

A Novel Local Adaptive Percentage Split Distribution Method for Image Binarization and Classification

Joy Christy A.

SASTRA Deemed to be University, Thanjavur, India
E-mail: joychristy001@gmail.com
ORCID iD: <https://orcid.org/0000-0002-3250-3492>

Umamakeswari A.*

SASTRA Deemed to be University, Thanjavur, India
E-mail: umamakeswari.arumugam@gmail.com
ORCID iD: <https://orcid.org/0000-0001-9724-1115>
*Corresponding Author

Shanthi P.

SASTRA Deemed to be University, Thanjavur, India
E-mail: shanthip@cse.sastra.ac.in
ORCID iD: <https://orcid.org/0000-0001-7995-9991>

Srilakshmi A.

SASTRA Deemed to be University, Thanjavur, India
E-mail: srilakshmi@cse.sastra.ac.in
ORCID iD: <https://orcid.org/0000-0001-7639-6863>

Siva Chandrasekaran

Swimburne University of Technology, Melbourne, Australia
E-mail: schandrasekaran@swin.edu.au
ORCID iD: <https://orcid.org/0000-0003-2871-880X>

Received: 15 June, 2023; Revised: 05 September, 2023; Accepted: 31 October, 2024; Published: 08 April, 2025

Abstract: Binary thresholding methods separate image pixels into two groups as 0s or 1s. The two types of binary thresholding methods are global thresholding and local thresholding. Global thresholding methods are appropriate for binarizing the images that has smooth and contrast distribution of pixels. The performance of global thresholding struggles with distorted and tampered images as it introduces additional noise and causes variation in contrast and illumination. Local adaptive thresholding methods address the issue with every pixel a threshold based on the contrast distribution of neighboring pixels. This paper introduces Local Adaptive Percentage Split Distribution (LAPSD) method for binarization. LAPSD computes threshold based on percentage wise split of neighboring pixels. The performance of LAPSD is compared with benchmark binary thresholding methods such as Bradley's, Niblack's, and Sauvola's against PSNR, SSIM and MSE metrics. The accuracy of LAPSD image binarization is measured using Convolution Neural Network (CNN) models and the results prove that the performance of the proposed method surpasses traditional methods in all means.

Index Terms: Binary thresholding, global thresholding, local adaptive thresholding, PSNR, SSIM, MSE

1. Introduction

Images contain text, shapes, drawing and graphic regions [1]. Most of the image processing applications use image segmentation techniques to extract meaningful features from images [2,3]. In such instances, it is preferable to segment

images into binary regions. The goal of image binarization is to transform the given coloured or grayscale image into a bi-level image representation in order to segregate the features present in the foreground from background [4,5].

Image binarization methods are widely adopted in deep learning models for image classification and segmentation problems [6-8]. Since, images can be represented as a matrix of pixel values, the researchers use binarization techniques in complex networks [9]. Thresholding is a direct and simple approach for image binarization [10-12]. Thresholding based image binarization methods convert an image into two relevant groups based on boundary threshold(s). The methods assign each pixel of an image into a group that is defined with constant boundary (ies) in such a way that pixels in one group are similar to each other than the pixels in other groups. Image binarization is obtained using global thresholding and local adaptive thresholding methods [13,14]. Global thresholding methods use a single threshold value 'T' for binarizing the whole image [15]. In contrast, the local thresholding methods separate the image into multiple windows and each pixel in the image gets its own threshold based on the neighborhood pixels in the window [16].

There are plenty of global and local thresholding-based image binarization techniques available in the literature, none of them are gold standard [17], as they impact with factors like luminance, contrast and noise [18,19]. The other factors that affect the performance of traditional binarization methods include: accuracy, time and computational complexity [20-23]. Having considering the above problem statements, in this work, we present a local adaptive thresholding method called Local Adaptive Percentage Split Distribution (LAPSD). The idea behind LAPSD thresholding relies on calculating a threshold for each pixel based on the neighboring pixels on fixed window with size 'w' using percentage split distribution method.

2. Literature Review

The existing local adaptive thresholding methods available in the literature are plenty and each of them significantly contributes for the improved performance of image binarization. Almost all the existing local adaptive thresholding methods use integral sum of image for computing the local threshold of a window with size 'w'. A pixel in an integral image 'intSum' is the cumulative sum of all pixels embraced within the left and top of the corresponding pixel from input image 'I' [24-26]. The integral sum of a pixel(x,y) of image 'intSum' can be calculated using the notion given in Equ.1.

$$\text{intSum}_{(x,y)} = \sum_{i=1}^x \sum_{j=1}^y I_{(i,j)} \quad (1)$$

Equ.1. may further be simplified in a single pass using the Equations 2 and 3.

