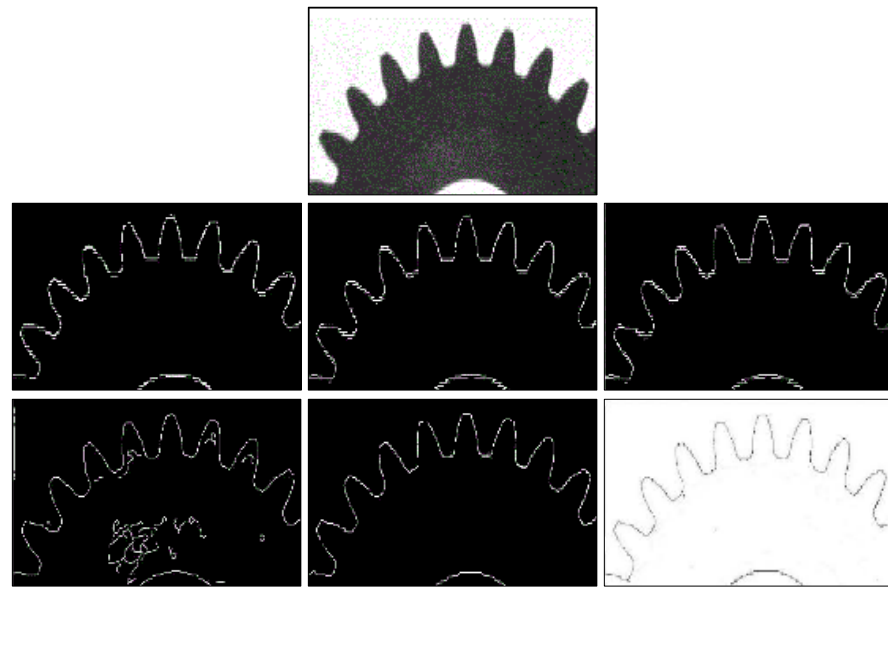
Method	Computation time (msec)
Robert	271.4
Sobel	292.1
Prewitt	780.5
Canny	994.2
[66]	129.1
[64]	23.38
Proposed PICA	20.59

As shown in Table 4, the proposed PICA has outperformed other competitive research works in term of the computation time.





**FIGURE 26.** IEE process of a black-white board sample (a), using several edge detectors; Robert (b), Sobel (c), Prewitt (d), Canny (e), [66] (f), and the proposed PICA (g).



**FIGURE 27.** A gear-sample (a) edge extraction process using a number of edge detectors which are Robert (b), Sobel (c), Prewitt (d), Canny (e), [66] (f), and the proposed PICA (g).



**FIGURE 28.** Performance of edges extraction process of an original image (a), using Robert (b), Sobel (c), Prewitt (d), Canny (e), [66] (f), and the proposed PICA (g).

2) ACCURACY BASED PICA COMPARED TO A NUMBER OF THE STATE-OF-THE-ART METHODS

The proposed PICA is compared to a number of the state-of-the-art methods in term of accuracy of edges detection.

Obtained results are shown in figures 26, 27, and 28. Original images shown in figures 26, 27, and 28 are available in [66].

As shown in Figure 26, edges are obviously extracted and accurately detected using all mentioned detectors.

**TABLE 5. PICA vs. state-of-the-art methods—a quantitative comparison.**

Method	No. of images/ samples used in experiment
[67]	200
[68]	300
[69]	200
Proposed PICA	526

But, in Figure 27, Canny detector fails to accurately and correctly extract edges while the rest detectors are suitable to be used with such images. For the sample shown in Figure 28 (a), three out of six detectors have failed to extract correctly edges where it is clear that the written phrase shown in Figure 28 (a) is not clearly readable and edges are not understandable as in Figure (b), (c) and (d). These three examples have confirmed that the proposed PICA can extract objects’ edges even there are two adjacent edges inside a small region of images (as proven in example shown in Figure 26 (a) and (g)). Additionally, PICA is able to extract small edges and texts as shown in Figure 28 (a) and (g).

**3) SAMPLES-BASED QUANTITATIVE COMPARISON BETWEEN PICA AND SEVERAL STATE-OF-THE-ART METHODS**

In this comparison, PICA is evaluated based on the samples have been collected and used in experiments for the edges extraction process. It is hence compared to other several methods applied to extract edges. In Table 5, a quantitative comparison is provided in which the number of samples used for the IEE process is used.

As shown in Table 5, the proposed PICA has been tested using 526 samples of images including several images dimension(s) and conditions.

