

Image Segmentation Based on Block Level and Hybrid Directional Local Extrema

Ghanshyam Raghuwanshi¹, Yogesh Gupta², Deepak Sinwar¹, Dilbag Singh³, Usman Tariq⁴,
Muhammad Attique⁵, Kuntha Pin⁶ and Yunyoung Nam^{7,*}

¹Manipal University Jaipur, Jaipur, 302026, India

²BML Munjal University, Gurugram, India

³Bennett University, Greater Noida, India

⁴College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Khraj, Saudi Arabia

⁵Department of Computer Science, HITEC University Taxila, Taxila, Pakistan

⁶Department of ICT Convergence, Soonchunhyang University, Asan 31538, Korea

⁷Department of Computer Science and Engineering, Soonchunhyang University, Asan, Korea

*Corresponding Author: Yunyoung Nam. Email: ynam@sch.ac.kr

Received: 08 March 2021; Accepted: 05 May 2021

Abstract: In the recent decade, the digitalization of various tasks has added great flexibility to human lifestyle and has changed daily routine activities of communities. Image segmentation is a key step in digitalization. Segmentation plays a key role in almost all areas of image processing, and various approaches have been proposed for image segmentation. In this paper, a novel approach is proposed for image segmentation using a nonuniform adaptive strategy. Region-based image segmentation along with a directional binary pattern generated a better segmented image. An adaptive mask of 8×8 was circulated over the pixels whose bit value was 1 in the generated directional binary pattern. Segmentation was performed in three phases: first, an image was divided into sub-images or image chunks; next, the image patches were taken as input, and an adaptive threshold was generated; and finally the image chunks were processed separately by convolving the adaptive mask on the image chunks. Gradient and Laplacian of Gaussian algorithms along with directional extrema patterns provided a double check for boundary pixels. The proposed approach was tested on chunks of varying sizes, and after multiple iterations, it was found that a block size of 8×8 performs better than other chunks or block sizes. The accuracy of the segmentation technique was measured in terms of the count of ill regions, which were extracted after the segmentation process.

Keywords: Image segmentation; HDEP; block-level processing; adaptive threshold

1 Introduction

Image segmentation is the key stage and works as a vital step for many image-processing applications [1]. In an image segmentation process, an image is divided into uniform or



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

homogeneous regions, and these regions can be decided by some common attributes such as the position of image pixels, color properties, patterns, and shape features [2]. Human perception of image geometry or object identification and the gap in understanding the system can be minimized using proper image segmentation. Filling this semantic gap remains the most difficult challenge, and therefore it has been deeply analyzed and investigated in recent years [3]. An image can be composed of different types of intensity values. An image can be categorized into background and foreground regions based on the intensity values. Image analysis can be more effective when using image segmentation techniques. Previously, various approaches have been introduced for segmenting an image into meaningful areas called objects [4,5]. Segmentation techniques are categorized into two types: similarity-based and discontinuity-based segmentation. Segmentation plays a key role in almost all areas such as image retrieval, object identification, medical image processing [6,7], image de-noising, and remote sensing image processing [8,9]. Recently, deep learning [10,11] and constrained based image segmentation were proposed for improving image segmentation [12]. Images, in general, have richer contents, and the area of interest in images cannot be determined accurately without a proper image segmentation technique. To achieve a perfect image analysis i.e., to better identify the target, it is critical to filter out unwanted content from the image [13]. Image segmentation can be done using image features such as texture, homogeneity, and intensity value of a pixel. Foreground intensity pixels refer to the specific region or target of interests in the image. To extract these target areas from the image, proper segmentation techniques are required. Sometimes the object can be of lighter intensities, whereas the background pixels are of higher intensities, leading to inaccurately segmented regions [14]. This issue can be resolved by adopting a dynamic or adaptive thresholding technique, as well as by processing the image at a block or region-level. Each block can be processed independently using its own threshold. At a later stage, adjacent blocks or regions can be merged based on their block or region properties or boundary value analysis. Region growing and merging is also one of the most widely used approaches for image segmentation. The basic idea behind this approach is to start with the initial properties of the block or the seed value provided by a user [15]. Various approaches have been published to date for improving results [16] using the region growing and merging approach [17,18]. After image segmentation, the results are evaluated based on some parameters. More importantly, the next stage can only be started if an image is segmented by a more optimal segmentation technique. There are two ways to segment an image either at a region or block level or at a global level. A globally extracted threshold may not justify the image geometry perfectly and requires a deeper analysis of image geometry before calculating the threshold value [19]. The global threshold can be used to divide an image into two regions: foreground (object) and background. However, if the image has richer contents or more than one object, the global threshold will not work. However, local analysis will be fruitful if the image has richer content. If multiple objects are there within the image boundary along with the background, then multiple thresholds are required at different blocks, which is referred to as local threshold processing for each block [20,21].

Processing the entire image at one level is not a good idea because there may be blocks or parts that do not contribute significantly to the definition of the object or content. Moreover, these regions may be considered ill or noncontributing regions. Segmentation is one method for identifying and removing noncontributing parts of an image to make it more meaningful, and it serves as an input for subsequent processing stages. To achieve better feature extraction, proper image segmentation is critical; otherwise, improper segmentation may result in information loss and have a negative impact on the feature extraction process.

1.1 Threshold Calculation

Image thresholding is an important preprocessing step [22]. There are three ways to calculate the threshold of an image. The first approach follows the threshold calculation at a global or image level. In this approach, a collective decision is taken regarding the threshold value, and the same calculated threshold value is applied to complete the image. It is easier to calculate and can benefit images with homogeneous pixel values or image intensities that are nearly identical, such as texture images. The formula for calculating the global threshold is as follows:

$$g(x, y) = \begin{cases} 1 & \text{if } (x, y) > T \\ 0 & \text{if } (x, y) \leq T \end{cases} \quad (1)$$

The global threshold approach assumes that the histogram must be bimodal. Hence, the extraction of the contributing objects can be performed by creating a threshold boundary between the background pixels and objects, as shown in Fig. 1.