The integral sum of 1st row and 1st column of integral image 'intSum' is calculated using the equations

$$\text{intSum}_{(1,y)} = I_{(1,y)} + \text{intSum}_{(1,y-1)}, \quad y = 2 \dots n \quad (2)$$

$$\text{intSum}_{(x,1)} = I_{(x,1)} + \text{intSum}_{(x-1,1)}, \quad x = 2 \dots m \quad (3)$$

The other entries (x,y) of image the integral image is computed as given in Equ.4.

$$\text{intSum}_{(x,y)} = I_{(x,y)} + \text{intSum}_{(x,y-1)} + \text{intSum}_{(x-1,y)} - \text{intSum}_{(x-1,y-1)}, \quad x = 2 \dots m; y = 2 \dots n \quad (4)$$

The local sum of a pixel S(x,y) of a window 'w' is calculated in single pass with linear complexity using two additions and one subtraction operations as depicted in Equ.5.

$$S_{(x,y)} = [\text{intSum}_{(x+d-1,y+d-1)} + \text{intSum}_{(x-d,y-d)}] - [\text{intSum}_{(x-d,y+d-1)} + \text{intSum}_{(x+d-1,y-d)}] \quad (5)$$

Where, $d=w/2$

The local arithmetic mean $M(x,y)$ is the average of pixels within the window 'w×w' and is computed using the Equ. 6.

$$M_{(x,y)} = S_{(x,y)} / (w \times w) \quad (6)$$

The local adaptive thresholding method proposed by Niblack [27-31] computes the threshold of a pixel T(x,y) as denoted in Equ.7:

$$T_{(x,y)} = M_{(x,y)} + k\delta_{(x,y)} \quad (7)$$

Where,

$M_{(x,y)}$ is the local arithmetic mean of a local window 'w'

k is the bias

$\delta_{(x,y)}$ is the standard deviation of pixels inside the local window 'w'

Though, Niblack's method is a widely used image binarization method, it works effectively only on low contrast images. The method could introduce additional noise, with small window sizes. Sauvola et al. [32-35] have proposed yet other integral image based local adaptive thresholding method for calculating the threshold of pixel T(x,y) as denoted in Equ.8.

$$T_{(x,y)} = M_{(x,y)} [1 + k(\delta_{(x,y)}/R - 1)] \quad (8)$$

The method overcomes the additional noise problem found in Niblack's method, yet struggles, if contrast between foreground and background is low.

The local adaptive thresholding method proposed by Bradley's [36-38] uses different notion for calculating the local sum S(x,y) of a window 'w' as denoted in Equ.9.

$$S_{(x,y)} = \text{intSum}_{(x2,y2)} - \text{intSum}_{(x2,y1-1)} - \text{intSum}_{(x1-1,y2)} + \text{intSum}_{(x1-1,y1-1)} \quad (9)$$

The values of x1, x2, y1 and y2 will be calculated using the Equations 10 and 11.

$$x1 = x - w/2; \quad x2 = x + w/2 \quad (10)$$

$$y1 = y - w/2; \quad y2 = y + w/2 \quad (11)$$

Finally the output image is constructed using the equation shown in Equ.12.

$$O_{(x,y)} = \begin{cases} 1 & \text{if } (I_{(x,y)} * \text{count}) \leq S_{(x,y)} \times (100 - tp)/100 \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

where,

$$\text{count} = (x2 - x1) \times (y2 - y1)$$

tp is the threshold percentage

The method is easy to implement, but the sparsity in foreground object results in significant background noise in the output image.

Bernsen's method [34] uses Equ.13 to calculate threshold value T(x,y) at (x,y) of a window size w×w as follows:

$$T_{(x,y)} = 0.5(I_{\max(i,j)} + I_{\min(i,j)}) \quad (13)$$

where,

$I_{\max(i,j)}$ and $I_{\min(i,j)}$ are maximum and minimum values of within the local window

The raise in contrast and more neighborhood values increases the process time of the method.

Since, the binarization methods presented in the literature review have all got some limitations of its kind, there is a need to propose an extensive binarization method, that overcomes these limitations.

3. Research Context and Methodology

In this paper, we propose an integral sum based local adaptive thresholding method 'LAPSD' for image binarization. The method computes a threshold for each pixel in the input image (x,y) by considering a window of neighbouring pixels with size 'w×w'. The method calculates 'intSum' – the integral sum of image using Equ.4. The minimum and maximum values of each 'intSum' window is then computed and used in the calculation of threshold for each pixel. Having the pixel (x,y) at the centre of the 'intSum' window 'w×w', the minimum and maximum values of neighbouring pixels are computed using Equations 14 and 15.

$$\text{minv}_{(x,y)} = \min(A, B) \quad (14)$$

$$\text{maxv}_{(x,y)} = \max(A, B) \quad (15)$$

The formulas for calculating A and B are shown in Equations 16 and 17.

$$A = [\text{intSum}_{(x+d-1,y+d-1)} + \text{intSum}_{(x-d,y-d)}] \quad (16)$$

$$B = [\text{intSum}_{(x-d,y+d-1)} + \text{intSum}_{(x+d-1,y-d)}] \quad (17)$$

where $d = \text{round}(w/2)$

In general, the window size 'WindowSize' is 'w×w', but varies for the pixels located in the brinks of the image. The minimum and maximum values of 'intSum' window are applied in Equ.18 to derive the threshold of pixel(x,y) of input image.

$$T_{(x,y)} = (\max_{(x,y)} - \min_{(x,y)}) \times tp / \text{WindowSize} \quad (18)$$

Where 'tp' refers to threshold percentage, by default it is set to 0.5.