**4) A COMPARISON BETWEEN PICA AND SEVERAL STATE-OF-THE-ART METHODS IN TERMS OF IMAGES’ SIZES, IMAGE CONDITIONS, AND ADVANTAGES**

In Table 6, the proposed PICA is compared to a number of the state-of-the-art methods quantitatively and qualitatively. Table 5 shows that PICA has the highest number of samples used for experiment of the edge extraction process.

**TABLE 6. PICA vs. state-of-the-art methods—a quantitative and qualitative evaluation.**

[Ref.]	Advantages	Image conditions	Suitable for real time applications	#Samples	Image size
[68]	- Suitable with robots’ applications and medical images	- Low contrast - Varied background	NA*	300	NA
[69]	- Applicable with heavy detailed images - Adaptive edge detector	- Complex background	Yes	200	480×640
[66]	- Accurate	- Stationary background	Yes	<30	550×550
Proposed PICA	- Adaptive threshold - Accurate	- Low contrast - Complex background	Yes	526	352×288, 640×480, and 726×544

\* ‘NA’ denotes for “Not Available” information.

As shown in Table 6, the proposed PICA has outperformed a number of the state-of-the-art methods in terms of no. of samples used per experiment, the variety of images sizes, and image conditions including ‘low contrast’ and ‘complex background’ images.

**5) CODE COMPLEXITY-BASED COMPARISON BETWEEN PICA AND SEVERAL STATE-OF-THE-ART METHODS**

In Table 7, a simple comparison between PICA and some other state-of-the-art methods in term of code complexity is performed.

**TABLE 7. PICA vs. state-of-the-art methods—code complexity analysis.**

Detector	Code complexity (Big-O-Notation)	PICA : detector
Robert	$O(m) \times O(n) \times O(r_1) \times O(r_2)$	1 : $r_1 \times r_2$
Sobel	$O(m) \times O(n) \times O(s_1) \times O(s_2)$	1 : $s_1 \times s_2$
Prewitt	$O(m) \times O(n) \times O(p_1) \times O(p_2)$	1 : $p_1 \times p_2$
[66]	$O(m) \times O(n) \times O(u_1) \times O(u_2)$	1 : $u_1 \times u_2$
Proposed PICA	$O(m) \times O(n)$	1 : 1

The provided comparison in Table 7 shows that the code complexity of PICA equals to  $O(m) \times O(n)$  which is less than other competitive methods/ detectors by  $c_1 \times c_2$  or  $c^2$  in case  $c_1 = c_2$ . The PICA includes less loop(s) than others by  $c^2$  as mentioned earlier.

**SOURCE CODE**

The related code: “PICA-4-IEE process code” has been uploaded to the Internet and is available via this link: [https://github.com/abbasghaili/PICA-4-IEE\\_code](https://github.com/abbasghaili/PICA-4-IEE_code)

**VI. CONCLUSION**

This paper has proposed a Pixel Intensity-based Contrast Algorithm (PICA) for Image Edges Extraction (IEE). The proposed PICA consists of several steps in which one of its main processes is to use a simple mask that processes pixel-by-pixel individually and in a group of lines to extract objects’ edges inside an image. The proposed mask depends on processing two pixels in a parallel way to reduce an

image's processing time so that the computation time will be enhanced and reduced specifically for images with large size or those images contain a huge portion of pixels and details. Besides, PICA is designed with such a way to reduce the use of while-loop(s) so that both computation time and code complexity get enhanced. The proposed PICA has been evaluated in terms of computation time, rate of enhancement of processing a single image, accuracy, and code complexity and compared to some other competitive research works.

- The PICA's computation time was compared to other competitive research works. PICA has less computation time than other research works. For processing an image with size equals to  $352 \times 288$ , PICA consumes 5.7 ms to process a single image. This computation time is less by about ten times when comparing to Sobel operator. Computation time of PICA has shown better performance for different sizes of images' sizes.
- In regard to enhancement rate of processing a single image with size of  $352 \times 288$ , PICA has enhanced the rate by about 92.1%.
- Results show that PICA for IEE performs accurately with different images' sizes and conditions.
- The code complexity of the proposed PICA has been analyzed and it is less by  $u^2$  than other competitive methods such as Sobel operator.
- Analysis of PICA's code and internal architecture and design of the mask show that PICA doesn't use additional while-loop(s) for the mask movement like the traditional methods. This has reduced PICA's code complexity and the computation time. This has contributed much in term of computation time and made PICA suitable for real-time applications.
- Results have shown accurate performance of PICA for IEE process. Robustness evaluation of the proposed PICA has shown high performance.

