

Morphology for hexagonal image processing: a comprehensive simulation analysis

Taner Cevik¹, Sajjad Nematzadeh², Jawad Rasheed^{3,4,5}, Abdulaziz Alshammari⁶

¹Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Arel University, Istanbul, Turkey

²Department of Computer Engineering, Faculty of Engineering, Istanbul Topkapi University, Istanbul, Turkey

³Department of Computer Engineering, Faculty of Engineering and Natural Sciences, Istanbul Sabahattin Zaim University, Istanbul, Turkey

⁴Deep Learning and Medical Image Analysis Laboratory, Bogazici University, Istanbul, Turkey

⁵Department of Software Engineering, Faculty of Engineering and Architecture, Istanbul Nisantasi University, Istanbul, Turkey

⁶College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University, Riyadh, Saudi Arabia

Article Info

Article history:

Received Sep 3, 2023

Revised Oct 20, 2023

Accepted Dec 2, 2023

Keywords:

Closing

Dilation

Erosion

Hexagonal image processing

Image analysis

Morphology

Opening

ABSTRACT

Morphological operators for binary and grayscale images are commonly used to eliminate noise, recognize contours or specific structures, and arrange shapes in image processing for physiological modeling and biomechanics applications. Even though morphology has been substantially developed in square-pixel-based-image-processing (SIP), no effort has been made to construct morphological operators in hexagonal-pixel-based-image-processing (HIP) yet. In this paper, we transform basic SIP-domain-morphological operators such as dilation, erosion, closing, and opening into HIP-domain and compare their performance with their SIP counterparts. It is the first time to give the fundamental morphological operators in the HIP domain. The operators developed in this paper initiate the research about morphology in the HIP domain by successfully filling a significant gap by eliminating HIP's lack of basic operators, thus capable of producing enhanced images for better analysis in anatomical models related to biology and medicine research fields.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Jawad Rasheed

Deep Learning and Medical Image Analysis Laboratory, Bogazici University

Istanbul, Turkey

Email: jawad.rasheed@boun.edu.tr

1. INTRODUCTION

The art of replicating human vision and converting it to computer vision is image processing. Indeed, the data in the light physical medium is continuous, and this continuous data is obtained using specific sensors. These sensors are utilized in square or rectangular arrays and differ in the light spectrum to which they are sensitive. Computers can only process digital data, even though light data is continuous. Continuous light data must therefore be sampled and digitized. Because square or rectangular sensor arrays are employed, downstream computer processing is developed accordingly. As a result, the smallest data unit of digitized data in a computer environment, the pixel, is built as a square. However, sampling light data on a hexagonal lattice and then processing it as a hexagon domain can change many things and produce promising results. For decades, hexagonal geometry has been studied. It was assumed until Hales [1], [2] proved otherwise that hexagons were the most significant way to split a plane into areas of equal area. Honeycombs are another natural hexagonal encounter with hexagonal geometry, in addition to the natural hexagonal arrangement of

photoreceptors in the fovea [3].

The use of hexagonal pixels for picture capture is a more recent innovation in hexagonal image processing (HIP). The hexagonal lattice construction has several advantages over its square counterpart. More excellent radial symmetry permits the application of circular symmetric kernels, which improves detection accuracy for both straight and curved edges and the homogeneity of the hexagonal lattice format, which gives local equality and uniqueness [4], [5].

A collection of operations on Euclidean space for quantitative description of geometrical structures has been proposed as mathematical morphology [6]. The set theory supports mathematical analysis, integral geometry, and lattice algebra. In recent years, mathematical morphology is becoming crucial in image processing and computer vision applications. The development of mathematical morphology is characterized by cross-fertilization between applications, methodologies, theories, and algorithms. It leads to several processing tools for image filtering, image segmentation and classification, image measurements, pattern recognition, or texture analysis and synthesis [7]. In industrial vision applications, mathematical morphology can be used to implement fast object recognition [8]–[10], image segmentation [11], [12], and industrial inspection [13]–[15].

The principles of mathematical morphology and its applications were first developed systematically in [16]–[18]. Further applications for signal and image processing were presented by Giardina and Dougherty [19], followed by the tutorial papers in [20], [21]. Extending the theory to gray-scale images was done in [17], [22], [23].

Despite its significant advantages, the hexagonal grid has not been widely used in computer vision and graphics until recently. The fundamental issue limiting the usage of the hexagonal picture structure is a lack of hardware for capturing and displaying hexagon-based images. Various attempts have mimicked hexagonal grids using regular rectangular grid technology. Simulation techniques include rectangular pixels, pseudo-hexagonal pixels, simulated hexagonal pixels, and virtual hexagonal pixels. While none of these simulation approaches can fully demonstrate the benefits of a proper hexagonal structure, their use provides us with practical tools for image processing on hexagonal grids. It allows us to continue our theoretical research into hexagonal structures in modern computer vision and graphics systems [24].

So far, no morphology studies have been performed in the HIP domain, and the most basic operations such as dilation and erosion have not been suggested on how to do it in the hexagonal domain. In mathematical morphology, the basic operations are dilation and erosion, combined to form other operations like opening and closing [14]. The kernel utilized in a morphological operation is the structuring element (SE). This study develops the HIP counterparts of these primary square image processing (SIP) morphological operations. The most basic SE is a straight line with an arbitrary length, either vertically or horizontally. Other more complex SEs can be easily derived from these straight-line SEs. The directions of these straight-line SEs on the SIP domain are mainly 0° , 45° , 90° , 135° while they are mainly 0° , 60° and 120° on the HIP domain. The article proposes these primary straight-length SEs on the HIP domain. While implementing morphology on the HIP domain firstly, the results of the morphological operations are carefully analyzed by taking into 65 similarity and dissimilarity metrics that have been proposed in the literature so far.

This article is categorized as follows: section 2 gives a brief introduction to the fundamentals of the HIP. Section 3 describes the principles of mathematical morphology. Section 4 presents the HIP equivalents of the essential SIP morphological operations. Section 5 introduces the dataset used for the testing, the experimental setup, and the 68 similarity-dissimilarity metrics for binary and four metrics for grey-scale that are taken into account and the discussions of the results achieved. Finally, section 6 states the conclusions we have drawn and the possible future directions of our research.

2. FUNDAMENTALS OF HEXAGONAL IMAGE PROCESSING

The digital images in the HIP domain contain regular hexagonal cells (Hexel), which correspond to the concept of the pixel in SIP. Due to their unique characteristics, hexels may be a viable alternative for conveying visual data. In an ideal scenario, a hexel-supported-camera-sensor should provide an excellent physical infrastructure to extract intensity and color information and display it on a hexel-supported-monitor. Few goods are publicly available at the time of authoring this article. As a result, we used mimic procedures to convert pixels to hexels. The differences in coordinate systems of SIP and HIP drove us to create a mechanism to project pixels on hexels. Figure 1 demonstrates the relations between cells and their neighbors. Hexels of upper and lower rows are calculated by averaging the corresponding pixels. Other values are copied directly.

The following sections explain the fundamental concepts of morphology customized for hexagonal systems.

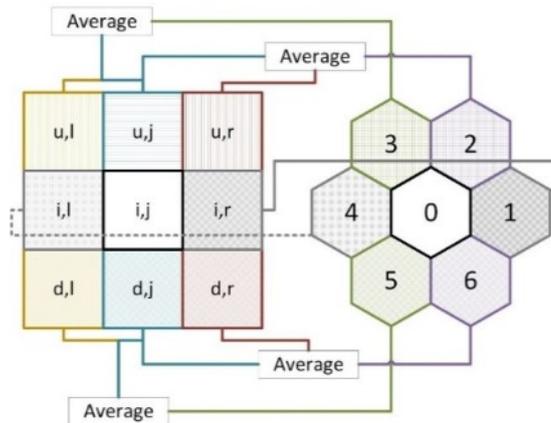


Figure 1. Identifying neighbors of a hexel using the pixel counterpart

3. MATHEMATICAL MORPHOLOGY

A set of pixels can be used to represent an image. Working with two images is how morphological operators can be visualized. The active image is the one that is being processed, and the other image, which is the kernel or so-called nonlinear filter, is referred to as the configuration element [25]. Matheron and Serra established a family of nonlinear filters called mathematical morphology in their mineralogical work in the early 1960s [26], [27]. Each configuration element has a unique design that acts as a probe or filter for the active image. The geometry of the configuration element determines the filter's effect on the image. All morphological filters combine two basic operators, erosion and expansion. The mathematical morphology was later generalized to the case of grayscale images [28], [29], whereas the original theory was considered for binary images.

3.1. Binary morphology

Pixels are added to a configuration element when it touches at least one pixel during the dilation process, expanding the item. The term dilation has different meanings in different places. For instance, if each point a of A is a seed that produces the flower B , the union of all the flowers is the dilation of A by B (by placing the origin of B at every a). Isotropic expansion techniques are commonly used in binary image processing, and dilation by disk structuring components corresponds to them. The 8-neighborhood operation of dilation by a small square (3×3), often known as "fill," "expand," or "grow," is easily achieved using adjacently connected array designs. Dilation increases the visibility of an object by filling small gaps in it. In (1) defines the dilation of the binary image A caused by configuring element B [14].

$$A \oplus_b B = \{c \in E^N \mid c = a + b \text{ for some } a \in A \text{ and } b \in B\} \quad (1)$$

where E^N is the set of all points $p = (x_1, x_2, \dots, x_N)$ in N -dimension Euclidean space, whilst A and B are subsets of E^N .