Once, the threshold for pixel_(x,y) is computed, an if-then association rules is then executed to binarize the input image 'I' as depicted in Equ.19.

$$O_{(x,y)} = \begin{cases} 1 & \text{if } (I_{(x,y)} < T_{(x,y)}) \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

The performance of the LAPSD is kept increasing with different values of threshold percentage 'tp' with varying window size 'w'. To reduce the errors that occurs in the brinks of binarized image, 'intSum' is padded 'd' times with last row and column. The step-by-step process of the proposed LAPSD method is shown in Algorithm 1.

Algorithm 1: LAPSD

```

Procedure. LAPSD (I, O, width, height)
1. for i=1 to width do
2.   sum=0
3.   for j=1 to height do
4.     sum=sum+I[i,j]
5.     if i=0 then
6.       intSum[i,j]=sum
7.     else
8.       intSum[i,j]=intSum[i-1,j]+sum
9.     end if
10.  end for
11. end for
12. d=round(window size/2)
13. for i=1 to d do
14.   cbind(intSum,intSum[height])
15. for j=1 to d do
16.   rbind(intSum,intSum[width])
17. for i=1 to width do
18.   for j=1 to height do
19.     A=intSum[i+d-1, j+d-1]+intSum[i-d,j-d]; B=intSum[i-d, j+d-1]+intSum[i+d-1,j-d]
20.     minv=min (A, B); maxv=max (A, B); t[i,j]=((maxv-
inv)×threshold)/(WindowSize)
21.     if (I[i,j]<t[i,j]) then
22.       O[i,j]=0
23.     else
24.       O[i,j]=1
25.     end if
26.   end for
27. end for

```

If the look up index of 'intSum' is not found within the range of 'intSum', the corresponding index's 'intSum' value is considered to be '0'. The overview of the proposed methodology is demonstrated using a 3×3 input image 'I' with window size 'w' as 3, 'd' as 2 and threshold percentage 'tp' as 0.5. Fig.1. denotes the derivation of integral image 'intSum'. Fig.2. previews the neighboring pixels of each element in input image 'I'.

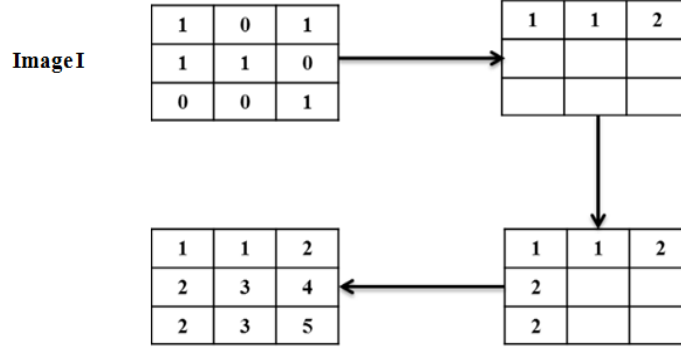


Fig. 1. Integral image 'intSum'

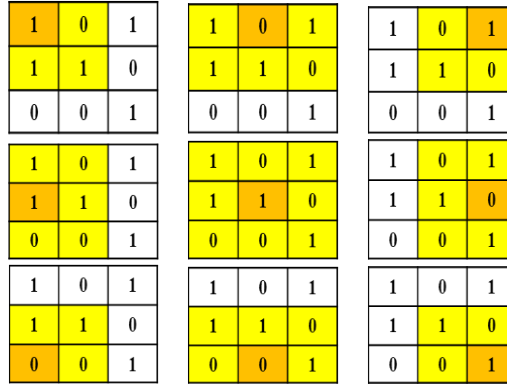


Fig. 2. Neighboring pixels of each element in Input Image 'I' (w×w size)

The threshold for the element (2,2) of the input image 'I' can be calculated as follows:

$$\begin{aligned}
 A &= \text{intSum}_{(2+2-1,2+2-1)} + \text{intSum}_{(2-2,2-2)} \\
 B &= [\text{intSum}_{(2-2,2+2-1)} + \text{intSum}_{(2+2-1,2-2)}] \\
 A &= [\text{intSum}_{(3,3)} + \text{intSum}_{(0,0)}] \\
 B &= [\text{intSum}_{(0,3)} + \text{intSum}_{(3,0)}] \\
 A &= [5 + 0] = 5 \\
 B &= [0 + 0] = 0 \\
 \text{maxv}_{(x,y)} &= \max(A, B) = \max(5, 0) = 5 \\
 \text{minv}_{(x,y)} &= \min(A, B) = \min(5, 0) = 0
 \end{aligned}$$

Having computed the values $\text{maxv}_{(2,2)}$ and $\text{minv}_{(2,2)}$, The threshold $T_{(2,2)}$ can be computed as

$$T_{(2,2)} = \left(\left(\text{max}_{(2,2)} - \text{minv}_{(2,2)} \right) \times tp \right) / \text{WindowSize} = ((5 - 0) \times 0.5) / 9 = 2.5 / 9 = 0.278$$

The step-by-step calculation of threshold $T_{(2,2)}$ is presented in Fig.3.