## REFERENCES

- [1] R. Subramanian and S. Sarkar, "Evaluation of algorithms for orientation invariant inertial gait matching," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 2, pp. 304–318, Feb. 2019, doi: [10.1109/TIFS.2018.2850032](https://doi.org/10.1109/TIFS.2018.2850032).
- [2] A. M. Nayak and R. Vemuri, "A secure tunable-precision architecture for image processing applications," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Kuala Lumpur, Malaysia, Jan. 2018, pp. 1–5, doi: [10.1109/ICCE.2018.8326340](https://doi.org/10.1109/ICCE.2018.8326340).
- [3] Y. Yan, W. Ren, and X. Cao, "Recolored image detection via a deep discriminative model," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 1, pp. 5–17, Jan. 2019, doi: [10.1109/TIFS.2018.2834155](https://doi.org/10.1109/TIFS.2018.2834155).
- [4] C. Lin and A. Kumar, "A CNN-based framework for comparison of contactless to contact-based fingerprints," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 3, pp. 662–676, Mar. 2019, doi: [10.1109/TIFS.2018.2854765](https://doi.org/10.1109/TIFS.2018.2854765).
- [5] S. Voloshynovskiy, O. Koval, M. K. Mihcak, and T. Pun, "The edge process model and its application to information-hiding capacity analysis," *IEEE Trans. Signal Process.*, vol. 54, no. 5, pp. 1813–1825, May 2006, doi: [10.1109/TSP.2006.871965](https://doi.org/10.1109/TSP.2006.871965).
- [6] Y. Zhang, L. Y. Zhang, J. Zhou, L. Liu, F. Chen, and X. He, "A review of compressive sensing in information security field," *IEEE Access*, vol. 4, pp. 2507–2519, 2016, doi: [10.1109/ACCESS.2016.2569421](https://doi.org/10.1109/ACCESS.2016.2569421).
- [7] Y. Guo, X. Cao, W. Zhang, and R. Wang, "Fake colored image detection," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 8, pp. 1932–1944, Aug. 2018, doi: [10.1109/TIFS.2018.2806926](https://doi.org/10.1109/TIFS.2018.2806926).
- [8] O.-J. Kwon, S. Choi, and B. Lee, "A watermark-based scheme for authenticating JPEG image integrity," *IEEE Access*, vol. 6, pp. 46194–46205, 2018, doi: [10.1109/ACCESS.2018.2866153](https://doi.org/10.1109/ACCESS.2018.2866153).
- [9] R. Das, E. Piciuccio, E. Maiorana, and P. Campisi, "Convolutional neural network for Finger-Vein-Based biometric identification," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 2, pp. 360–373, Feb. 2019, doi: [10.1109/TIFS.2018.2850320](https://doi.org/10.1109/TIFS.2018.2850320).
- [10] A. S. Mahmood and M. S. M. Rahim, "Novel method for image security system based on improved SCAN method and pixel rotation technique," *J. Inf. Secur. Appl.*, vol. 42, pp. 57–70, Oct. 2018, doi: [10.1016/j.jisa.2018.08.001](https://doi.org/10.1016/j.jisa.2018.08.001).
- [11] Z. Liu, H. Chen, W. Blondel, Z. Shen, and S. Liu, "Image security based on iterative random phase encoding in expanded fractional Fourier transform domains," *Opt. Lasers Eng.*, vol. 105, pp. 1–5, Jun. 2018, doi: [10.1016/j.optlaseng.2017.12.007](https://doi.org/10.1016/j.optlaseng.2017.12.007).
- [12] D. C. Mishra, R. K. Sharma, S. Suman, and A. Prasad, "Multi-layer security of color image based on chaotic system combined with RP2DFRFT and arnold transform," *J. Inf. Secur. Appl.*, vol. 37, pp. 65–90, Dec. 2017, doi: [10.1016/j.jisa.2017.09.006](https://doi.org/10.1016/j.jisa.2017.09.006).
- [13] M. Xu and Z. Tian, "A novel image encryption algorithm based on self-orthogonal Latin squares," *Optik*, vol. 171, pp. 891–903, Oct. 2018, doi: [10.1016/j.ijleo.2018.06.112](https://doi.org/10.1016/j.ijleo.2018.06.112).
- [14] T. Zhao, Q. Ran, L. Yuan, Y. Chi, and J. Ma, "Security of image encryption scheme based on multi-parameter fractional Fourier transform," *Opt. Commun.*, vol. 376, pp. 47–51, Oct. 2016, doi: [10.1016/j.optcom.2016.05.016](https://doi.org/10.1016/j.optcom.2016.05.016).