The properties of the dilation operation are:

Commutative: $A \oplus_b B = B \oplus_b A$

Associative: $A \oplus_b (B \oplus_b C) = (A \oplus_b B) \oplus_b C$

Translation invariance: $(A)_x \oplus_b B = (A \oplus_b B)_x$

Increasing: If $A \subseteq B$, then $A \oplus_b D \subseteq B \oplus_b D$

Distributive: $(A \cap B) \oplus_b C \subseteq (A \oplus_b C) \cap (B \oplus_b C)$

$(A \cup B) \oplus_b C = (A \oplus_b C) \cup (B \oplus_b C)$

The morphological dual of dilation is erosion. It combines two sets by subtracting set components using vector subtraction. When a configuration element comes into touch with at least one pixel during the erosion process, pixels are eliminated, effectively shrinking the image's objects. Erosion does not have the feature of commutativity. The locus of all centers c can be read as the erosion of A by B , with the translation

B wholly contained within set A . The object's size is reduced through erosion until only the most durable remains. The erosion of a binary image A by SE B is represented by (2) [14]:

$$A \ominus_b B = \{x \in E^N | x + b \in A \text{ for every } b \in B\} \tag{2}$$

The properties of the dilation operation are:

Non-commutative: $A \ominus_b B \neq B \ominus_b A$

Associative: $A \ominus_b (B \ominus_b C) = (A \ominus_b B) \ominus_b C$

Translation invariance: $(A)_x \ominus_b B = (A \ominus_b B)_x$

Increasing: If $A \subseteq B$, then $A \ominus_b D \subseteq B \ominus_b D$

Distributive: $(A \cap B) \ominus_b C = (A \ominus_b C) \cap (B \ominus_b C)$

$(A \cup B) \ominus_b C \supseteq (A \ominus_b C) \cup (B \ominus_b C)$

As a result of combining the dilation and erosion operations, we may create several composite morphological filters. The opening and closing operations, which are defined as the sequences of erosion-dilation and dilation-erosion, respectively using by (3) and (4), are the most common composite operators [30]:

$$A \circ B = (A \ominus B) \oplus B \tag{3}$$

$$A \bullet B = (A \oplus B) \ominus B \tag{4}$$

Idempotency is the primary property of the opening and closing operations. That is, the image does not change anymore when the open and close operations with the same SE are repeated. In (5) and (6) perform this operation.

$$A \circ B \circ B = A \circ B \tag{5}$$

$$A \bullet B \bullet B = A \bullet B \tag{6}$$

Figure 2 shows the primary SE used in dilation and erosion, while Figure 3 depicts the results of the dilation, erosion, opening, and closing operations by applying the primary SEs given in Figure 2.

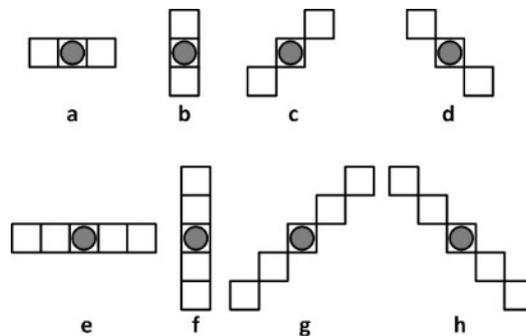


Figure 2. Basic SEs are used in dilation and erosion. (a) $SE_{0^\circ, L=3}$, (b) $SE_{90^\circ, L=3}$, (c) $SE_{45^\circ, L=3}$, (d) $SE_{135^\circ, L=3}$, (e) $SE_{0^\circ, L=5}$, (f) $SE_{90^\circ, L=5}$, (g) $SE_{45^\circ, L=5}$, and (h) $SE_{135^\circ, L=5}$.

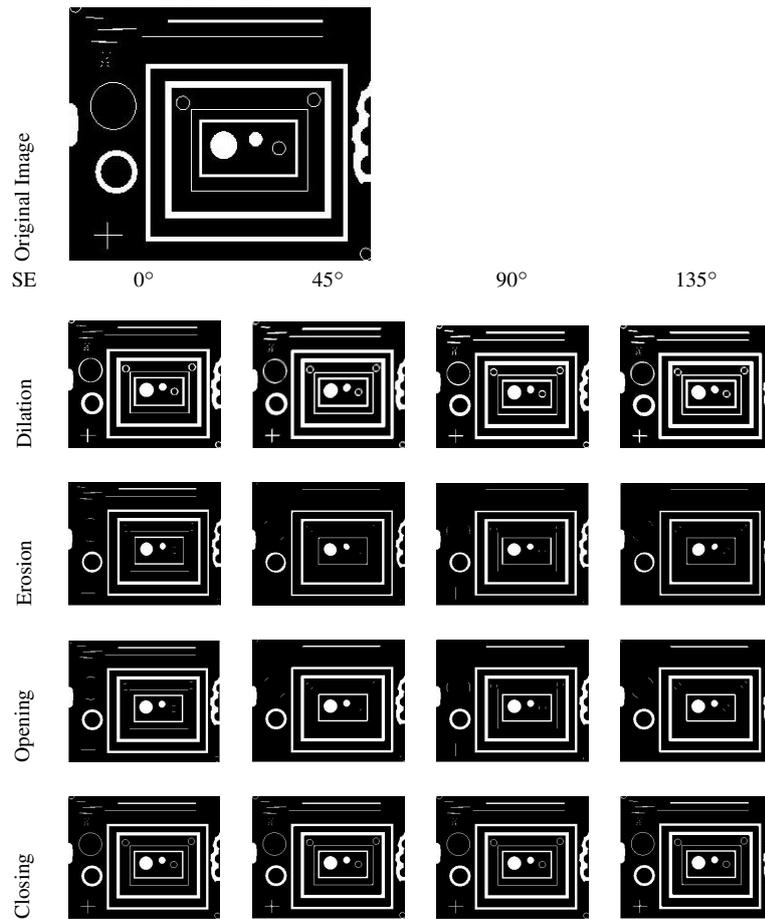


Figure 3. Results of the dilation, erosion, opening, and closing operations by applying the basic SEs given in Figure 2

3.2. Gray-scale morphology

A grayscale image is a three-dimensional set with the first two elements being the pixel's x and y coordinates and the third element being the grayscale value. Binary morphology can easily be stretched to grayscale morphology. The main differences come from the definitions of dilation and erosion, which are essential in other procedures. Logic operations are transformed into mathematical equivalents. In most circumstances, grayscale morphology is often limited to plane SE, resulting in simple dilation and erosion procedures using min and max functions [31], [32]. The dilation and erosion of a gray-scale image f by the SEs are respectively given in (7) and (8).

$$f \oplus s = \max\{(f)_p | p \in s\} = \max_{p \in s}(f)_p \quad (7)$$

$$f \ominus s = \min\{(f)_p | p \in s\} = \min_{p \in s}(f)_p \quad (8)$$

With this concept, the gray-scale opening and closing operations are defined in (9) and (10).

$$f \circ s = \max_{(a \in s)} \min_{(b \in s)} f(x - a + b)_0 \quad (9)$$

$$f \bullet s = \min_{(a \in s)} \max_{(b \in s)} f(x - a + b)_0 \quad (10)$$

Figure 4 illustrates the results of the dilation, erosion, opening, and closing operations on a gray-scale image by applying the basic SEs given in Figure 2.

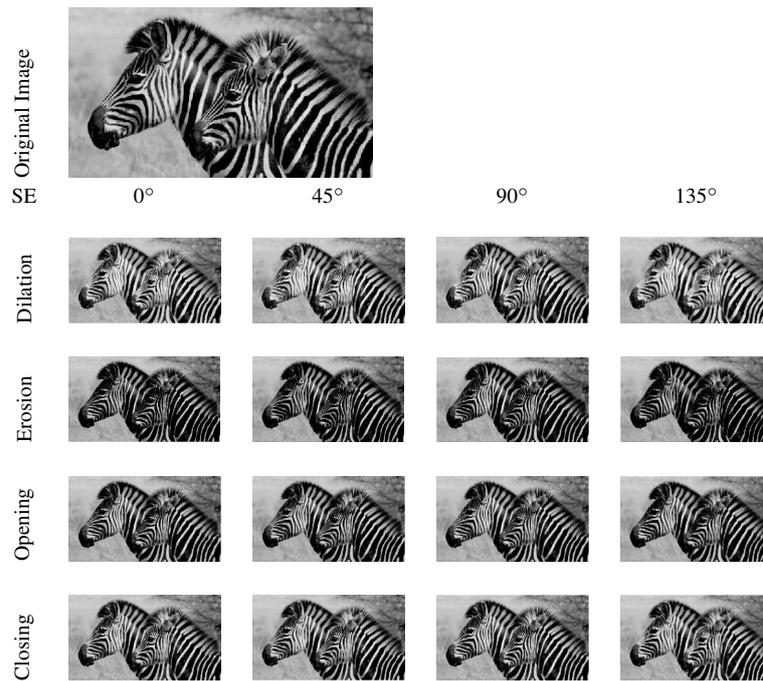


Figure 4. Results of the gray-scale dilation, erosion, opening, and closing operations by applying the basic SEs given in Figure 2

4. MORPHOLOGY FOR HIP

Since morphology is used in various fields of image processing such as image filtering, image segmentation, and classification, image measurements, pattern recognition, or texture analysis and synthesis, the definition of morphology in the HIP will fill a critical deficiency. Whereas in the classical SIP domain, there are 4-links or 8-links between each pixel and neighboring pixels, there are only six in HIP. Furthermore, based on these neighborhoods, SIP has 4 major angular directions, 0° , 45° , 90° , and 135° , while HIP has 3 angular directions, 0° , 60° , and 120° . Because of these structural differences, the basic SEs defined in SIP need to be redefined in HIP. To better express the relationships between pixels and their calculations, neighborhood-based indexing representation is preferred rather than ordinary numerical indexing, as illustrated in Figure 5. Based on this structure, 1-tier correspondents of the basic binary SEs illustrated in Figure 2 are depicted in Figure 6.