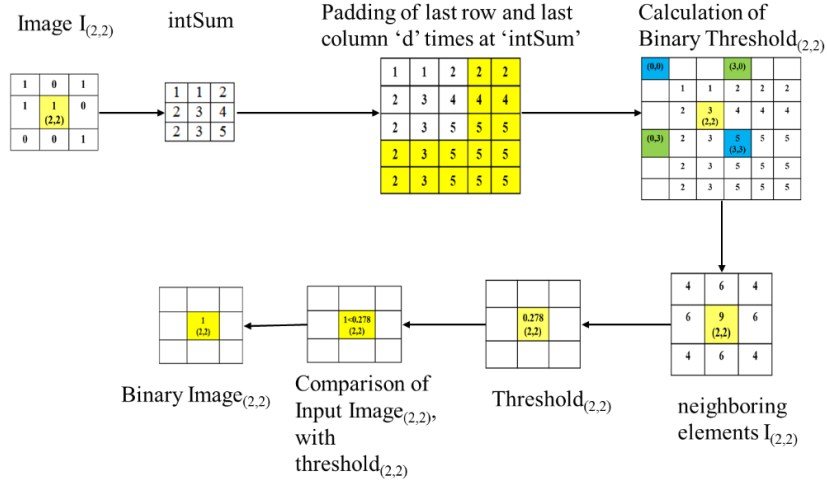


Fig. 3. Demonstration of LAPSD method for an element (2,2) of input image 'I'

Thresholds for each element in the input image 'I' and the corresponding output is shown in Fig.4.

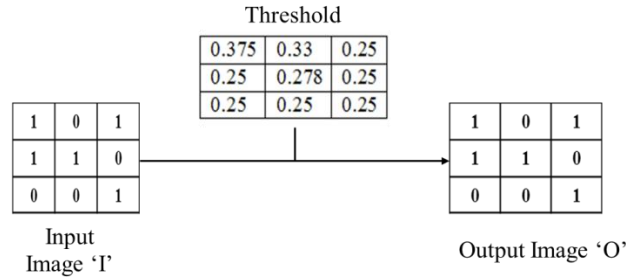


Fig. 4. LAPSD Thresholds for each element in 'I'

4. Experimentation

The proposed LAPSD method, is an image binarization technique, and can be used for recognizing objects, shapes and characters. The method could clearly separate object of interest from background. Thus, the method can be adopted as an image-preprocessing technique for segmenting tasks. The efficiency of the proposed method is tested to classify the clipart images. This section describes the experimentation of proposed methodology over some bench mark clipart images taken from https://favpng.com/png_search/tree [39]. The dataset contains 5,000 colored images, out of which several random images of 'COCONUT' and 'FIG' trees have been chosen for the experimentation. The experimental images are varying in size with odd and even dimensions. Fig.5. shows the experimental images. The images have specific challenging characteristics such as water marking, low-high contrast variations, additive random noises, spatial variations in lightings, smoothed borders and motifs of structured noise. The objective of the experimentation is to analyze the performance of the proposed LAPSD method over these challenging images. Experimentation of the proposed work is categorized into two sections, i). conventional experimentation and convolutional experimentation. Under conventional experimentation the performance of the LAPSD method is verified against Peak Signal to Noise Ratio (PSNR), Structural Similarity Index (SSIM) and Mean-Square Error (MSE) metrics. On the other hand, in convolutional experimentation the performance of the proposed LAPSD method is verified against accuracy and loss metrics with respect to image classification.

4.1. Conventional Experimentation

The conventional experimentation is conducted using R Studio version 3.6.1 on a workstation with Intel Core i5 processor with Windows 10 Operating system (64 bits) installed in it. In conventional experimentation we pass the input images into four image binarization methods such as proposed LAPSD, Bradley, Niblack and Sauvola and measure the performance of the methods against PSNR, SSIM and MSE metrics for different window sizes. A details description on the conventional metrics is discussed in this section.

The Mean-Square Error (MSE) is the summation of squared error between the original and segmented image [40]. The lower the value of MSE, the lower the error is. Equ.20. is used to compute the MSE

$$MSE = \sum_{(x,y)} [O(x,y) - S(x,y)]^2 / (X \times Y) \quad (20)$$

Where,

O refers to the original image

S refers to the segmented image

X and Y refers to the rows and columns in the original and segmented images

PSNR metric computes the Peak Signal-to-Noise-Ratio (PSNR) between original and segmented image in decibels. The higher the PSNR, the better the quality of the segmented image is [41]. PSNR is a measure of peak error, derived from MSE. The formula for computing PSNR is denoted in Equ.21.

$$PSNR = 10\log_{10}(F^2/MSE) \quad (21)$$

Where,

F refers to the maximum fluctuation in the image data type.

SSIM is a multiplicative combination of luminance (l), contrast (c) and structure (St)(Ho et al., 2019), explained using the notions given from Equ.22. to Equ.27.

The luminance of comparison of original and segmented image is computed using the formula shown in Equ.22.

$$l(O, S) = (2\mu_o\mu_s + C_1)/(\mu_o^2 + \mu_s^2 + C_1) \quad (22)$$

where,

μ_o refers to the mean of original image

μ_s refers to the mean of segmented image

C_1 is a constant for stabilizing the division with weak denominator which is calculated using the Equ.23.

$$C_1 = (k1 \cdot L)^2 \quad (23)$$

Where $k1$ is a constant with value 0.01

L is dynamic range of pixel-values which is $2^{\# \text{ of pixels}} - 1$

The contrast comparison of original and segmented image may be computed using the formula shown in Equ.24.