- [15] W.-C. Lin and J.-W. Wang, "Edge detection in medical images with quasi high-pass filter based on local statistics," *Biomed. Signal Process. Control*, vol. 39, pp. 294–302, Jan. 2018, doi: [10.1016/j.bspc.2017.08.011](https://doi.org/10.1016/j.bspc.2017.08.011).
- [16] D. Gupta and R. S. Anand, "A hybrid edge-based segmentation approach for ultrasound medical images," *Biomed. Signal Process. Control*, vol. 31, pp. 116–126, Jan. 2017, doi: [10.1016/j.bspc.2016.06.012](https://doi.org/10.1016/j.bspc.2016.06.012).
- [17] L. Luo, Y. Tang, Q. Lu, X. Chen, P. Zhang, and X. Zou, "A vision methodology for harvesting robot to detect cutting points on peduncles of double overlapping grape clusters in a vineyard," *Comput. Ind.*, vol. 99, pp. 130–139, Aug. 2018, doi: [10.1016/j.compind.2018.03.017](https://doi.org/10.1016/j.compind.2018.03.017).
- [18] M. Cai, J. Song, and M. R. Lyu, "A new approach for video text detection," in *Proc. Int. Conf. Image Process.*, vol. 1, 2002, pp. I-117–I-120, doi: [10.1109/ICIP.2002.1037973](https://doi.org/10.1109/ICIP.2002.1037973).
- [19] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (ALPR): A state-of-the-art review," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 2, pp. 311–325, Feb. 2013, doi: [10.1109/TCSVT.2012.2203741](https://doi.org/10.1109/TCSVT.2012.2203741).
- [20] A. Almagambetov, S. Velipasalar, and M. Casares, "Robust and computationally lightweight autonomous tracking of vehicle taillights and signal detection by embedded smart cameras," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3732–3741, Jun. 2015, doi: [10.1109/TIE.2015.2400420](https://doi.org/10.1109/TIE.2015.2400420).
- [21] H. Ma, E. Smart, A. Ahmed, and D. Brown, "Radar image-based positioning for USV under GPS denial environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 72–80, Jan. 2018, doi: [10.1109/TITS.2017.2690577](https://doi.org/10.1109/TITS.2017.2690577).
- [22] Z. Chen, C. Wang, H. Luo, H. Wang, Y. Chen, C. Wen, Y. Yu, L. Cao, and J. Li, "Vehicle detection in high-resolution aerial images based on fast sparse representation classification and multiorder feature," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2296–2309, Aug. 2016, doi: [10.1109/TITS.2016.2517826](https://doi.org/10.1109/TITS.2016.2517826).
- [23] Z. Chen, C. Wang, C. Wen, X. Teng, Y. Chen, H. Guan, H. Luo, L. Cao, and J. Li, "Vehicle detection in high-resolution aerial images via sparse representation and superpixels," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 103–116, Jan. 2016, doi: [10.1109/TGRS.2015.2451002](https://doi.org/10.1109/TGRS.2015.2451002).
- [24] M. Kmiec and A. Glowacz, "Object detection in security applications using dominant edge directions," *Pattern Recognit. Lett.*, vol. 52, pp. 72–79, Jan. 2015, doi: [10.1016/j.patrec.2014.09.018](https://doi.org/10.1016/j.patrec.2014.09.018).
- [25] A. M. Al-Ghaili, S. Mashohor, A. R. Ramli, and A. Ismail, "Vertical-edge-based car-license-plate detection method," *IEEE Trans. Veh. Technol.*, vol. 62, no. 1, pp. 26–38, Jan. 2013, doi: [10.1109/TVT.2012.2222454](https://doi.org/10.1109/TVT.2012.2222454).
- [26] D. Bradley and G. Roth, "Adaptive thresholding using the integral image," *J. Graph. Tools*, vol. 12, no. 2, pp. 13–21, Jan. 2007, doi: [10.1080/2151237X.2007.10129236](https://doi.org/10.1080/2151237X.2007.10129236).
- [27] F. Shafait, D. Keysers, and T. M. Breuel, "Efficient implementation of local adaptive thresholding techniques using integral images," *Proc. SPIE*, vol. 6815, Jan. 2008, Art. no. 681510.
- [28] I. Sobel and G. Feldman, "A  $3 \times 3$  isotropic gradient operator for image processing," in *Pattern Classification and Scene Analysis*, R. Duda and P. Hart, Eds. New York, NY, USA: Wiley, 1973, pp. 271–272.