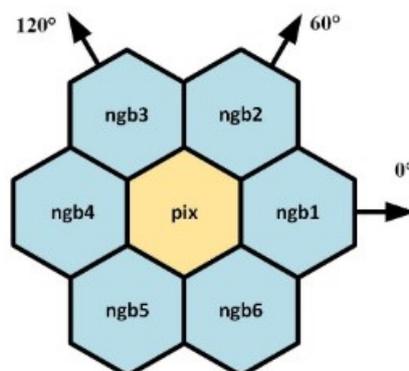


Figure 5. The 1-hop-neighborhood-based indexing in HIP

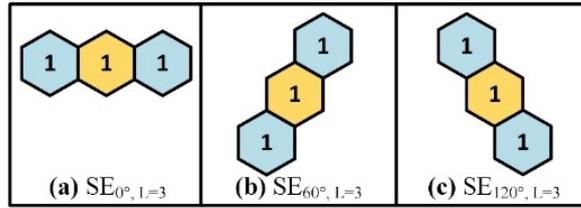


Figure 6. HIP 1-tier correspondents of the basic binary SEs that are illustrated in Figure 1

Algebra is not defined for HIP; for example, there is no corresponding matrix or convolution operator. Thus, all matrix-based algebraic operations are performed individually, pixel by pixel. Dilation and erosion operations in the HIP domain for the 1-tier architecture are performed as in (11)-(16).

$$f_p \oplus seHex_{0^\circ, L=3} = (p.ngb4V seHex_{0^\circ, L=3-1})V(pV seHex_{0^\circ, L=3_0})V(p.ngb1V seHex_{0^\circ, L=3+1}) \quad (11)$$

$$f_p \oplus seHex_{60^\circ, L=3} = (p.ngb5V seHex_{60^\circ, L=3-1})V(pV seHex_{60^\circ, L=3_0})V(p.ngb2V seHex_{60^\circ, L=3+1}) \quad (12)$$

$$f_p \oplus seHex_{120^\circ, L=3} = (p.ngb5V seHex_{120^\circ, L=3-1})V(pV seHex_{120^\circ, L=3_0})V(p.ngb2V seHex_{120^\circ, L=3+1}) \quad (13)$$

$$f_p \ominus seHex_{0^\circ, L=3} = (p.ngb4\Lambda seHex_{0^\circ, L=3-1})\Lambda(p\Lambda seHex_{0^\circ, L=3_0})\Lambda(p.ngb1\Lambda seHex_{0^\circ, L=3+1}) \quad (14)$$

$$f_p \ominus seHex_{60^\circ, L=3} = (p.ngb5\Lambda seHex_{60^\circ, L=3-1})\Lambda(p\Lambda seHex_{60^\circ, L=3_0})\Lambda(p.ngb2\Lambda seHex_{60^\circ, L=3+1}) \quad (15)$$

$$f_p \ominus seHex_{120^\circ, L=3} = (p.ngb5\Lambda seHex_{120^\circ, L=3-1})\Lambda(p\Lambda seHex_{120^\circ, L=3_0})\Lambda(p.ngb2\Lambda seHex_{120^\circ, L=3+1}) \quad (16)$$

With the same logic, the 1-tier opening and closing operations in the HIP are implemented in (17)-(22).

$$f_p \circ seHex_{0^\circ, L=3} = (f_p \ominus seHex_{0^\circ, L=3}) \oplus seHex_{0^\circ, L=3} \quad (17)$$

$$f_p \bullet seHex_{0^\circ, L=3} = (f_p \oplus seHex_{0^\circ, L=3}) \ominus seHex_{0^\circ, L=3} \quad (18)$$

$$f_p \circ seHex_{60^\circ, L=3} = (f_p \ominus seHex_{60^\circ, L=3}) \oplus seHex_{60^\circ, L=3} \quad (19)$$

$$f_p \bullet seHex_{60^\circ, L=3} = (f_p \oplus seHex_{60^\circ, L=3}) \ominus seHex_{60^\circ, L=3} \quad (20)$$

$$f_p \circ seHex_{120^\circ, L=3} = (f_p \ominus seHex_{120^\circ, L=3}) \oplus seHex_{120^\circ, L=3} \quad (21)$$

$$f_p \bullet seHex_{120^\circ, L=3} = (f_p \oplus seHex_{120^\circ, L=3}) \ominus seHex_{120^\circ, L=3} \quad (22)$$

The same reasoning applies to morphologic operations involving lengthier or larger SE, but the processes taken are naturally more complicated. Pixels' connectivity and neighborhood arrangement for the 2-tier architecture is shown in Figure 7. Based on this structure, 2-tier correspondents of the primary SIP SEs illustrated in Figure 2 are depicted in Figure 8.

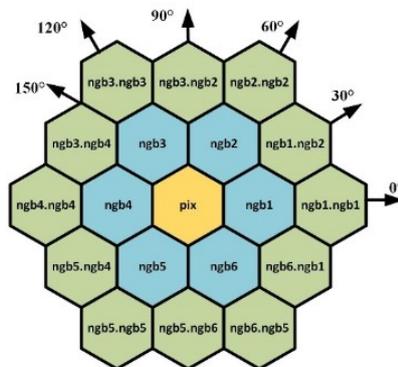


Figure 7. Representation of the 2-hop-neighborhood-based indexing in HIP

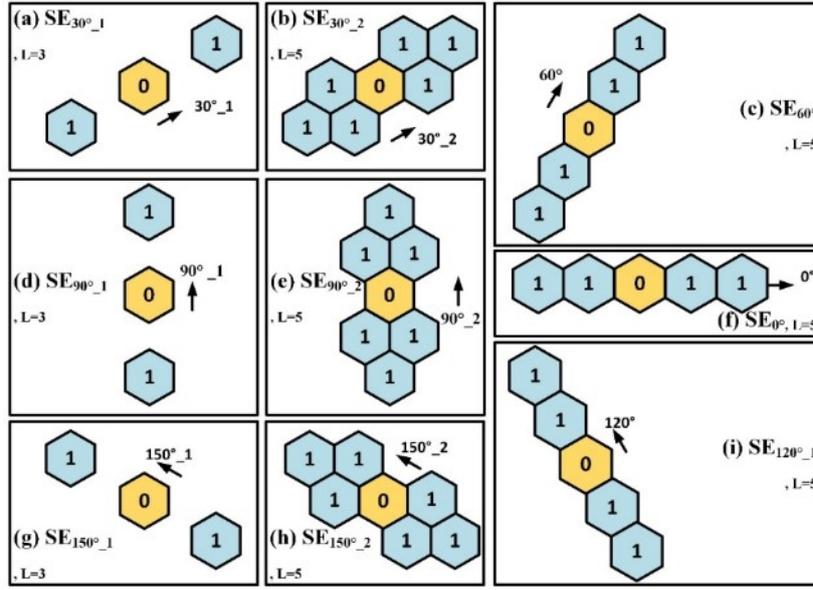


Figure 8. HIP 2-tier correspondents of the basic SEs are illustrated in Figure 2

Dilation and erosion operations in the HIP domain for the 2-tier architecture are performed in (23)-(31):

$$f_p \oplus seHex_{0^{\circ}, L=5} = (p.ngb4.ngb4V seHex_{0^{\circ}, L=5-2})V(p.ngb4V seHex_{0^{\circ}, L=5-1}) \quad (23)$$

$$V(pV seHex_{0^{\circ}, L=5_0})V(p.ngb1V seHex_{0^{\circ}, L=5+1})V(p.ngb1.ngb1V seHex_{0^{\circ}, L=5+2})$$

$$f_p \oplus seHex_{30^{\circ}_1, L=3} = (p.ngb5.ngb4V seHex_{30^{\circ}_1, L=3-1})V(pV seHex_{30^{\circ}_1, L=3_0}) \quad (24)$$