$$C(O, S) = (2\sigma_o\sigma_s + C_2)/(\sigma_o^2 + \sigma_s^2 + C_2) \quad (24)$$

where,

σ_o refers to the variation of original image

σ_s refers to the variation of the segmented image

C_2 is a constant for stabilizing the division with weak denominator which is calculated using the Equ.25.

$$C_2 = (k2 \cdot L)^2 \quad (25)$$

Where $k2$ is a constant with value 0.03

L is dynamic range of pixel-values which is $2^{\# \text{ of pixels}} - 1$

The structural similarity of the original image and segmented image is computed using Equ.26.

$$St(O, S) = (\sigma_o S + C_3)/(\sigma_o\sigma_s + C_3) \quad (26)$$

where,

σ_o refers to the variation of original image

σ_s refers to the variation of the segmented image

$C_3 = C_2/2$

Finally, Structural Similarity Index (SSIM) is computed as the weighted combination of the comparative measures of luminance, contrast and Structure using the formula denoted in Equ.27.

$$SSIM_{(O,S)} = [l(O, S)]^\alpha \cdot [c(O, S)]^\beta \cdot [St(O, S)]^\gamma \quad (27)$$

where,

α , β and γ are weights set to 1.

Fig.5. shows the experimental images that are used in the conventional experimentation.

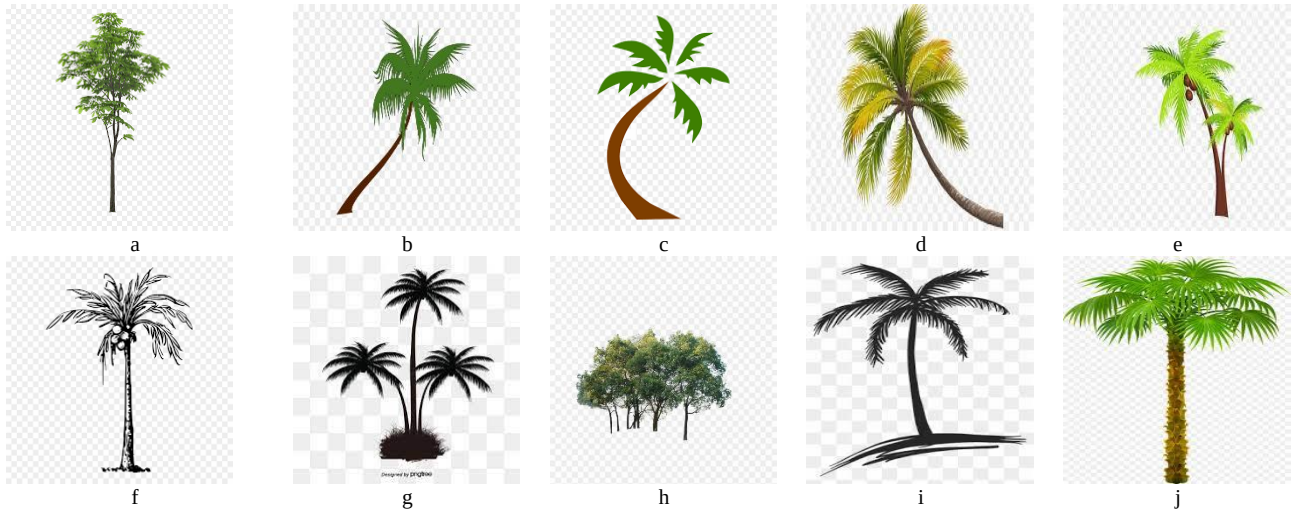


Fig. 5. Experimental Images from image dataset

The metrics are normalized to the range $[0, \infty]$ for PSNR metric and $[0, 1]$ for SSIM and MSE metrics respectively. Table.1. denotes the desirable and non-desirable values of the conventional metrics.

Table 1. Conventional Metrics - Desirable/Non-Desirable Values

Metrics	Desirable	Non-desirable
PSNR	>20	0
SSIM	1	0
MSE	0	1

4.2. Convolutional Experimentation

The objective of the convolutional experimentation is to prove that the proposed LAPSD image binarization method will be helpful for CNN models to classify images. The proposed LAPSD method acts as a pre-processing step for CNN models to separate foreground objects from background noise. The convolutional experimentation is conducted using Jupyter Notebook version 6.3.0 on a workstation with Intel Core i5 processor with Windows 10 Operating system (64 bits) installed in it. The convolutional experimentation takes the output binary images from conventional experimentation and passes the images into four VGG16 (Visual Geometry Group 16) convolution neural network models created for each of the experimental methods such as proposed LAPSD, Bradley, Niblack and Sauvola. The description of a VGG16 CNN model is shown in Fig. 6.