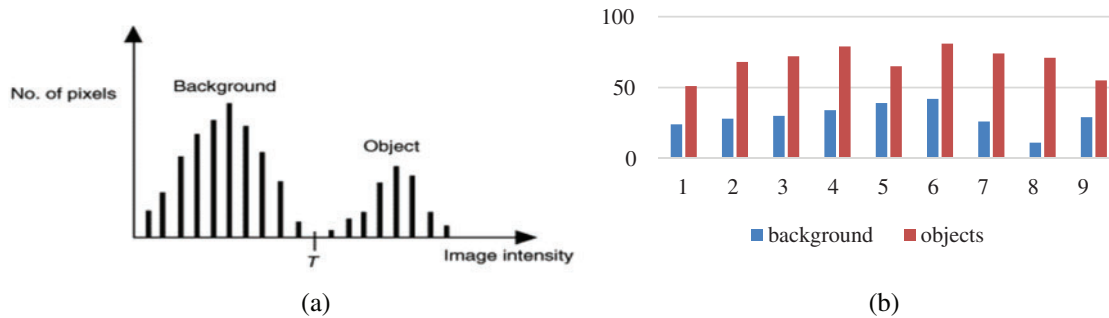


Figure 1: Separation of background pixels from the objects based on (a) Local threshold (b) global threshold

The resultant thresholded image will be a binary image, where 0 and 1 represent the pixel values. The pixel value 1 denotes objects, whereas 0 denotes background regions. However, it suffers from the problem of not considering the local geometry or local intensity of the image; hence, this approach is unable to justify the procedure at the local level; in the case of heterogeneous content, this approach degrades the performance of image segmentation. Local thresholding is another approach that considers local image geometry by considering each pixel value in the threshold calculation process. Usually, the pixel value indicated by 0 represents white, whereas 255 represents black. However, pixel values between 1 and 254 represent different intensity levels. This technique works better than the global thresholding method because it allows images to have different intensity (contrast) levels. Various thresholding approaches have been reported to date for image thresholding. The assumption for local thresholding is that small patches of images are most suitable for a similar intensity illumination, which results in an almost uniform pixel distribution. Dividing images into patches is the first preprocessing step of image segmentation for local thresholding, and then the optimal threshold for each image patch is calculated by investigating its histogram. The threshold calculation for each pixel is performed by interpolating the results of all sub-image patches. The only issue with this approach is its computational cost; therefore, it is not suitable for applications where time is of the essence, such as real-time image processing applications.

1.2 Motivation

The proposed method addresses the issues and limitations of existing segmentation techniques and proposes a solution for existing issues in the following manner.

The global thresholding approach results in injustice with the local geometry of the image, which is resolved by providing a novel adaptive thresholding technique at the regional level.

The gradient and Laplacian of Gaussian algorithms cannot perfectly justify image boundaries. Two-stage verification of the boundary pixels is required for a better analysis of the boundary regions. The proposed method includes two-fold verification of boundary pixels, in which the Laplacian of Gaussian is used in the first fold and the Extrema Pattern is used in four directions in the second fold.

The correlation of the local properties of the image patch with the global properties of the image must be done correctly. The proposed method addresses these issues by establishing a proper relationship between the local intensity values and the global intensity of the entire image.

A nonuniform adaptive approach should be followed for acquiring segments of variable sizes. The proposed method adds a variable threshold that maintains the adaptive behavior for each image patch and behaves according to the local image geometry.

2 Proposed Method

The proposed method segments the image using a nonuniform and adaptive approach. As a preprocessing step, the proposed method first divides the image into non-overlapping sub-blocks. Each block is treated as a separate image or block, and each block is processed separately. Instead of global thresholding, a local thresholding approach was considered for individual blocks. Local image geometry can be handled better by including the adaptive threshold, as well as by considering the local intensity values when processing the adaptive threshold.

2.1 Adaptive Threshold Calculation

The proposed method considers the local thresholding of an image. At the initial stage, the local intensity contribution at each block is calculated by analyzing the relationships among the intensity values within an image patch of 8×8 . Generally, most of the information remain in the central part of the image; however, boundary pixels may also contribute to selecting the edges of objects or segments. Each pixel contributes to the threshold calculation process. Hereafter, boundary pixels can be considered by padding the pixel values at the boundary areas of the image.

Threshold calculation is done as follows:

$$T[x, y, f(x, y), p(x, y)] = f(x, y) * \frac{\max(block)}{\min(block) + \text{average}(block)} \quad (2)$$

Here, x and y are the coordinates in the horizontal and vertical directions, $f(x, y)$ is the location of the pixel, and $p(x, y)$ is the intensity value at the location. As shown in Eq. (2), the proposed method prepares an adaptive mask by placing the calculated threshold value at the corresponding location in the generated mask. The generated adaptive mask is illustrated in Fig. 2.

Threshold values are related to the geometry of the local block. According to the formula given in Eq. (2), if the minimum pixel values and average pixel values are the same, the region is a texture region or a region with homogenous pixel values. According to the object detection rule, abrupt changes in the intensity value and the collection of heterogeneous pixel intensity values

are the primary sources for the possibility of finding objects in those areas. The proposed image segmentation system is shown in Fig. 3, and the working procedure of the proposed method is presented in Tab. 1.

T1	T2	T3	T4	T5	T6	T7	T8
T9	T10	T11	T12	T13	T14	T15	T16
T17	T18	T19	T20	T21	T22	T23	T24
T25	T26	T27	T28	T29	T30	T31	T32
T33	T34	T35	T36	T37	T38	T39	T40
T41	T42	T43	T44	T45	T46	T47	T48
T49	T50	T51	T52	T53	T54	T55	T56
T57	T58	T59	T60	T61	T62	T63	T64

Figure 2: Adaptive mask at block-level

Algorithm 1: Binary Threshold Calculation

Input: Gray Scale image

Output: Segmented Image

Step 1: Check whether image is gray scale or Colored if colored then convert it into in grayscale precision

Step 2: Divide the image into non-overlapping blocks.

Step 3: Process each block separately for checking the similarity among the pixels of the block using following Eq. (3):

$$sim_{per}(region_i) = \sum_{ij} \frac{G(i,j)}{1 + |i - j|} \quad (3)$$

Where $G(I, j)$ id the gray level co-occurrence matrix and $sim_{per}(region_i)$ is the similarity percentage for the region.

Step 4: if value of threshold is greater than 0.85 then image patch is considered as homogeneous otherwise considered as heterogeneous image patch and go to step 5.

Step 5: Divide the image patch into sub-patches of 8×8 .