$$V(p.ngb1.ngb2V seHex_{30^{\circ}_1, L=3+1})$$

$$f_p \oplus seHex_{30^{\circ}_2, L=5} = (p.ngb5.ngb4V seHex_{30^{\circ}_2, L=5-2})V(p.ngb4V seHex_{30^{\circ}_2, L=5-1, -1}) \quad (25)$$

$$V(p.ngb5V seHex_{30^{\circ}_2, L=5-1, +1})V(pV seHex_{30^{\circ}_2, L=5_0})V(p.ngb2V seHex_{30^{\circ}_2, L=5+1, -1})$$

$$V(p.ngb1V seHex_{30^{\circ}_2, L=5+1, +1})V(p.ngb1.ngb2V seHex_{30^{\circ}_2, L=5+2})$$

$$f_p \oplus seHex_{60^{\circ}, L=5} = (p.ngb5.ngb5V seHex_{60^{\circ}, L=5-2})V(p.ngb5V seHex_{60^{\circ}, L=5-1}) \quad (26)$$

$$V(pV seHex_{60^{\circ}, L=5_0})V(p.ngb2V seHex_{60^{\circ}, L=5+1})V(p.ngb2.ngb2V seHex_{60^{\circ}, L=5+2})$$

$$f_p \oplus seHex_{90^{\circ}_1, L=3} = (p.ngb5.ngb6V seHex_{90^{\circ}_1, L=3-1})V(pV seHex_{90^{\circ}_1, L=3_0}) \quad (27)$$

$$V(p.ngb3.ngb2V seHex_{90^{\circ}_1, L=3+1})$$

$$f_p \oplus seHex_{90^{\circ}_2, L=5} = (p.ngb5.ngb6V seHex_{90^{\circ}_2, L=5-2})V(p.ngb5V seHex_{90^{\circ}_2, L=5-1, -1}) \quad (28)$$

$$V(p.ngb6V seHex_{90^{\circ}_2, L=5-1, +1})V(pV seHex_{90^{\circ}_2, L=5_0})V(p.ngb3V seHex_{90^{\circ}_2, L=5+1, -1})$$

$$V(p.ngb2V seHex_{90^{\circ}_2, L=5+1, +1})V(p.ngb3.ngb2V seHex_{90^{\circ}_2, L=5+2})$$

$$f_p \oplus seHex_{120^{\circ}, L=5} = (p.ngb6.ngb5V seHex_{120^{\circ}, L=5-2})V(p.ngb6V seHex_{120^{\circ}, L=5-1}) \quad (29)$$

$$V(pV seHex_{120^{\circ}, L=5_0})V(p.ngb3V seHex_{120^{\circ}, L=5+1})V(p.ngb3.ngb3V seHex_{120^{\circ}, L=5+2})$$

$$f_p \oplus seHex_{150^{\circ}_1, L=3} = (p.ngb6.ngb1V seHex_{150^{\circ}_1, L=3-1})V(pV seHex_{150^{\circ}_1, L=3_0}) \quad (30)$$

$$V(p.ngb3.ngb4V seHex_{150^{\circ}_1, L=3+1})$$

$$f_p \oplus seHex_{150^{\circ}_2, L=5} = (p.ngb6.ngb1V seHex_{150^{\circ}_2, L=5-2})V(p.ngb6V seHex_{150^{\circ}_2, L=5-1, -1}) \quad (31)$$

$$V(p.ngb1V seHex_{150^{\circ}_2, L=5-1, +1})V(pV seHex_{150^{\circ}_2, L=5_0})V(p.ngb4V seHex_{150^{\circ}_2, L=5+1, -1})$$

$$V(p.ngb3V seHex_{150^{\circ}_2, L=5+1, +1})V(p.ngb3.ngb4V seHex_{150^{\circ}_2, L=5+2})$$

The 2- tier opening and closing operations in HIP are implemented using by (32)-(49):

$$f_p \circ seHex_{0^\circ, L=5} = (f_p \ominus seHex_{0^\circ, L=5}) \oplus seHex_{0^\circ, L=5} \quad (32)$$

$$f_p \bullet seHex_{0^\circ, L=3} = (f_p \oplus seHex_{0^\circ, L=5}) \ominus seHex_{0^\circ, L=5} \quad (33)$$

$$f_p \circ seHex_{30_1^\circ, L=3} = (f_p \ominus seHex_{30_1^\circ, L=3}) \oplus seHex_{30_1^\circ, L=3} \quad (34)$$

$$f_p \bullet seHex_{30_1^\circ, L=3} = (f_p \oplus seHex_{30_1^\circ, L=3}) \ominus seHex_{30_1^\circ, L=3} \quad (35)$$

$$f_p \circ seHex_{30_2^\circ, L=5} = (f_p \ominus seHex_{30_2^\circ, L=5}) \oplus seHex_{30_2^\circ, L=5} \quad (36)$$

$$f_p \bullet seHex_{30_2^\circ, L=5} = (f_p \oplus seHex_{30_2^\circ, L=5}) \ominus seHex_{30_2^\circ, L=5} \quad (37)$$

$$f_p \circ seHex_{60^\circ, L=5} = (f_p \ominus seHex_{60^\circ, L=5}) \oplus seHex_{60^\circ, L=5} \quad (38)$$

$$f_p \bullet seHex_{60^\circ, L=5} = (f_p \oplus seHex_{60^\circ, L=5}) \ominus seHex_{60^\circ, L=5} \quad (39)$$

$$f_p \circ seHex_{90_1^\circ, L=3} = (f_p \ominus seHex_{90_1^\circ, L=3}) \oplus seHex_{90_1^\circ, L=3} \quad (40)$$

$$f_p \bullet seHex_{90_1^\circ, L=3} = (f_p \oplus seHex_{90_1^\circ, L=3}) \ominus seHex_{90_1^\circ, L=3} \quad (41)$$

$$f_p \circ seHex_{90_2^\circ, L=5} = (f_p \ominus seHex_{90_2^\circ, L=5}) \oplus seHex_{90_2^\circ, L=5} \quad (42)$$

$$f_p \bullet seHex_{90_2^\circ, L=5} = (f_p \oplus seHex_{90_2^\circ, L=5}) \ominus seHex_{90_2^\circ, L=5} \quad (43)$$

$$f_p \circ seHex_{120^\circ, L=5} = (f_p \ominus seHex_{120^\circ, L=5}) \oplus seHex_{120^\circ, L=5} \quad (44)$$

$$f_p \bullet seHex_{120^\circ, L=5} = (f_p \oplus seHex_{120^\circ, L=5}) \ominus seHex_{120^\circ, L=5} \quad (45)$$

$$f_p \circ seHex_{150_1^\circ, L=3} = (f_p \ominus seHex_{150_1^\circ, L=3}) \oplus seHex_{150_1^\circ, L=3} \quad (46)$$

$$f_p \bullet seHex_{150_1^\circ, L=3} = (f_p \oplus seHex_{150_1^\circ, L=3}) \ominus seHex_{150_1^\circ, L=3} \quad (47)$$

$$f_p \circ seHex_{150_2^\circ, L=5} = (f_p \ominus seHex_{150_2^\circ, L=5}) \oplus seHex_{150_2^\circ, L=5} \quad (48)$$

$$f_p \bullet seHex_{150_2^\circ, L=5} = (f_p \oplus seHex_{150_2^\circ, L=5}) \ominus seHex_{150_2^\circ, L=5} \quad (49)$$

In the following section, we present the results of the basic morphological operations that are implemented in the HIP domain. Besides, the images formed by morphology are compared with the original images. Also, similarity and dissimilarity analyses are performed, and the results are compared with their conjugates in the SIP domain.

5. EXPERIMENTAL RESULTS

The experimental analysis is performed on the Barcelona images for perceptual edge detection (BIPED) dataset [33] contains 250 high-definition outdoor images of 1280×720 pixels each. The dataset includes both the colored images and their edge maps called ground truths. An example-colored image and its edge map are shown in Figures 9(a) and 9(b).



(a)



(b)

Figure 9. An example-colored image and its edge-map from the BIDEPE dataset (a) original image and (b) edge-map

Morphology operations are performed in both domains as grayscale and binary. SIP-morphology is implemented on the original images, while HIP-morphology is applied on the HIP-converted versions. Therefore, before applying the HIP morphology, all images in the dataset are first converted to grayscale, and then both grayscale and binary edge maps are transferred to the HIP domain. Figure 10 illustrates the entire methodology and the individual steps implemented. After processing images using the morphology implementation, the results are analyzed for similarity and dissimilarity between the SIP and HIP-domain-processed images and the originals. For similarity-dissimilarity performance analysis of gray-scale morphology implementation, three benchmark metrics shown in Table 1 are taken into account [34], [35].