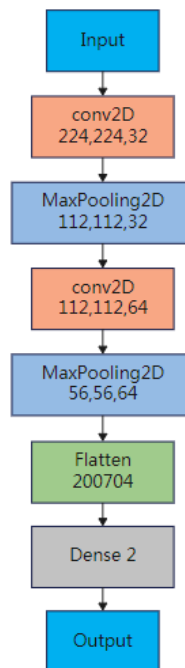


Fig. 6. VGG16 CNN architecture

The convolution layer of VGG16 CNN model, passes series of filters that constitutes a base for CNN. Filters aka kernels stride through the images to learn the patterns that helps the CNN model to classify them. Equ.28 denotes the process of convolution layer

$$G[m, n] = \sum_{j=0}^J \sum_{k=0}^K h[j, k], f[m - j, n - k] \quad (28)$$

where,

G is the output

h is the kernel

f is the input image

Equ.28 may result negative values, activation layer converts the negative values to zero, as the pixel values are to be positive numbers. ReLU activation function is best suited for activation layer. Equ.29 denotes the notion of ReLU

$$\text{ReLU} = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (29)$$

Pooling layer generalizes the patterns extracted by convolutional filters to help model recognize patterns independent of their location in the image. It also performs the down-sampling of image by reducing spatial size of the input image. The CNN models used in the experimentation uses max pooling where the maximum value in each stride is down sampled and sent to next layers in CNN.

The output produced by pooling layer is a 2-dimensional array. Flattening layer converts the 2D array into a single long continuous linear vector, which is fed as input to dense layer to classify the images.

In dense layer (fully connected layer) each neuron in the CNN model receives input from all neurons from the previous layer. Number of nodes in this layer is same as number of classes in the problem domain. Fully connected layer can be represented using the formula given in Equ.30.

$x \in R^N$ and $y_i \in R$ is the output of i^{th} node of fully connected layer, then

$$y_i = \sigma(\sum x_i w_i) \quad (30)$$

where,

σ is an activation function

$x_i w_i$ is the learnable parameter of the network

The performance of the proposed image binarization method is verified using two metrics in convolutional experimentation called accuracy and loss metrics. Accuracy metric is the fraction of correct predictions with wrong predictions in image classification. Equ.31 denotes the formula for calculating accuracy.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (31)$$

Where,

True Positive (TP): images classified correctly in group x

False Positive (FP): images classified incorrectly in group x

True Negative (TN): images classified correctly as not in group x

False Negative (FN): images classified incorrectly as not in group x

Loss metric is a penalty to the model for an incorrect prediction. Loss value indicates how poor the model's prediction on a single example. The formula for a loss metric is denoted in Equ.32

$$\text{Loss} = 1/N \sum_{(x,y) \in D} (y - \text{prediction}(x))^2 \quad (32)$$

where,

x is a testing image

y is the class

$\text{prediction}(x)$ is the class predicted by the CNN model for the testing image x

D is the testing data

N is the number of images in testing data

Table 2 denotes the desirable and non-desirable values of the convolutional metrics.

Table 2. Convolutional Metrics - Desirable/Non-Desirable Values

Metrics	Desirable	Non-desirable
Accuracy	1	0
Loss	0	1

Fig.7. shows the sample images that are used in the convolutional experimentation.

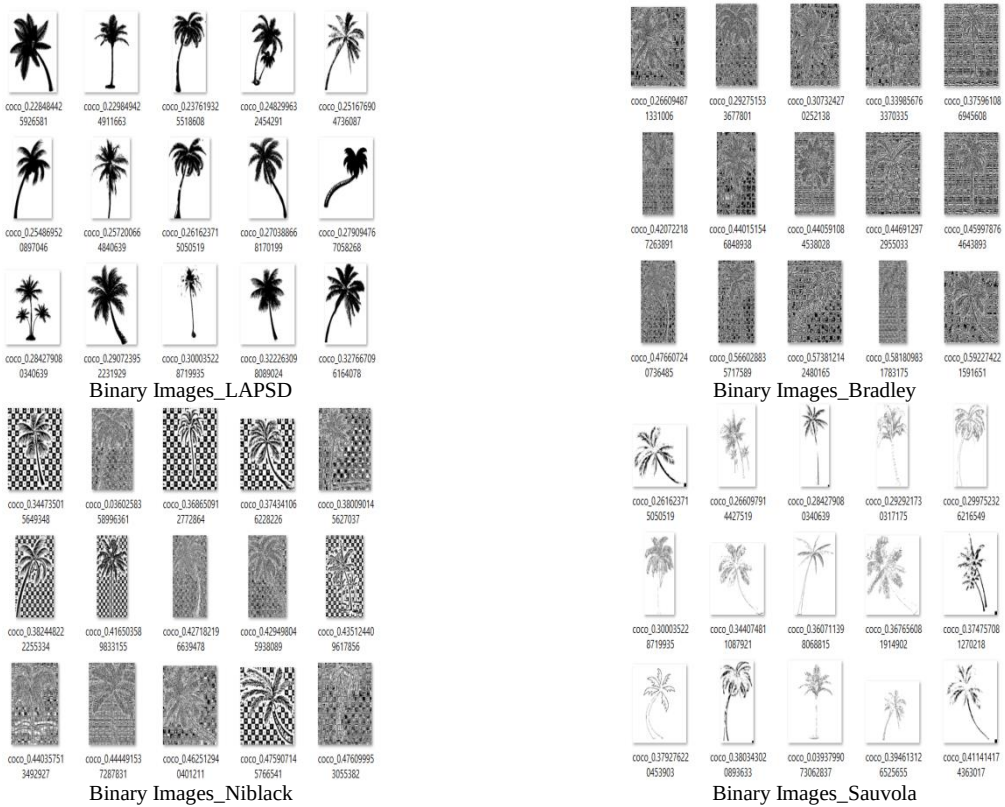


Fig. 7. Sample Image Dataset for Convolutional Experimentation

The framework of the experimentation is shown in Fig.8.