Step 6: Prepare an adaptive mask of 8×8 i.e., values within the mask must be different for each image sub-patch using Eq. (2)

Step 7: Convolve the mask on the Image block separately and perform the following operation on each pixel of the block

$$Mask_{image} = f(x, y) - mask(x, y) \text{ where } x \in \{0, 2, \dots, 7\} \text{ and } y \in \{0, 2, \dots, 7\}$$

Step 8: Check the lower values of image after convolve i.e., Re-adjust the pixel values after subtraction of the mask from the original image using Eq. (4):

for $i = 0:7$

for $j = 0:7$

If $(Mask_{image}(i, j) < 0)$ then

$$I_{first} = \sum_{i=1}^M \sum_{j=1}^N \max(\min(I_{sub}(i, j), 1), 0)$$

(Continued)

Step 9: Subtract the new obtained image in step 8 with the original image using below formulation:

$$I_{second} = I_{Original} - I_{first}$$

Step 10: Re-adjust the values again in the range from [0–255] using following equation

for $i=0:M$

for $j=0:N$

If $(I_{second}(i,j) < 0)$ then

$$I_{second}(i,j) = 0$$

Step 11: Calculate the gradient and Laplacian for identifying the boundary of the segment

Step 12: Apply the Hybrid Local Extrema Pattern on I_{second} using Eq. (2) in three principle directions i.e., in $\{0^0, 90^0, 135^0\}$.

Step 13: Generate a binary thresholded image.

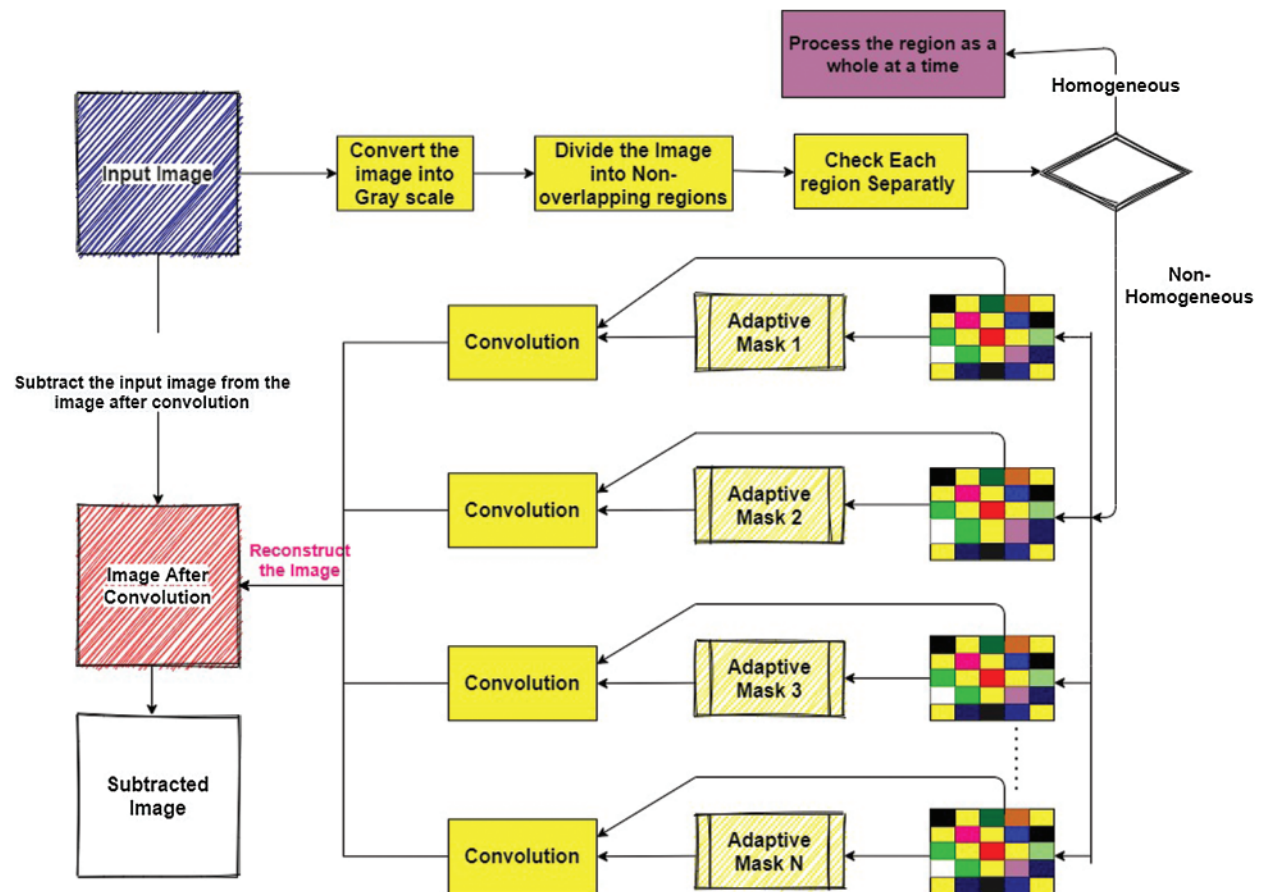


Figure 3: Proposed image segmentation system

2.2 Non-Uniform Segmentation

The proposed approach sub-divides the image into non-overlapping regions. However, the separation of adjacent regions may produce faulty segments if the adjacent regions have similar properties. Uniform segmentation does not deal with the faulty segments, and a segmented image will have more ill objects. To alleviate this issue, the proposed approach follows the non-uniform

segmentation technique. The proposed method for achieving non-uniform segmentation is as follows:

Input: Region of the image

Output: Non-uniform region (after merging)

Step 1: for $i = 1$ to number_of_region

Step 2: if (gradient (i) \leq Threshold)

goto step 3

then

calculate Laplacian of region(i);

else

goto Step 4.

Step 3: Repeat the same process as done in Step 2 for next adjacent regions.

Step 4: Check the Laplacian(i) with the Laplacian(i+1) [i.e., of next adjacent region]

If $((\text{Laplacian}(i) - \text{Laplacian}(i+1)) \parallel \text{Laplacian}(i) - \text{Laplacian}(2i)) \parallel \text{Laplacian}(i) - \text{Laplacian}(2i+1)) < \beta$

then

merge the regions with region(i)

Step 5: Non-uniform regions

However, the first step in both uniform and non-uniform segmentation is the same, that is, both divide the image into non-overlapping regions. In uniform segmentation, by default, all segments are of the same size, and the decision to consider the segment is based on parameters such as homogeneity, intensity distribution, and percentage of object existence within the segment. However, the proposed segmentation technique is based on a nonuniform segmentation approach. The proposed approach considers the gradient descent and Laplacian of Gaussian for calculating the segment boundary, as explained in the algorithm presented in [Tab. 1](#). The proposed method captures the adaptive behavior of the filter and manages the non-uniformity of the segment. A proper segment does not bind to any specific size, and it can vary according to the image properties.