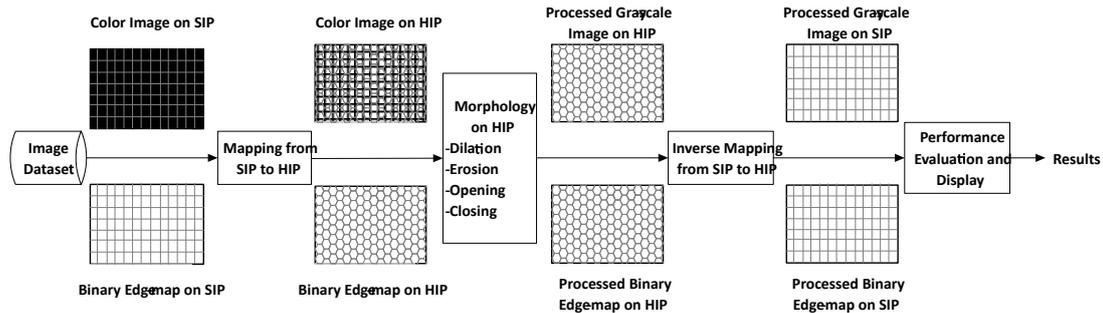


Figure 10. The illustration of the entire methodology

Table 1. Evaluation metrics for similarity-dissimilarity performance analysis of gray-scale morphology implementation

Tag	Description
SSIM	Structural similarity index measure - similarity index [34]
PSNR	Peak signal-to-noise ratio [35]
MSE	Root mean square error [35]

Since there will not be many variations in 1-tier morphology, the results are analyzed by applying 2-tier morphology to make the changes more evident. First of all, dilation, erosion, opening, and closing operations on binary images are applied in both domains, SIP and HIP, respectively. The SEs illustrated in Figure 2 are applied to the binary ground truth images in SIP, and those in Figure 8 are applied in the HIP domain. Figure 11 shows the similarity and dissimilarity analysis results measured during simulations. In this figure, algorithms in the HIP domain and SIP domain are denoted by HEX and SQ, respectively. Inherently, the similarity index between an image's original version and its processed one is desired to be low, indicating low distortion after morphology. As above-mentioned, dilation operation eventually stands for edge thickening while erosion indicates edge thinning. During dilation, among the hexagonal SEs, the highest deformation occurs after the implementation of $SE_{90^\circ, L=5}$ and $SE_{120^\circ, L=5}$ while the images are least distorted when $SE_{30^\circ, L=5}$ is applied. During erosion operation, the highest deformation occurs after the implementation of $SE_{150^\circ, L=5}$ while the images are least distorted when $SE_{0^\circ, L=5}$ is applied. In the SIP domain, $SE_{90^\circ, L=5}$ causes the highest deformation during both dilation and erosion.

Remind that morphological closing consists of the subsequent implementation of dilation and erosion operators. In contrast, morphological opening consists of implementing erosion and dilation operators subsequently. This time, $SE_{60^\circ, L=5}$ causes the highest distortion for both opening and closing operations. In the SIP domain, again, $SE_{90^\circ, L=5}$ causes the highest deformation during both opening and closing. Interestingly, there is a huge gap between the distortion effect of the SEs, $SE_{45^\circ, L=5}$ and $SE_{135^\circ, L=5}$ during dilation erosion. They distort the image less during dilation and severely during erosion. The exact determinations mentioned above are also valid for analyzes made on the measured grayscale morphology similarity-dissimilarity values (see Tables 2-5).

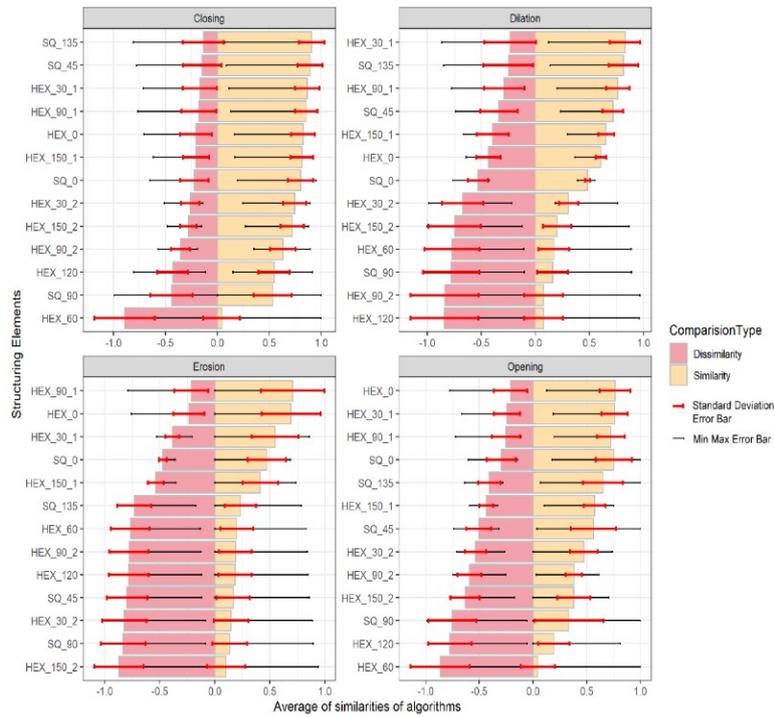


Figure 11. The similarity and dissimilarity analysis results of binary closing, dilation, erosion, and opening

Table 2. The similarity and dissimilarity analysis results measured after gray-scale dilation

	Hex_B_0_O	Hex_B_30_1_O	Hex_B_30_2_O	Hex_B_60_O	Hex_B_90_1_O	Hex_B_90_2_O	Hex_B_120_O
SSIM	0.751	0.853	0.491	0.312	0.672	0.300	0.296
PSNR	0.675	0.767	0.441	0.281	0.604	0.270	0.266
MSE	0.245	0.216	0.375	0.590	0.274	0.613	0.622
	Hex_B_150_1_O	Hex_B_150_2_O	SQ_B_0_O	SQ_B_45_O	SQ_B_90_O	SQ_B_135_O	
SSIM	0.788	0.510	0.667	0.857	0.146	0.844	
PSNR	0.708	0.458	0.600	0.771	0.132	0.759	
MSE	0.234	0.361	0.276	0.215	0.976	0.218	

Table 3. The similarity and dissimilarity analysis results measured after gray-scale erosion

	Hex_B_0_O	Hex_B_30_1_O	Hex_B_30_2_O	Hex_B_60_O	Hex_B_90_1_O	Hex_B_90_2_O	Hex_B_120_O
SSIM	0.681	0.938	0.777	0.189	0.777	0.136	0.191
PSNR	0.612	0.843	0.698	0.170	0.698	0.122	0.172
MSE	0.270	0.196	0.237	0.973	0.237	0.988	0.962
	Hex_B_150_1_O	Hex_B_150_2_O	SQ_B_0_O	SQ_B_45_O	SQ_B_90_O	SQ_B_135_O	
SSIM	0.569	0.762	0.464	0.368	0.270	0.363	
PSNR	0.511	0.684	0.417	0.330	0.243	0.326	
MSE	0.323	0.241	0.396	0.500	0.681	0.507	

Table 4. The similarity and dissimilarity analysis results measured after gray-scale closing

	Hex_B_0_O	Hex_B_30_1_O	Hex_B_30_2_O	Hex_B_60_O	Hex_B_90_1_O	Hex_B_90_2_O	Hex_B_120_O
SSIM	0.845	0.901	0.794	0.132	0.923	0.717	0.607
PSNR	0.759	0.809	0.713	0.119	0.829	0.644	0.545
MSE	0.218	0.205	0.232	0.991	0.200	0.257	0.303
	Hex_B_150_1_O	Hex_B_150_2_O	SQ_B_0_O	SQ_B_45_O	SQ_B_90_O	SQ_B_135_O	
SSIM	0.877	0.792	0.839	0.992	0.595	0.994	
PSNR	0.788	0.711	0.754	0.891	0.535	0.893	
MSE	0.210	0.233	0.220	0.186	0.309	0.185	

Table 5. The similarity and dissimilarity analysis results measured after gray-scale opening

	Hex_B_0_O	Hex_B_30_1_O	Hex_B_30_2_O	Hex_B_60_O	Hex_B_90_1_O	Hex_B_90_2_O	Hex_B_120_O
SSIM	0.881	0.882	0.589	0.236	0.784	0.396	0.247
PSNR	0.729	0.793	0.530	0.212	0.704	0.356	0.222
MSE	0.227	0.209	0.313	0.780	0.235	0.465	0.744
	Hex_B_150_1_O	Hex_B_150_2_O	SQ_B_0_O	SQ_B_45_O	SQ_B_90_O	SQ_B_135_O	
SSIM	0.807	0.572	0.780	0.654	0.258	0.658	
PSNR	0.725	0.514	0.701	0.588	0.232	0.591	
MSE	0.228	0.322	0.236	0.282	0.714	0.280	

Tables 6 and 7 show the histogram-similarity-and-dissimilarity-analysis-results measured after gray-scale dilation, erosion, closing, and opening operations implemented on both SIP and HIP domains. As an alternative, similarity analysis between an original image and its SIP, HIP-morphology implemented versions is conducted by histogram-similarity comparison. After applying a process to an image, histogram-similarity-analysis provides essential information about how much the original deviates from its original version. During histogram-similarity-analysis on grayscale images, six metrics [36] (correlation, Spearman, Kullback-Leibler-divergence, Chi-square-statistic, histogram intersection, and quadratic form distance statistics) were taken into account. The measured values for these measures drop as the deterioration following morphological treatment increases. The most severe image degradation occurs after applying morphology using the HIP domain’s HEX_GS_30_1, HEX_GS_150_1 operator, and the SIP domain’s SQ_GS_45, SQ_GS_90 operators; because these operators work on hexels in mid-directions in the HIP domain, but in the SIP domain, they work on pixels in diagonal directions. Hexels and pixels in the directions mentioned above have more details than others; hence, changing their intensity causes more distortion. Figures 12-15 show how an image’s histogram evolves following dilation, erosion, closing, and opening operations applied in the SIP and HIP domains, respectively, to better represent similarity and dissimilarity analysis via histograms. Thus, this approach can further be tested for various problems such as [37], and [38].