- [29] E. Sariyanidi, H. Gunes, and A. Cavallaro, "Robust registration of dynamic facial sequences," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1708–1722, Apr. 2017, doi: [10.1109/TIP.2016.2639448](https://doi.org/10.1109/TIP.2016.2639448).
- [30] A. de la Torre, A. M. Peinado, J. C. Segura, J. L. Perez-Cordoba, M. C. Benitez, and A. J. Rubio, "Histogram equalization of speech representation for robust speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 3, pp. 355–366, May 2005, doi: [10.1109/TSA.2005.845805](https://doi.org/10.1109/TSA.2005.845805).
- [31] J. R. Taylor and S. T. Lovell, "Mapping public and private spaces of urban agriculture in Chicago through the analysis of high-resolution aerial images in Google Earth," *Landscape Urban Planning*, vol. 108, no. 1, pp. 57–70, Oct. 2012.
- [32] M. Tiwari, S. S. Lamba, and B. Gupta, "An image processing and computer vision framework for efficient robotic sketching," *Procedia Comput. Sci.*, vol. 133, pp. 284–289, Jan. 2018, doi: [10.1016/j.procs.2018.07.035](https://doi.org/10.1016/j.procs.2018.07.035).
- [33] H.-Y. Liu, W.-J. Wang, R.-J. Wang, C.-W. Tung, P.-J. Wang, and I.-P. Chang, "Image recognition and force measurement application in the humanoid robot imitation," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 1, pp. 149–161, Jan. 2012, doi: [10.1109/TIM.2011.2161025](https://doi.org/10.1109/TIM.2011.2161025).
- [34] J. Salmeron-Garcia, P. Inigo-Blasco, F. Diaz-del-Rio, and D. Cagigas-Muniz, "A tradeoff analysis of a cloud-based robot navigation assistant using stereo image processing," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 444–454, Apr. 2015, doi: [10.1109/TASE.2015.2403593](https://doi.org/10.1109/TASE.2015.2403593).
- [35] X. Liang, H. Wang, W. Chen, D. Guo, and T. Liu, "Adaptive image-based trajectory tracking control of wheeled mobile robots with an uncalibrated fixed camera," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 6, pp. 2266–2282, Nov. 2015, doi: [10.1109/TCST.2015.2411627](https://doi.org/10.1109/TCST.2015.2411627).
- [36] F. C. A. Groen, G. A. den Boer, A. van Inge, and R. Stam, "A chess-playing robot: Lab course in robot sensor integration," *IEEE Trans. Instrum. Meas.*, vol. 41, no. 6, pp. 911–914, Dec. 1992, doi: [10.1109/19.199432](https://doi.org/10.1109/19.199432).
- [37] J. L. Powell, D. Grossi, R. Corcoran, F. Gobet, and M. García-Fiñana, "The neural correlates of theory of mind and their role during empathy and the game of chess: A functional magnetic resonance imaging study," *Neuroscience*, vol. 355, pp. 149–160, Jul. 2017, doi: [10.1016/j.neuroscience.2017.04.042](https://doi.org/10.1016/j.neuroscience.2017.04.042).
- [38] J. Arunehru, G. Chamundeeswari, and S. P. Bharathi, "Human action recognition using 3D convolutional neural networks with 3D motion cuboids in surveillance videos," *Procedia Comput. Sci.*, vol. 133, pp. 471–477, Jan. 2018, doi: [10.1016/j.procs.2018.07.059](https://doi.org/10.1016/j.procs.2018.07.059).
- [39] G. S. Kumar, U. V. Painumgal, M. N. V. C. Kumar, and K. H. V. Rajesh, "Autonomous underwater vehicle for vision based tracking," *Procedia Comput. Sci.*, vol. 133, pp. 169–180, Jan. 2018, doi: [10.1016/j.procs.2018.07.021](https://doi.org/10.1016/j.procs.2018.07.021).
- [40] T. Tayama, Y. Kurose, T. Nitta, K. Harada, Y. Someya, S. Omata, F. Arai, F. Araki, K. Totsuka, T. Ueta, Y. Noda, M. Takao, M. Aihara, N. Sugita, and M. Mitsuishi, "Image processing for autonomous positioning of eye surgery robot in micro-cannulation," *Procedia CIRP*, vol. 65, pp. 105–109, Jan. 2017, doi: [10.1016/j.procir.2017.04.036](https://doi.org/10.1016/j.procir.2017.04.036).