Table 1: Position of block at different angles

S.No.	Postion	Partition	Angle
1	Top	Top_left	0°, 270°
2		Top_Middle	0°, 180°, 270°
3		Top_right	180°, 270°
4	Middle	Middle_left	0°, 90°, 270°
5		Middle_middle	0°, 90°, 180°, 270°
6		Middle_right	90°, 180°, 270°
7	Bottom	Bottom_left	0°, 90°
8		Bottom_Middle	0°, 90°
9		Bottom_right	180°, 270°

Image decomposition results in different chunks of specific sizes as shown in Fig. 4. Each chunk is classified into a specific category based on its placement in the original image, such as at the top, bottom, or middle. Furthermore, these categories were further divided into three categories, as shown in Fig. 4. The association of these chunks into specific categories is done to determine the boundary value of the chunks with their neighboring chunks. However, each chunk will be compared along with the boundary pixels with the neighboring pixels. There are certain angles on which comparison is done as shown in Fig. 6. Meanwhile, if the neighboring chunks meet the desired criteria, only merging can be done. Boundary pixels are shown in different colors in Fig. 6, from which it can be understood that certain boundary pixels will be overlapped (at the corners of the chunk) each time. This overlapping will help decide whether a boundary pixel is belonging to more than one chunk.

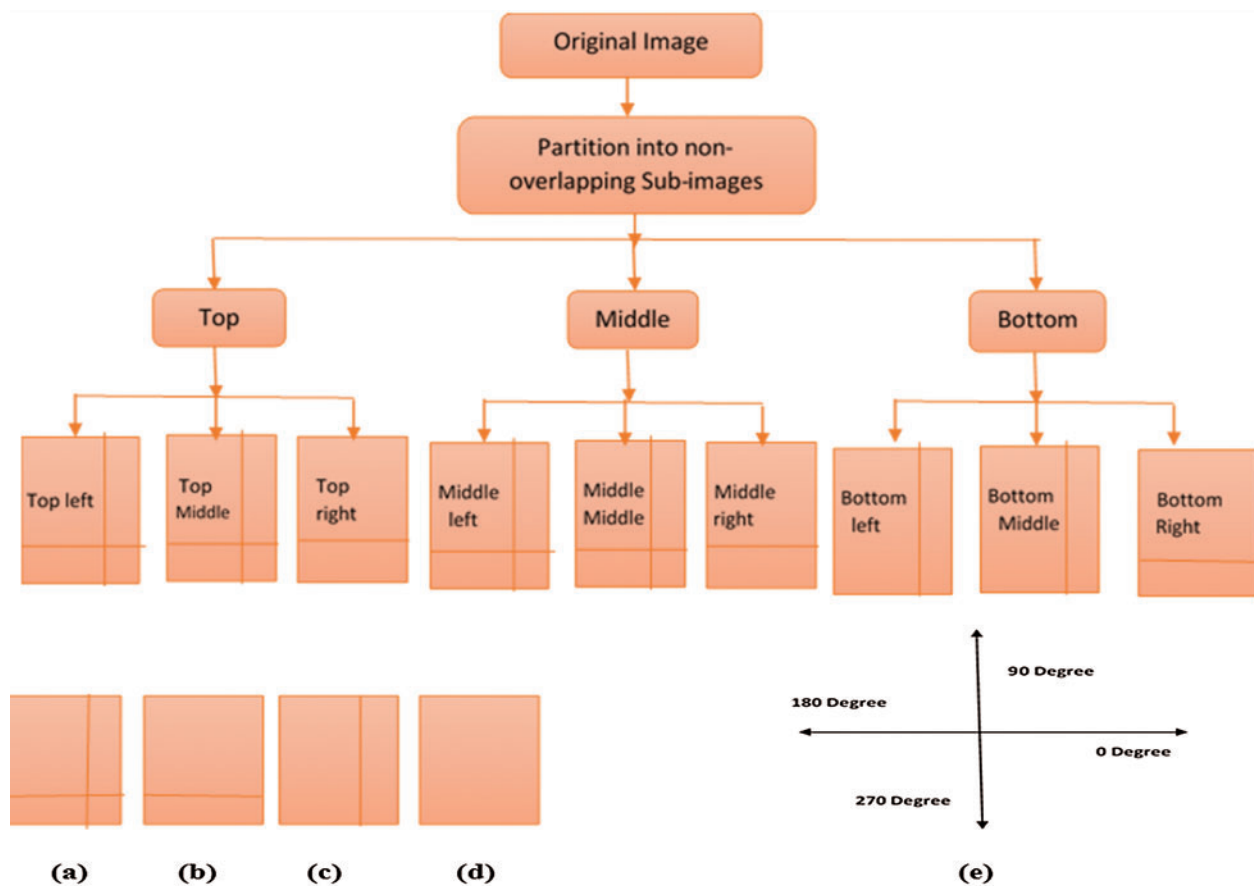


Figure 4: Different categories of partitioned areas of image. (a) Top left chunk; (b) Top middle chunk; (c) Top right chunk; (d) Middle left chunk, and (e) Angles at different degrees

2.3 Effect of Block Size and Mask Size

The proposed approach considers chunks of different sizes. However, the size of each block did not reflect the size of the mask. Blocks of smaller sizes result in the presence of ill objects in the segmented image, as shown in Fig. 5. Another effect of small chunks is that more merging

steps are required in every phase. The proposed method iterates through the different images on the chunks of different sizes, and after executing multiple iterations, it is concluded that an 8×8 block size is more suitable for the segmentation process. However, Fig. 5 shows that the number of ill objects is smaller in the blocks of 16×16 and 32×32 sizes, but the performance of adaptive mask decreases as the size of each block is increased from 8×8 . The performance of the adaptive mask decreases with an increase in the block size because it takes the local maximum intensity value in the threshold generation process. If the size of the mask is larger, it creates an adverse effect on the adaptive mask calculation because the radius of the local geometry will be increased; the local maximum intensity value may not be able to justify with the complete image patch.