Table 6. The histogram-similarity-and-dissimilarity-analysis-results measured after gray-scale dilation and erosion

	Correlation	Spearman	Kullback_leibler divergence	Chi_square_statistics	Histogram_intersection	Q_F_S
HEX_GS_0_D	0.1452	0.0138	16422	5713	0.2083	7459
HEX_GS_30_1_D	0.1052	0.0162	12127	4532	0.1821	6505
HEX_GS_30_2_D	0.2000	0.0249	23221	7564	0.2471	8602
HEX_GS_60_D	0.1822	0.0192	22003	7349	0.2391	8294
HEX_GS_90_1_D	0.1090	0.0098	13130	4768	0.1900	6652
HEX_GS_90_2_D	0.2122	0.0220	25472	8208	0.2566	8840
HEX_GS_120_D	0.1782	0.0197	21493	7360	0.2377	8227
HEX_GS_150_1_D	0.1021	0.0160	12025	4574	0.1807	6444
HEX_GS_150_2_D	0.1968	0.246	22760	7546	0.2451	8544
SQ_GS_0_D	0.1507	0.0256	INF	6085	0.2139	7593
SQ_GS_45_D	0.1019	0.0038	11949	4487	0.1819	6437
SQ_GS_90_D	0.1831	0.0133	INF	7486	0.2401	8312
SQ_GS_135_D	0.1062	0.0035	12325	45445	0.1839	6539
HEX_GS_0_E	0.1141	0.0625	INF	5767	0.2404	9723
HEX_GS_30_1_E	0.1052	0.0162	12127	4532	0.1821	6505
HEX_GS_30_2_E	0.1640	0.1811	INF	9069	0.2995	12426
HEX_GS_60_E	0.1266	0.1651	INF	7675	0.2789	11047
HEX_GS_90_1_E	0.1090	0.0098	13130	4768	0.1900	6652
HEX_GS_90_2_E	0.1713	0.1440	INF	9316	0.3148	12932
HEX_GS_120_E	0.1259	0.1700	INF	7747	0.2780	11030
HEX_GS_150_1_E	0.0588	0.1214	INF	4504	0.1943	7357
HEX_GS_150_2_E	0.1600	0.1824	INF	9053	0.2986	12333
SQ_GS_0_E	0.1147	0.1477	INF	61870	2477	9914
SQ_GS_45_E	0.0581	0.1270	INF	4710	0.1952	7370
SQ_GS_90_E	0.1264	0.1021	INF	7600	0.2821	11081
SQ_GS_135_E	0.0608	0.1415	INF	4767	0.1985	7497

Table 7. The histogram-similarity-and-dissimilarity-analysis-results measured after gray-scale closing and opening

	Correlation	Spearman	Kullback_leibler divergence	Chi_square_statistics	Histogram_intersection	Q_F_S
HEX_GS_0_C	0.1141	0.0625	INF	5767	0.2404	9723
HEX_GS_30_1_C	0.0621	0.0980	INF	4450	0.1969	7463
HEX_GS_30_2_C	0.1640	0.1811	INF	9069	0.2995	12426
HEX_GS_60_C	0.1266	0.1651	INF	7675	0.2789	11047
HEX_GS_90_1_C	0.0638	0.0610	INF	4145	0.2015	7613
HEX_GS_90_2_C	0.1713	0.1440	INF	9316	0.3148	12932
HEX_GS_120_C	0.1713	0.1440	INF	9316	0.3148	12932
HEX_GS_150_1_C	0.0588	0.1214	INF	4504	0.1943	7357
HEX_GS_150_2_C	0.1600	0.1824	INF	9053	0.2986	12333
SQ_GS_0_C	0.0767	0.0220	INF	2760	0.1366	5496
SQ_GS_45_C	0.0479	0.0010	4639	1785	0.1063	4374
SQ_GS_90_C	0.0998	0.0149	INF	3562	0.1597	6227
SQ_GS_135_C	0.0488	0.0008	4690	1808	0.1091	4415
HEX_GS_0_O	0.0329	0.0422	INF	1550	0.1170	4730
HEX_GS_30_1_O	0.0145	0.0468	INF	1298	0.0912	3485
HEX_GS_30_2_O	0.0408	0.0719	INF	2379	0.1354	5571
HEX_GS_60_O	0.0282	0.1649	INF	2196	0.1287	5269
HEX_GS_90_1_O	0.0125	0.0842	INF	1314	0.1007	3537
HEX_GS_90_2_O	0.0459	0.1353	INF	2724	0.1558	6318
HEX_GS_120_O	0.0377	0.1693	INF	2767	0.1483	5762
HEX_GS_150_1_O	0.0117	0.0482	INF	1288	0.0904	3276
HEX_GS_150_2_O	0.0365	0.0790	INF	2266	0.1309	5332
SQ_GS_0_O	0.0330	0.1203	INF	1929	0.1228	4959
SQ_GS_45_O	0.0116	0.1271	INF	1769	0.0959	3399
SQ_GS_90_O	0.0378	0.0980	INF	3049	0.1540	6132
SQ_GS_135_O	0.0137	0.1410	INF	1822	0.0976	3561

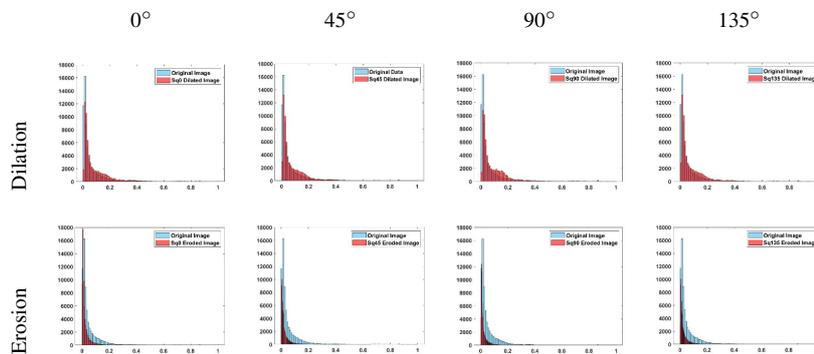


Figure 12. Changes in the histogram after dilation and erosion operations implemented in the SIP domain

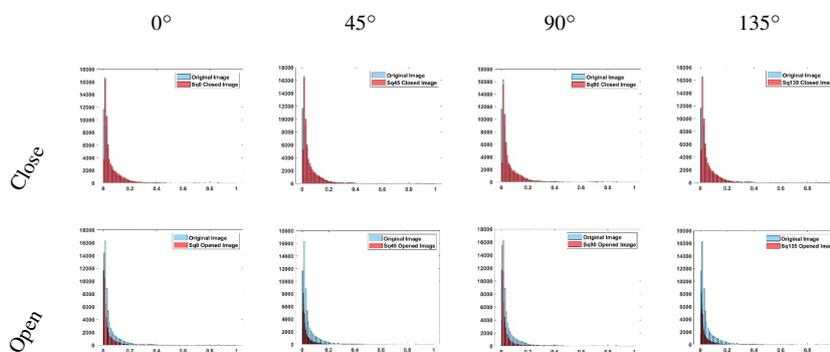


Figure 13. Changes in the histogram after closing and opening operations implemented in the SIP domain