- [41] S. Li, J. T.-Y. Kwok, I. W.-H. Tsang, and Y. Wang, "Fusing images with different focuses using support vector machines," *IEEE Trans. Neural Netw.*, vol. 15, no. 6, pp. 1555–1561, Nov. 2004, doi: [10.1109/TNN.2004.837780](https://doi.org/10.1109/TNN.2004.837780).
- [42] L. He, Y. Li, X. Li, and W. Wu, "Spectral–Spatial classification of hyperspectral images via spatial translation-invariant wavelet-based sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2696–2712, May 2015, doi: [10.1109/TGRS.2014.2363682](https://doi.org/10.1109/TGRS.2014.2363682).
- [43] X. Qian, D. Lu, Y. Wang, L. Zhu, Y. Y. Tang, and M. Wang, "Image re-ranking based on topic diversity," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3734–3747, Aug. 2017, doi: [10.1109/TIP.2017.2699623](https://doi.org/10.1109/TIP.2017.2699623).
- [44] A. Castiglione, G. Cattaneo, and A. De Santis, "A forensic analysis of images on online social networks," in *Proc. 3rd Int. Conf. Intell. Netw. Collaborative Syst.*, Nov. 2011, pp. 679–684, doi: [10.1109/INCoS.2011.17](https://doi.org/10.1109/INCoS.2011.17).
- [45] Y. Gao, M. Wang, Z.-J. Zha, J. Shen, X. Li, and X. Wu, "Visual-textual joint relevance learning for tag-based social image search," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 363–376, Jan. 2013, doi: [10.1109/TIP.2012.2202676](https://doi.org/10.1109/TIP.2012.2202676).
- [46] R. H. H. M. Philipsen, P. Maduskar, L. Hogeweg, J. Melendez, C. I. Sanchez, and B. van Ginneken, "Localized energy-based normalization of medical images: Application to chest radiography," *IEEE Trans. Med. Imag.*, vol. 34, no. 9, pp. 1965–1975, Sep. 2015, doi: [10.1109/TMI.2015.2418031](https://doi.org/10.1109/TMI.2015.2418031).
- [47] R. Bharath, P. K. Mishra, and P. Rajalakshmi, "Automated quantification of ultrasonic fatty liver texture based on curvelet transform and SVD," *Biocybernetics Biomed. Eng.*, vol. 38, no. 1, pp. 145–157, 2018, doi: [10.1016/j.bbe.2017.12.004](https://doi.org/10.1016/j.bbe.2017.12.004).
- [48] A. Qayyum, S. M. Anwar, M. Awais, and M. Majid, "Medical image retrieval using deep convolutional neural network," *Neurocomputing*, vol. 266, pp. 8–20, Nov. 2017, doi: [10.1016/j.neucom.2017.05.025](https://doi.org/10.1016/j.neucom.2017.05.025).
- [49] A. Kanso and M. Ghebleh, "An efficient lossless secret sharing scheme for medical images," *J. Vis. Commun. Image Represent.*, vol. 56, pp. 245–255, Oct. 2018, doi: [10.1016/j.jvcir.2018.09.018](https://doi.org/10.1016/j.jvcir.2018.09.018).
- [50] R. Lan, H. Wang, S. Zhong, Z. Liu, and X. Luo, "An integrated scattering feature with application to medical image retrieval," *Comput. Electr. Eng.*, vol. 69, pp. 669–675, Jul. 2018, doi: [10.1016/j.compeleceng.2018.01.027](https://doi.org/10.1016/j.compeleceng.2018.01.027).
- [51] B. Aribisala and O. Olabanjo, "Medical image processor and repository—MIPAR," *Inform. Med. Unlocked*, vol. 12, pp. 75–80, Jan. 2018, doi: [10.1016/j.imu.2018.06.005](https://doi.org/10.1016/j.imu.2018.06.005).
- [52] J. S. H. Baxter, E. Gibson, R. Eagleson, and T. M. Peters, "The semiotics of medical image segmentation," *Med. Image Anal.*, vol. 44, pp. 54–71, Feb. 2018, doi: [10.1016/j.media.2017.11.007](https://doi.org/10.1016/j.media.2017.11.007).
- [53] K. G. Baum, M. Helguera, J. P. Hornak, J. P. Kerekes, E. D. Montag, M. Z. Unlu, D. H. Feiglin, and A. Krol, "Techniques for fusion of multimodal images: Application to breast imaging," in *Proc. Int. Conf. Image Process.*, Oct. 2006, pp. 2521–2524, doi: [10.1109/ICIP.2006.312806](https://doi.org/10.1109/ICIP.2006.312806).