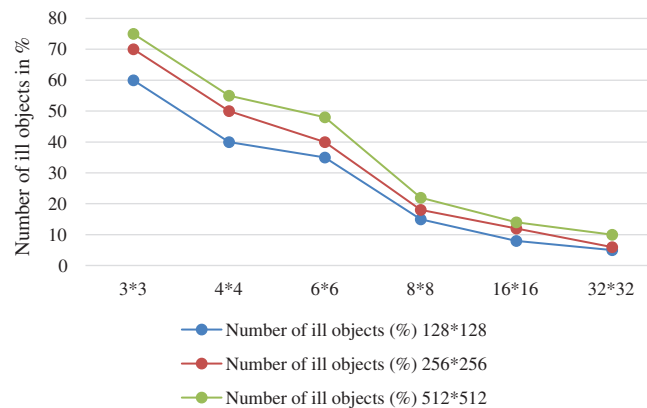


Figure 5: Relation of block size and ill (faulty) objects

55	45	66	76	23	11	25	15
90	88	89	78	66	67	66	55
9	77	71	23	56	91	11	23
44	67	88	99	34	43	23	36
54	56	67	44	23	34	78	90
34	23	45	29	107	98	12	65
56	78	99	77	89	123	11	55
34	45	56	67	78	89	90	88

55	45	66	76	23	11	25	15
90	88	89	78	66	67	66	55
9	77	71	23	56	91	11	23
44	67	88	99	34	43	23	36
54	56	67	44	23	34	78	90
34	23	45	29	107	98	12	65
56	78	99	77	89	123	11	55
34	45	56	67	78	89	90	88

Figure 6: Boundary pixels (in colors) for comparisons with neighboring regions

Block-level processing of an image is an important aspect of the proposed image segmentation system. Block-level processing reduces the number of pixels in the pattern generation phase and simultaneously increases the discriminative power. Small blocks lead to more information loss, whereas larger blocks result in less discriminative power. Fig. 5 shows that blocks of smaller sizes result in more information loss.

2.4 Extrema Pattern Calculation

The extrema pattern determines the boundary value of any pixel in different directions. According to the placement of the image chunk or patch, the weight difference extrema pattern can be calculated as follows: The WD-EP was calculated using $D1 = IP_a - IP_c$ and $D2 = IP_c - IP_b$.

$$EP(\alpha) = \begin{cases} 0 & \text{if } ((IP_c > IP_a \ \&\& \ IP_c < IP_b) \text{ or } (IP_c \setminus IP_a \ \&\& \ IP_c \setminus IP_b)) \\ 1 & \text{else if } D1 \geq \phi \ D2 \geq \phi \\ 0 & \text{else} \end{cases} \quad (4)$$

$$EP = \sum_{p=1}^{BL} \{K_\alpha(g_c); K_\alpha(g_1); K_\alpha(g_2); \dots K_\alpha(g_8)\} \quad (5)$$

where $\alpha = \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$, BL is the number of image chunks or patches, and $K(g_i)$ is the resultant difference (either 0 or 1) between the neighboring pixels and the current central pixel in the specified α direction. ϕ is the user-specified threshold. IP_a , IP_b , and IP_c are the pixel values to be compared. Eq. (5) shows that if there is no edge in the generated pattern, the bit having value “0” will not change in EP. However, if there is an edge (i.e., the DLEP pattern has a bit value “1”), the bit value will be checked against the threshold and set to “0 s” if it does not satisfy the threshold condition.

The proposed method deals well with segments of irregular sizes by thoroughly investigating the boundary pixels with their neighboring pixels in the four principal directions (90° , 180° , 0° , and 270°). Eventually, the generated binary pattern determines whether the boundary pixels have relations with their adjacent neighboring pixels of another block. This will be decided in the following way:

$$EP_\alpha = \begin{cases} \text{True} & \text{if } (count_of_1bit\{K_\alpha(g_c); K_\alpha(g_1); K_\alpha(g_2); \dots K_\alpha(g_8)\}) > 6 \\ \text{False} & \text{otherwise} \end{cases} \quad (6)$$

Here, EP_α is the generated extrema pattern in the specific direction α (i.e., 0° , 90° , 180° , 270°). $count_of_1bit$ is a function that counts the occurrence of “1-bit” values in the generated pattern. The size of the pattern is nine bits long, and the above equation shows that if the total number of 1-bit values is more than six, then only will it be considered the boundary pixel, which is related to the neighboring block. There is a significant value of not only the count of 1-bit value but also the position of the pixel values in the generated pattern. A similar process of binary pattern generation is initiated from the neighboring block. If the $count_of_1 \ bit$ function returns “True,” both neighboring blocks will be merged; otherwise, they will be processed separately with their respective segments. The process of extreme pattern calculation minimizes the effect of ill or false segments from the generated results, as shown in Fig. 7.

Each boundary pixel was compared with the adjacent pixels in the 90° direction. It is shown in Fig. 8 that the generated pattern has 7 “1-bit” values. The proposed method sets the threshold of six for merging blocks; hence, adjacent blocks are merged in this case.

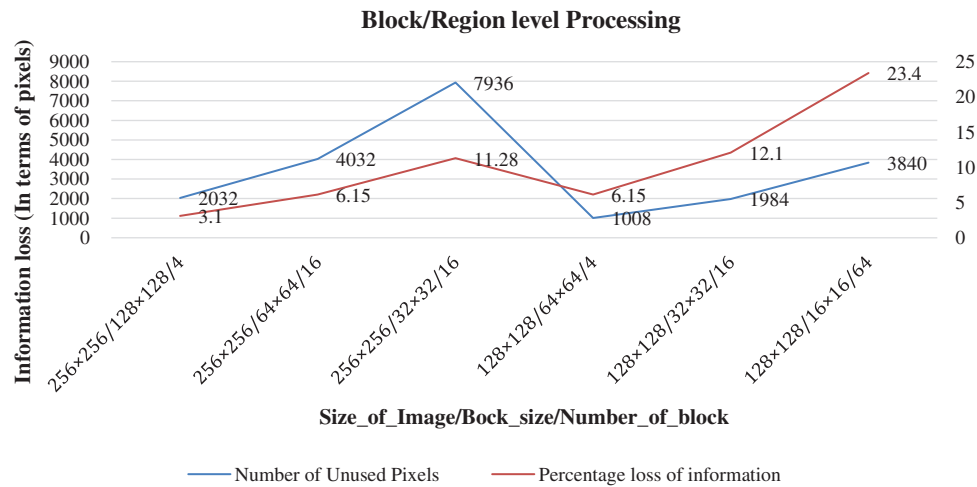


Figure 7: Effect of block size on information loss (unused pixels)