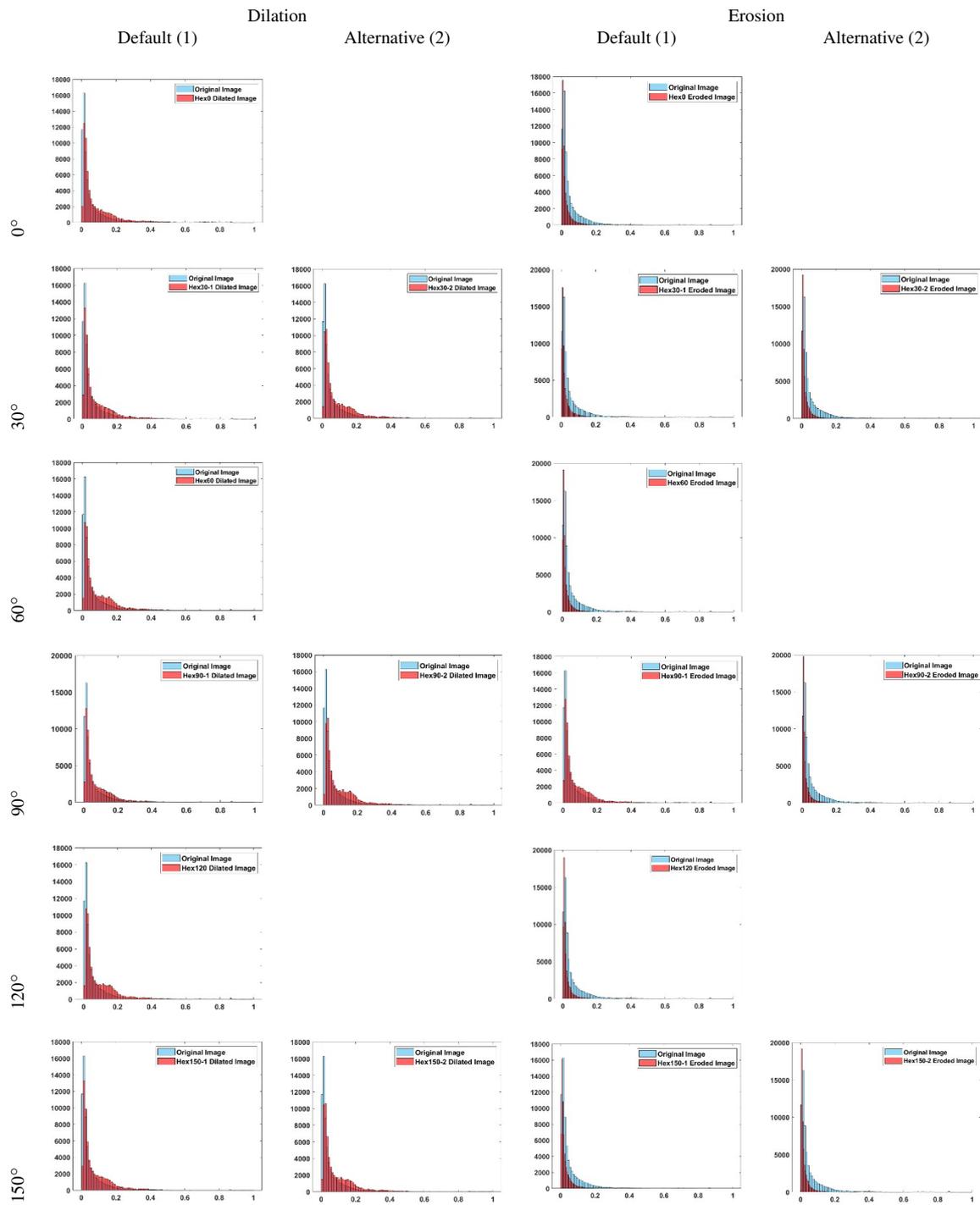


Figure 14. Changes in the histogram after dilation and erosion operations implemented in the HIP domain

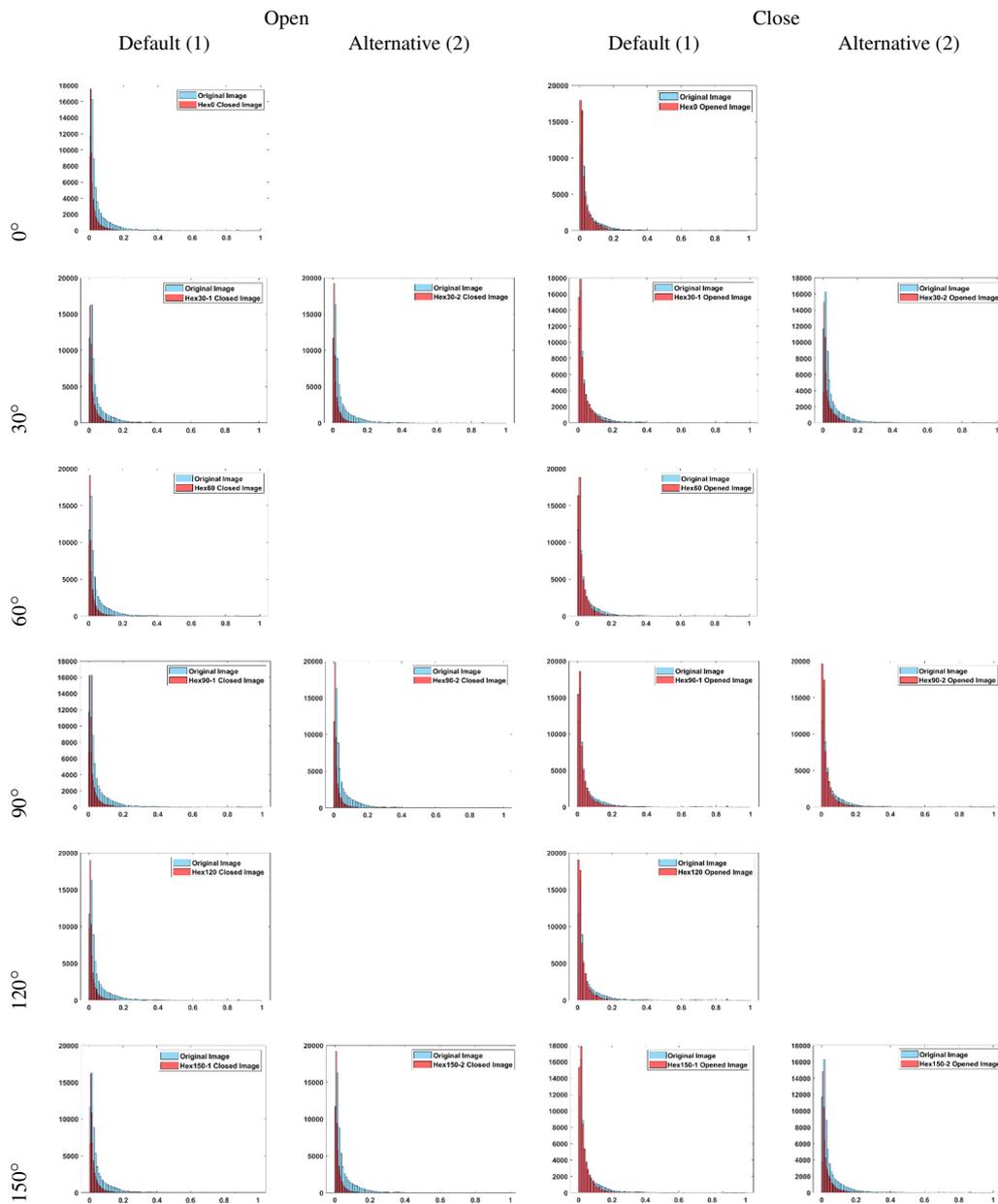


Figure 15. Changes in the histogram after closing and opening operations implemented in the HIP domain

6. CONCLUSION

In image processing, for binary and grayscale images, morphological operators are commonly used to eliminate noise, recognize contours or specific structures, and arrange shapes. Although morphology has been substantially developed in SIP, no effort has been made to construct morphological operators in the hexagonal domain -HIP- yet. In this paper, we transform basic SIP-domain-morphological operators such as dilation, erosion, closing, and opening into HIP-domain and compare their performance with their SIP counterparts in terms of the level of distortion occurring on the images after the process. Comprehensive simulations are conducted to measure distortion that arises after the operations, dilation, erosion, opening, and closing. The level of distortion is measured by considering the maintained similarity between the original images and those of their morphologically processed versions. While investigating how the morphology works in the HIP, both the binary and grayscale implementations are discussed. The results of our extensive simulations, the proposed HIP morphological operators work successfully and fill a significant gap by eliminating the lack of basic morphol-

ogy operators in the HIP. Complementing this significant step is achieved by presenting the HIP equivalence of one of the major constituents of image processing. After successfully implementing the basic morphology operators in the HIP domain, we are considering moving towards performing more complex operations using these operators in image processing in the HIP domain.