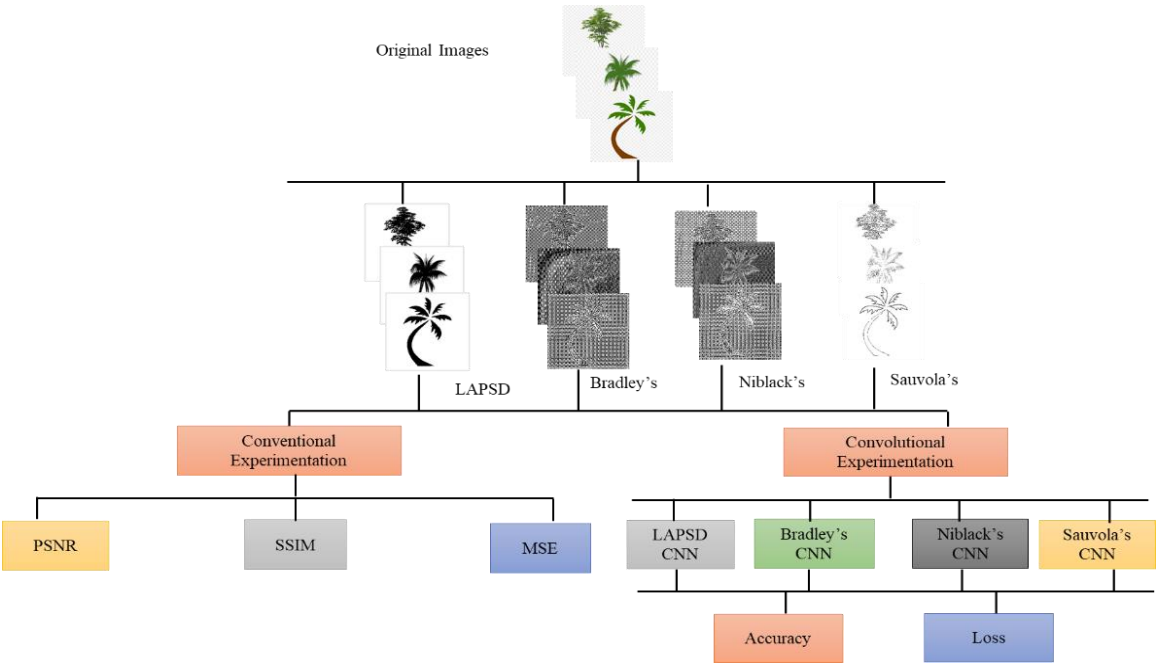
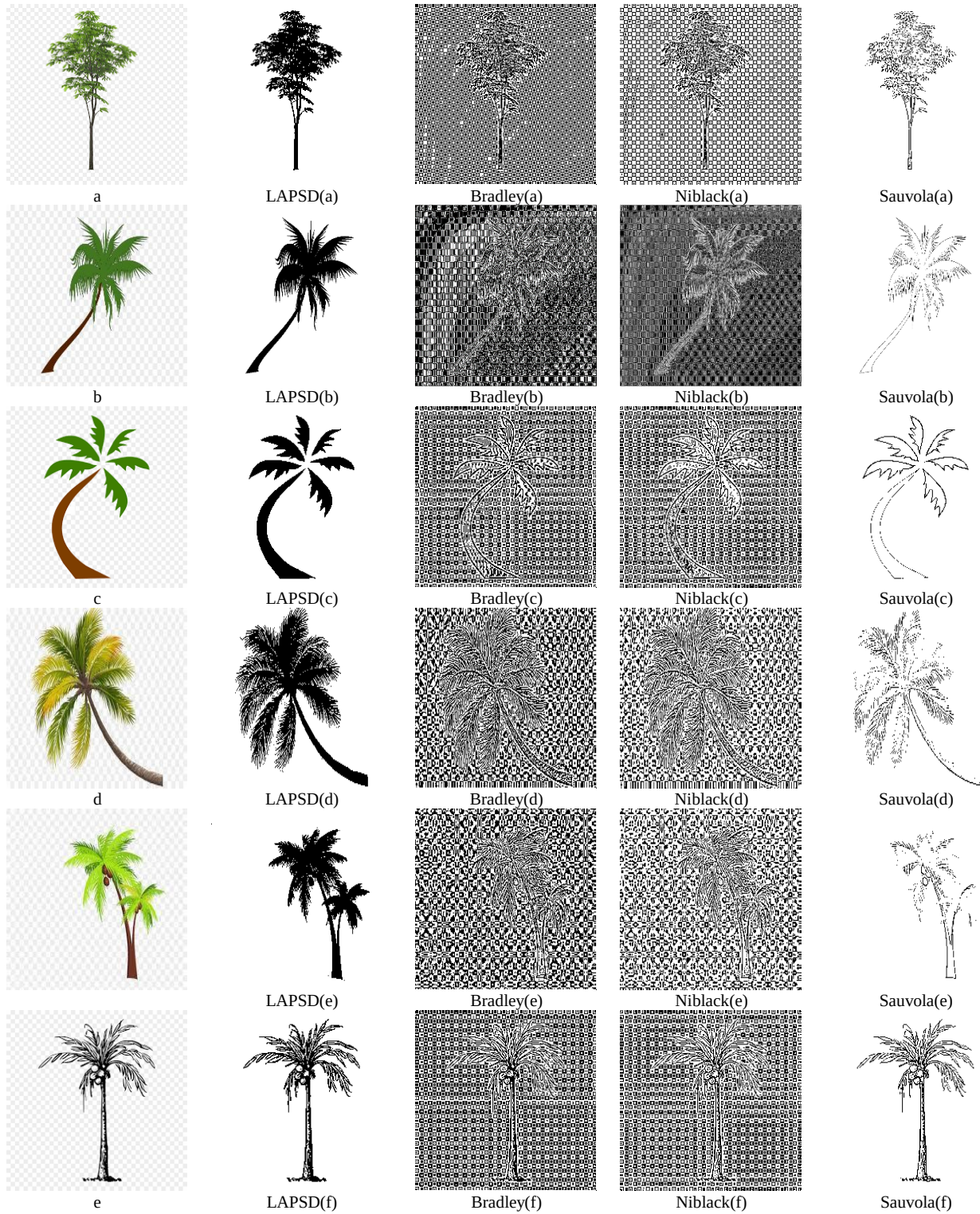