13	4	6	3	12
	7(8)	9(7)	11(6)	7
4	2(5)	8(0)	19(1)	10
8	6(4)	4(3)	14(2)	21
3	5	16	23	11

13	4	6	3	12
5	7(8)	9(7)	11(6)	7
4	2(5)	8(0)	19(1)	10
8	6(4)	4(3)	14(2)	21
3	5	16	23	11

13	4	6	3	12
5	7(8)	9(7)	11(6)	7
4	2(5)	8(0)	19(1)	10
8	6(4)	4(3)	14(2)	21
3	5	16	23	11

13	4	6	3	12
5	7(8)	9(7)	11(6)	7
4	2(5)	8(0)	19(1)	10
8	6(4)	4(3)	14(2)	21
3	5	16	23	11

13	4	6	3	12
5	7(8)	9(7)	11(6)	7
4	2(5)	8(0)	19(1)	10
8	6(4)	4(3)	14(2)	21
3	5	16	23	11

13	4	6	3	12
5	7(8)	9(7)	11(6)	7
4	2(5)	8(0)	19(1)	10
8	6(4)	4(3)	14(2)	21
3	5	16	23	11

13	4	6	3	12
5	7(8)	9(7)	11(6)	7
4	2(5)	8(0)	19(1)	10
8	6(4)	4(3)	14(2)	21
3	5	16	23	11

13	4	6	3	12
5	7(8)	9(7)	11(6)	7
4	2(5)	8(0)	19(1)	10
8	6(4)	4(3)	14(2)	21
3	5	16	23	11

13	4	6	3	12
5	7(8)	9(7)	11(6)	7
4	2(5)	8(0)	19(1)	10
8	6(4)	4(3)	14(2)	21
3	5	16	23	11

Figure 8: Extrema pattern (EP) for $\alpha = 90^0$ and pattern = 011111011

2.5 Experimental Result and Discussion

The proposed method was tested on images from the COREL 1k dataset. The image segmentation process was performed on images with heterogeneous content because the resultant segmented image contains regions of similar or homogeneous content. There were 10 categories in the COREL dataset. Representative images of the COREL dataset are presented in Fig. 9.



Figure 9: Example images from COREL database for image segmentation. (a) Image with boat; (b) Image with humans; (c) Image with human in a sea and boats, and (d) Humans running in a water

The efficiency of the image segmentation technique was determined by counting the number of ill or noncontributing regions. Faulty segments were extracted after segmentation if the segmented image contained more ill regions, resulting in faulty segments. Fig. 10 shows the resultant segmented reasons based on certain threshold values. The proposed method sets the area of the segment as a threshold; therefore, segmented regions with areas greater than 250 were considered and those with less than 250 were discarded.

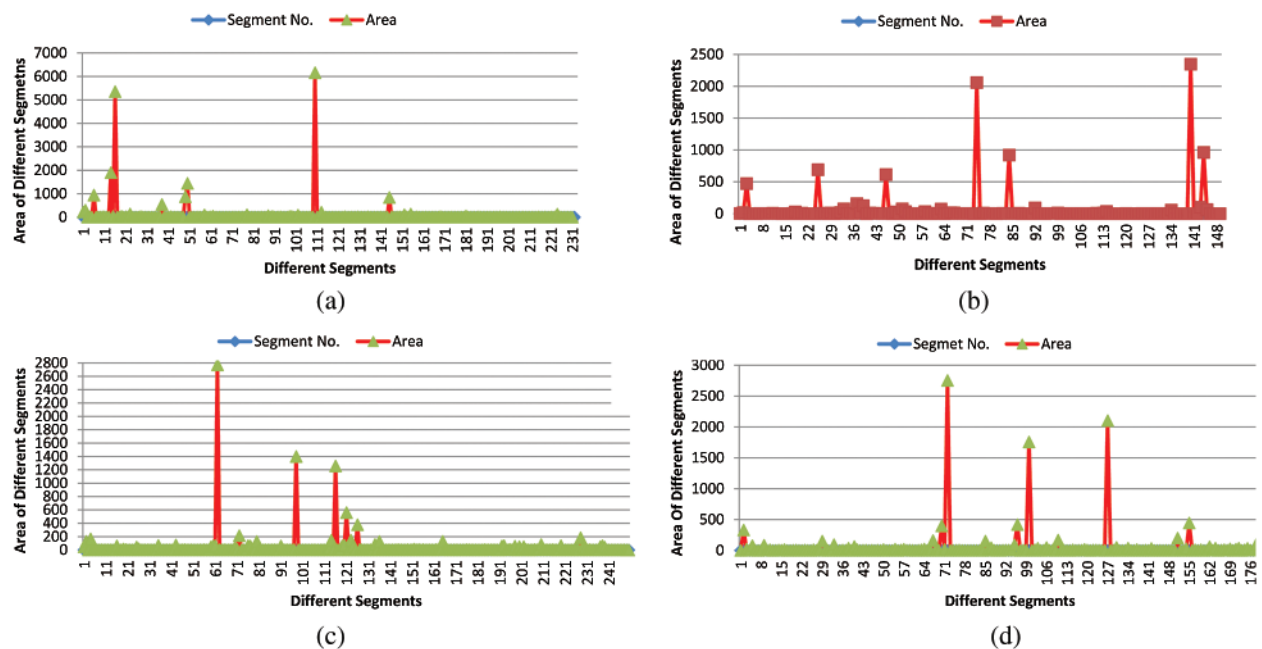


Figure 10: (a–d): Extracted segments, indicated in red are selected against the area threshold

The proposed method achieves better image segmentation by considering the local image geometry analysis of the images, as shown in Fig. 11. The segmented images at various stages are shown in Fig. 11. Fig. 11a shows the background of the image based on the adaptive threshold.

Fig. 11b shows the image after subtracting it from the background, and Fig. 11c shows the final segmented image after applying the mask of 8×8 . From Fig. 12, it is observed that meaningful objects can be extracted by minimizing the effect of the ill segments or regions in the segmented image. However, the nonuniform and adaptive threshold selection approach also significantly and collectively works together to achieve a better segmented image. The area threshold works well to achieve the segmented image, and segmented regions are extracted separately based on the calculated adaptive threshold, as shown in Fig. 12.