- [54] H. Gao, L. Dou, W. Chen, and G. Xie, "The applications of image segmentation techniques in medical CT images," in *Proc. 30th Chin. Control Conf.*, 2011, pp. 3296–3299.
- [55] W. Xu, H. Huang, and W. Yu, "A study of medical image tampering detection," in *Proc. Int. Conf. Med. Image Anal. Clin. Appl.*, Jun. 2010, pp. 131–134, doi: [10.1109/MIACA.2010.5528509](https://doi.org/10.1109/MIACA.2010.5528509).
- [56] K. C. L. Wong, T. Syeda-Mahmood, and M. Moradi, "Building medical image classifiers with very limited data using segmentation networks," *Med. Image Anal.*, vol. 49, pp. 105–116, Oct. 2018, doi: [10.1016/j.media.2018.07.010](https://doi.org/10.1016/j.media.2018.07.010).
- [57] T. Wang, Q. Sun, Z. Ji, Q. Chen, and P. Fu, "Multi-layer graph constraints for interactive image segmentation via game theory," *Pattern Recognit.*, vol. 55, pp. 28–44, Jul. 2016, doi: [10.1016/j.patcog.2016.01.018](https://doi.org/10.1016/j.patcog.2016.01.018).
- [58] Y. Chen and K. J. R. Liu, "A game theoretical approach for image denoising," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 1125–1128, doi: [10.1109/ICIP.2010.5649473](https://doi.org/10.1109/ICIP.2010.5649473).
- [59] C. DeLorenzo, X. Papademetris, L. H. Staib, K. P. Vives, D. D. Spencer, and J. S. Duncan, "Image-guided intraoperative cortical deformation recovery using game theory: Application to neocortical epilepsy surgery," *IEEE Trans. Med. Imag.*, vol. 29, no. 2, pp. 322–338, Feb. 2010, doi: [10.1109/TMI.2009.2027993](https://doi.org/10.1109/TMI.2009.2027993).
- [60] M. A. Bouteldja, M. Baadeche, and M. Batouche, "A novel approach for image denoising based on evolutionary game theory," in *Proc. 4th Int. Conf. Image Process. Theory, Tools Appl. (IPTA)*, Oct. 2014, pp. 1–6, doi: [10.1109/IPTA.2014.7001931](https://doi.org/10.1109/IPTA.2014.7001931).
- [61] P.-C. Hsiao and L.-W. Chang, "Image denoising with dominant sets by a coalitional game approach," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 724–738, Feb. 2013, doi: [10.1109/TIP.2012.2222894](https://doi.org/10.1109/TIP.2012.2222894).
- [62] Y. Chen and K. J. R. Liu, "Image denoising games," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 10, pp. 1704–1716, Oct. 2013, doi: [10.1109/TCSVT.2013.2255433](https://doi.org/10.1109/TCSVT.2013.2255433).
- [63] Z. Balint, B. Kiss, B. Magyari, and K. Simon, "Augmented reality and image recognition based framework for treasure hunt games," in *Proc. IEEE 10th Jubilee Int. Symp. Intell. Syst. Inform.*, Sep. 2012, pp. 147–152, doi: [10.1109/SISY.2012.6339504](https://doi.org/10.1109/SISY.2012.6339504).
- [64] A. M. Al-Ghaili, S. Mashohor, A. Ismail, and A. R. Ramli, "A new vertical edge detection algorithm and its application," in *Proc. Int. Conf. Comput. Eng. Syst.*, Nov. 2008, pp. 204–209, doi: [10.1109/ICCES.2008.4772997](https://doi.org/10.1109/ICCES.2008.4772997).
- [65] A. M. Al-Ghaili, S. Mashohor, A. R. Ramli, and A. Ismail, "Car license plate detection method for Malaysian plates-styles by using a Web camera," *J. Sci. Technol.*, vol. 18, no. 2, pp. 303–319, 2010.
- [66] M. Kumar Ray, D. Mitra, and S. Saha, "Simplified novel method for edge detection in digital images," in *Proc. Int. Conf. Signal Process., Commun., Comput. Netw. Technol.*, Jul. 2011, pp. 197–202, doi: [10.1109/ICSCCN.2011.6024543](https://doi.org/10.1109/ICSCCN.2011.6024543).
- [67] C. Hu, X. Liu, Z. Pan, and P. Li, "Automatic detection of single ripe tomato on plant combining faster R-CNN and intuitionistic fuzzy set," *IEEE Access*, vol. 7, pp. 154683–154696, 2019.