REFERENCES

- [1] T. Cevik, M. Fettaoglu, N. Cevik, and S. Yilmaz, "FTSH: a framework for transition from square image processing to hexagonal image processing," *Multimedia Tools & Applications*, vol. 79, no. 11–12, pp. 7021–7048, 2020, doi: 10.1007/s11042-019-08487-z.
- [2] T. C. Hales, "Cannonballs and honeycombs," *Notices of the American Mathematical Society*, vol. 47, no. 4, pp. 440–449, 2000.
- [3] T. C. Hales, "The honeycomb conjecture," *Discrete and Computational Geometry*, vol. 25, no. 1, pp. 1–22, 2001, doi: 10.1007/s004540010071.
- [4] S. Coleman, B. Scotney, and B. Gardiner, "Processing hexagonal images in a virtual environment," in *International Conference on Image Analysis and Processing*, 2009, pp. 920–928, doi: 10.1007/978-3-642-04146-4_98.
- [5] J. D. Allen, "Perfect reconstruction filter banks for the hexagon grid," in *2005 5th International Conference on Information Communications & Signal Processing*, 2005, pp. 73–76, doi: 10.1109/ICICS.2005.1689007.
- [6] J. G. Postaire, R. D. Zhang, and C. L. -Botte, "Cluster analysis by binary morphology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 2, pp. 170–180, 1993, doi: 10.1109/34.192490.
- [7] J. Debayle and J. -C. Pinoli, "Multiscale image filtering and segmentation by means of adaptive neighborhood mathematical morphology," in *IEEE International Conference on Image Processin*, 2005, vol. 3, doi: 10.1109/ICIP.2005.1530447.
- [8] S. R. Sternberg, "Industrial morphology," in *Applications of Digital Image Processing VII*, 1984, vol. 504, pp. 202–215, doi: 10.1117/12.944864.
- [9] F. Y. Shih and O. R. Mitchell, "Skeletonization and distance transformation by greyscale morphology," in *Automated Inspection and High-Speed Vision Architectures*, 1988, vol. 849, pp. 80–86, doi: 10.1117/12.942827.
- [10] F. Y. Shih and O. R. Mitchell, "Automated fast recognition and location of arbitrarily shaped objects by image morphology," in *The Computer Society Conference on Computer Vision and Pattern Recognition*, 1988, pp. 774–779, doi: 10.1109/CVPR.1988.196322.
- [11] S. R. Sternberg, "Biomedical image processing," *Computer*, vol. 16, no. 1, pp. 22–34, 1983, doi: 10.1109/MC.1983.1654163.
- [12] F. Y. Shih and O. R. Mitchell, "Industrial parts recognition and inspection by image morphology," in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, 1988, pp. 1764–1766, doi: 10.1109/ROBOT.1988.12321.
- [13] M. Bastir et al., "Workflows in a virtual morphology lab: 3D scanning, measuring, and printing," *Journal of anthropological sciences*, vol. 97, pp. 107–134, 2019.
- [14] C. C. Pu and F. Y. Shih, "Threshold decomposition of gray-scale soft morphology into binary soft morphology," *Graphical Models and Image Processing*, vol. 57, no. 6, pp. 522–526, 1995, doi: 10.1006/gmip.1995.1042.
- [15] S. R. Sternberg, "A morphological approach to finished surface inspection," in *1985 IEEE International Conference on Robotics and Automation*, 1985, vol. 2, pp. 462–465, doi: 10.1109/ROBOT.1985.1087252.
- [16] B. D. Ripley and G. Matheron, "Random sets and integral geometry," *Journal of the Royal Statistical Society. Series A (General)*, vol. 139, no. 2, pp. 277–278, 1976, doi: 10.2307/2345196.
- [17] C. O'Connor, "Image analysis and mathematical morphology," *Computers & Mathematics with Applications*, vol. 14, no. 9–12, p. 949, 1987, doi: 10.1016/0898-1221(87)90243-4.
- [18] W. Nagel, "Image analysis and mathematical morphology. Volume 2: theoretical advances," *Journal of Microscopy*, vol. 152, no. 2, pp. 597–597, 1988, doi: 10.1111/j.1365-2818.1988.tb01425.x.
- [19] C. R. Giardina and E. R. Dougherty, *Morphological methods in image and signal processing*. New Jersey, USA: Prentice Hall, 1988.
- [20] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE transactions on pattern analysis and machine intelligence*, vol. 9, no. 4, pp. 532–550, 1987.
- [21] P. Maragos, "Tutorial on advances in morphological image processing and analysis," in *Visual Communications and Image Processing*, 1986, pp. 64–74, doi: 10.1117/12.937249.
- [22] S. R. Sternberg, "Cellular computers and biomedical image processing," in *Lecture Notes in Medical Informatics*, 1982, pp. 294–319, doi: 10.1007/978-3-642-93218-2_27.
- [23] S. R. Sternberg, "Grayscale morphology," *Computer Vision, Graphics, and Image Processing*, vol. 35, no. 3, pp. 333–355, 1986, doi: 10.1016/0734-189X(86)90004-6.
- [24] X. He and W. Jia, "Hexagonal structure for intelligent vision," in *2005 International Conference on Information and Communication Technologies*, 2005, pp. 52–64, doi: 10.1109/ICICT.2005.1598543.
- [25] F. Y.-C. Shih and O. R. Mitchell, "A mathematical morphology approach to Euclidean distance transformation," *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 197–204, 1992, doi: 10.1109/83.136596.
- [26] P. A. Haas, G. Matheron, and J. Serra, "Morphologie mathématique et granulométries en place," *Annales des Mines*, vol. 11, pp. 736–753, 1967.
- [27] F. Y. Shih, *Image processing and mathematical morphology: fundamentals and applications*. Boca Raton: CRS Press, 2009.
- [28] Y. Nakagawa and A. Rosenfeld, "A note on the use of local min and max operations in digital picture processing," *IEEE Transactions on Systems Man and Cybernetics*, vol. 8, no. 8, pp. 632–635, 1987, doi: 10.1109/TSMC.1978.4310040.
- [29] F. G. B. D. Natale and G. Boato, "Detecting morphological filtering of binary images," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1207–1217, 2017, doi: 10.1109/TIFS.2017.2656472.
- [30] J. D. Trivedi, S. D. Mandalapu, and D. H. Dave, "Vision-based real-time vehicle detection and vehicle speed measurement using morphology and binary logical operation," *Journal of Industrial Information Integration*, vol. 27, 2022, doi: 10.1016/j.jii.2021.100280.
- [31] K. Parvati, B. S. P. Rao, and M. M. Das, "Image segmentation using gray-scale morphology and marker-controlled watershed transformation," *Discrete Dynamics in Nature and Society*, vol. 2008, pp. 1–8, 2008, doi: 10.1155/2008/384346.
- [32] O. Déforges, N. Normand, and M. Babel, "Fast recursive grayscale morphology operators: From the algorithm to the pipeline

- architecture,” *Journal of Real-Time Image Processing*, vol. 8, no. 2, pp. 143–152, 2013, doi: 10.1007/s11554-010-0171-8.
- [33] X. Soria, E. Riba, and A. Sappa, “Dense extreme inception network: towards a robust CNN model for edge detection,” in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 1912–1921, doi: 10.1109/WACV45572.2020.9093290.
- [34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004, doi: 10.1109/TIP.2003.819861.
- [35] A. Boutemedjet, C. Deng, and B. Zhao, “Robust approach for nonuniformity correction in infrared focal plane array,” *Sensors*, vol. 16, no. 11, 2016, doi: 10.3390/s16111890.
- [36] Y. Rubner, C. Tomasi, and L. J. Guibas, “Earth mover’s distance as a metric for image retrieval,” *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000, doi: 10.1023/A:1026543900054.
- [37] T. Cevik, N. Cevik, J. Rasheed, A. M. A. -Mahfouz, and O. Osman, “Facial recognition in hexagonal domain—a frontier approach,” *IEEE Access*, vol. 11, pp. 46577–46591, 2023, doi: 10.1109/ACCESS.2023.3274840.
- [38] M. S. Farooq et al., “A conceptual multi-layer framework for the detection of nighttime pedestrian in autonomous vehicles using deep reinforcement learning,” *Entropy*, vol. 25, no. 1, 2023, doi: 10.3390/e25010135.

BIOGRAPHIES OF AUTHORS



Taner Cevik    received the B.Sc. degree in computer engineering from Istanbul Technical University, Istanbul in 2001, and completed Ph.D. at Istanbul University in 2012. He joined the Department of Computer Engineering at Istanbul Arel University in 2023 and continues to work as Professor. His research interests are image processing, machine learning, and wireless communications. He can be contacted at email: tanercevik@arel.edu.tr.



Sajjad Nematzadeh    is a Ph.D. candidate and serving as a research assistant in the Department of Computer Engineering at Istanbul Topkapi University. His research interests include artificial intelligence and bioinformatics. He can be contacted at email: sajjadnematzadeh@topkapi.edu.tr.



Jawad Rasheed    received a B.S. degree in telecommunication engineering from the FAST-NUCES, Pakistan. He then moved to Turkey, where he earned an M.S. degree in electrical and electronics engineering and a Ph.D. degree in computer engineering. He was a gold medalist and was awarded the Academic Excellence Award for securing straight A’s in O’ Level exams held by Cambridge University. Later, he also received a prestigious doctorate and research scholarship for his Ph.D. studies (for three years). He is currently working as a senior researcher at the Deep Learning and Medical Image Analysis Laboratory, Bogazici University, Turkey. He is the author/co-author of more than 50 articles published in well-reputed journals and highly-ranked conferences. He serves as guest/lead-guest/topic editor for special issues at the *Symmetry*, *Mathematics*, *Healthcare*, *Applied Sciences*, *Electronics*, *Healthcare*, and *Journal of Sensor and Actuator Networks*. Recently, he served as a book editor for Springer LNDECT (FoNeS-IoT) and Springer LNNS (FoNeS-AIoT). In addition, he is the general chair of IEEE ICAIoT and IEEE FoNeS-AIoT and chairs the technical program committee of Springer FoNeS-IoT 2021. His research interests include artificial intelligence and image processing. pattern recognition, IoT, and data analytics. He can be contacted at email: jawad.rasheed@boun.edu.tr.



Abdulaziz Alshammari    received his Ph.D. degree in Computer Science and Informatics from Oakland University, Rochester, USA. Currently, he is serving as an assistant professor at Imam Mohammad Ibn Saud Islamic University, Saudi Arabia. His research interest includes software engineering, networking, human-computer interaction, and data science. He can be contacted at email: aashammari@imamu.edu.sa.