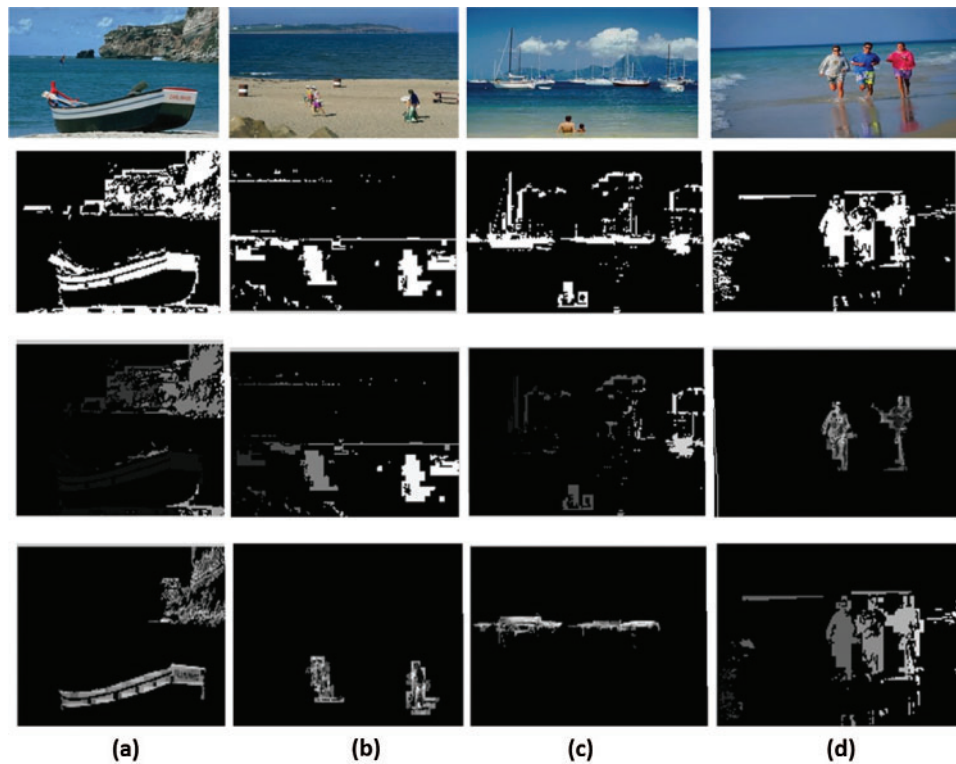
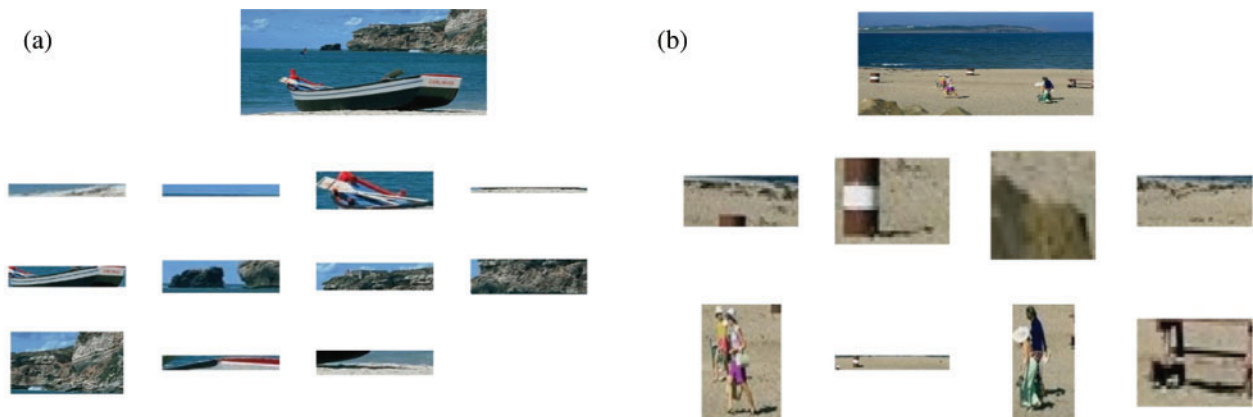


Figure 11: Image segmentation at different phases. (a) Adaptive thresholding; (b) Image subtraction; (c) Final image; (d) Refinement



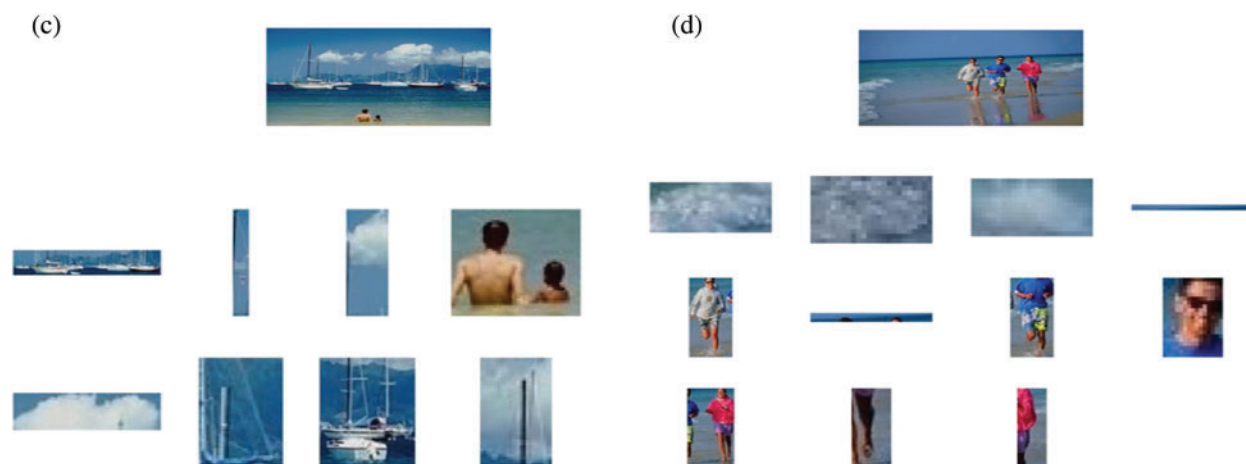


Figure 12: Extracted segments from the original image. (a–d) Meaningful segmented objects with the reference of Fig. 9