- [68] M. Mittal, A. Verma, I. Kaur, B. Kaur, M. Sharma, L. M. Goyal, S. Roy, and T.-H. Kim, "An efficient edge detection approach to provide better edge connectivity for image analysis," *IEEE Access*, vol. 7, pp. 33240–33255, 2019.
- [69] Y. Liu, Z. Xie, and H. Liu, "An adaptive and robust edge detection method based on edge proportion statistics," *IEEE Trans. Image Process.*, vol. 29, pp. 5206–5215, Mar. 2020.



**ABBAS M. AL-GHAILI** received the B.Eng. degree (Hons.) in computer engineering from the University of Science and Technology, Sana'a, Yemen, in 2005, and the M.Sc. and Ph.D. degrees in computer systems engineering from Universiti Putra Malaysia (UPM), Serdang, Malaysia, in 2009 and 2013, respectively.

He has been a Postdoctoral Researcher with the Institute of Informatics and Computing in Energy (IICE), Universiti Tenaga Nasional (UNITEN), Malaysia, since February 2018. His research interests include image processing, artificial intelligence, and energy informatics.

Dr. Al-Ghaili is a member of the International Association of Computer Science and Information Technology and the Universal Association of Computer and Electronics Engineers.



**HAIROLADENAN KASIM** is a Senior Lecturer with the College of Computing and Informatics (CCI), Universiti Tenaga Nasional (UNITEN), Kajang, Malaysia. His research interests include energy informatics, and energy and computing.



**NAIF MOHAMMED AL-HADA** received the B.Sc. degree in physics from Tamar University, Yemen, in 2002, and the M.Sc. degree in applied radiation physics and the Ph.D. degree in nanoscience and nanotechnology from Universiti Putra Malaysia (UPM), in 2011 and 2015, respectively.

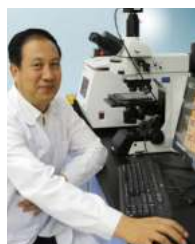
He was a Postdoctoral Researcher with the Department of Physics, UPM, from 2015 to 2019. He was a Visiting Researcher with Universiti Teknologi Malaysia (UTM), Malaysia, from June 2019 to October 2019. He has been a Lecturer of physics with Dezhou University (DZU), China, since October 2019. His research interests include nanoparticles synthesis and applications, metallic oxides nanostructures and its antibacterial activity, binary oxide nanostructures for solar cells and sensor applications, renewable energy, polymer composites/nanocomposites, conducting polymer nanocomposites, semiconductor nanotechnology, and applied radiation.



**MARINI OTHMAN** has served as the Director of the Institute of Informatics and Computing in Energy (IICE), Universiti Tenaga Nasional (UNITEN), Malaysia, from 2016 to 2019. Her research interests include information technology, energy informatics, Internet usage pattern, and data analytics.



**MUNEER AZIZ SALEH** received the B.Sc. degree in physics from Sana'a University, Yemen, the M.Sc. degree in applied physics from the National University of Malaysia, Malaysia, and the Ph.D. degree from Universiti Teknologi Malaysia (UTM), in January 2014. He is a Lecturer of the Nuclear Engineering Program, Faculty of Chemical and Energy Engineering, UTM. His research interests include environmental radiology, radiation dosimetry, nuclear power plant siting, radiation protection, and reactor physics.



**WANG JIHUA** is currently the Director of the Institute of Biophysics, Dezhou University, China. He is also a Ph.D. Tutor with Shandong Normal University. He is a Council Member of the Chinese Biophysics Society. He has undertaken more than ten scientific research funds, including funds from the National Science Foundation of China. He has published more than seventy articles in well-known journals such as *Nuclei Acids Research*, *Nature Communications*, *Biophysical Journal*, *Physics Review*, the *Journal of Theoretical Biology*, and *Biophysical Journal*. He held five national patents. He obtained the Special Allowance of the State Council and is nominated as a Scientist with Outstanding Contribution of Shandong. His research interests include long non-coding RNAs, intrinsically disordered proteins, and modern biophysical cross-discipline technologies.

...