3 Conclusion

The proposed method presents a system for digital image segmentation using Novel Extrema Pattern and Laplacian of Gaussian. Image analysis is done at block or region level by dividing the image into non-overlapping sub images or regions. The proper boundary analysis among the adjacent blocks is done by the proposed extrema pattern. Adaptive threshold is the another key feature of the proposed method which works as an input for the variable mask. Adaptive Threshold is calculated by the local properties of the local image patch and global properties of the image. Generate variable mask convolves over the image block and results in better image segmentation. The proposed segmentation system performs well with the images of dense diversity due to region level analysis. The Proposed method tested the accuracy of the segmentation on the heterogeneous images which are taken from the COREL dataset. However, proposed system works well with the images of homogeneous contents too. The experimental results show that the proposed segmentation method out-performs on the block size of 8×8 with better segmentation accuracy. The proposed method can be further extended in future for image retrieval purposes and critical analysis of the medical images.

Funding Statement: This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ICAN (ICT Challenge and Advanced Network of HRD) program (IITP-2021-2020-0-01832) supervised by the IITP (Institute of Information & Communications Technology Planning & Evaluation) and the Soonchunhyang University Research Fund.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] M. Nasir, I. U. Lali, T. Saba and T. Iqbal, "An improved strategy for skin lesion detection and classification using uniform segmentation and feature selection based approach," *Microscopy Research and Technique*, vol. 81, pp. 528–543, 2018.

- [2] T. Saba, A. Rehman and S. L. Marie-Sainte, "Region extraction and classification of skin cancer: A heterogeneous framework of deep CNN features fusion and reduction," *Journal of Medical Systems*, vol. 43, pp. 1–19, 2019.
- [3] S. Tongbram, B. A. Shimray, L. S. Singh and N. Dhanachandra, "A novel image segmentation approach using fcm and whale optimization algorithm," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 1–15, 2021.
- [4] M. Sharif, U. Tanvir, E. U. Munir and M. Yasmin, "Brain tumor segmentation and classification by improved binomial thresholding and multi-features selection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 7, pp. 1–20, 2018.
- [5] T. Akram, M. Sharif, A. Shahzad, K. Aurangzeb, M. Alhussein *et al.*, "An implementation of normal distribution based segmentation and entropy controlled features selection for skin lesion detection and classification," *BMC Cancer*, vol. 18, pp. 1–20, 2018.
- [6] S. Zhou, D. Nie, E. Adeli, J. Yin and D. Shen, "High-resolution encoder–decoder networks for low-contrast medical image segmentation," *IEEE Transactions on Image Processing*, vol. 29, pp. 461–475, 2019.
- [7] H. Li, X. Zhao, A. Su, H. Zhang and G. Gu, "Color space transformation and multi-class weighted loss for adhesive white blood cell segmentation," *IEEE Access*, vol. 8, pp. 24808–24818, 2020.
- [8] Y. Wang, Q. Qi, L. Jiang and Y. Liu, "Hybrid remote sensing image segmentation considering intrasegment homogeneity and intersegment heterogeneity," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, pp. 22–26, 2019.
- [9] L. Zhang, J. Ma, X. Lv and D. Chen, "Hierarchical weakly supervised learning for residential area semantic segmentation in remote sensing images," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, pp. 117–121, 2019.
- [10] Y. Li, G. Cao, T. Wang, Q. Cui and B. Wang, "A novel local region-based active contour model for image segmentation using Bayes theorem," *Information Sciences*, vol. 506, pp. 443–456, 2020.
- [11] M. B. Tahir, K. Javed, S. Kadry, Y.-D. Zhang, T. Akram *et al.*, "Recognition of apple leaf diseases using deep learning and variances-controlled features reduction," *Microprocessors and Microsystems*, vol. 8, pp. 104027, 2021.
- [12] M. S. Sarfraz, M. Alhaisoni, A. A. Albeshier, S. Wang and I. Ashraf, "Stomachnet: Optimal deep learning features fusion for stomach abnormalities classification," *IEEE Access*, vol. 8, pp. 197969–197981, 2020.
- [13] F. Jia, J. Liu and X.-C. Tai, "A regularized convolutional neural network for semantic image segmentation," *Analysis and Applications*, vol. 19, pp. 147–165, 2021.
- [14] P. Sathya, R. Kalyani and V. Sakthivel, "Color image segmentation using kapur, otsu and minimum cross entropy functions based on exchange market algorithm," *Expert Systems with Applications*, vol. 172, pp. 114636, 2021.
- [15] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 641–647, 1994.
- [16] J. Fan, G. Zeng, M. Body and M.-S. Hacid, "Seeded region growing: An extensive and comparative study," *Pattern Recognition Letters*, vol. 26, pp. 1139–1156, 2005.
- [17] A. Rehman, M. A. Khan, T. Saba, Z. Mehmood and N. Ayesha, "Microscopic brain tumor detection and classification using 3D CNN and feature selection architecture," *Microscopy Research and Technique*, vol. 84, pp. 133–149, 2021.
- [18] U. N. Hussain, I. U. Lali, K. Javed, I. Ashraf, J. Tariq *et al.*, "A unified design of ACO and skewness based brain tumor segmentation and classification from MRI scans," *Journal of Control Engineering and Applied Informatics*, vol. 22, pp. 43–55, 2020.
- [19] H. Andrea, I. Aranguren, D. Oliva, M. Abd Elaziz and E. Cuevas, "Efficient image segmentation through 2D histograms and an improved owl search algorithm," *International Journal of Machine Learning and Cybernetics*, vol. 12, pp. 131–150, 2021.

- [20] F. Saeed, M. A. Khan, M. Sharif, M. Mittal and S. Roy, "Deep neural network features fusion and selection based on PLS regression with an application for crops diseases classification," *Applied Soft Computing*, vol. 11, pp. 107164, 2021.
- [21] T. Akram, M. Sharif, M. Awais, K. Javed, H. Ali *et al.*, "CCDF: Automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep CNN features," *Computers and Electronics in Agriculture*, vol. 155, pp. 220–236, 2018.
- [22] F. Di Martino and S. Sessa, "PSO image thresholding on images compressed via fuzzy transforms," *Information Sciences*, vol. 506, pp. 308–324, 2020.