Fig. 8. Experimentation Framework

5. Interpretation

5.1. Conventional experimentation

The output of the LAPSD methodology and other local adaptive thresholding methods discussed in the literature such as Bradley's, Niblack's and Sauvola's Methods with window size 'w' as 3 is shown in Fig.9. The binarization outputs produced by the LAPSD look better than the other binarization methods.



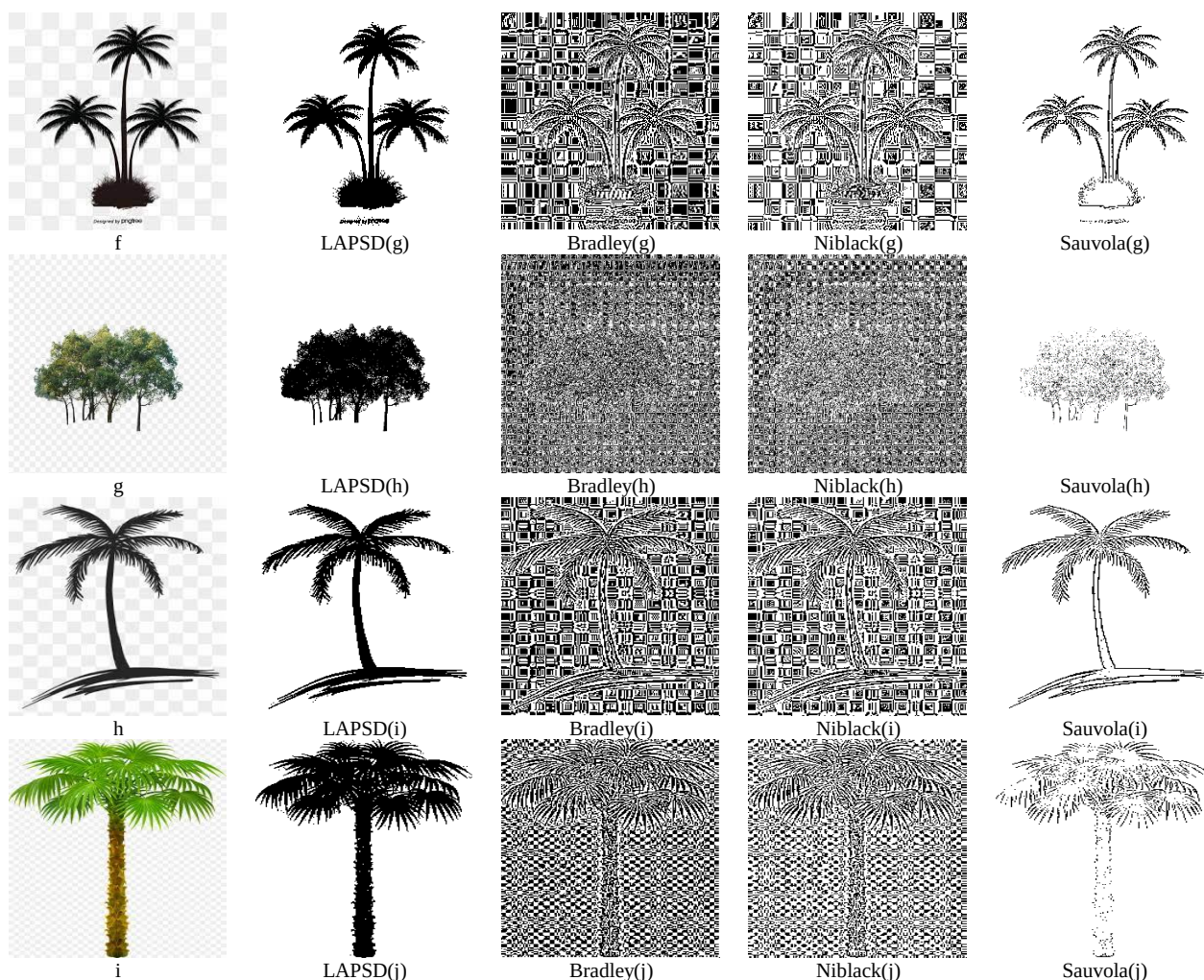
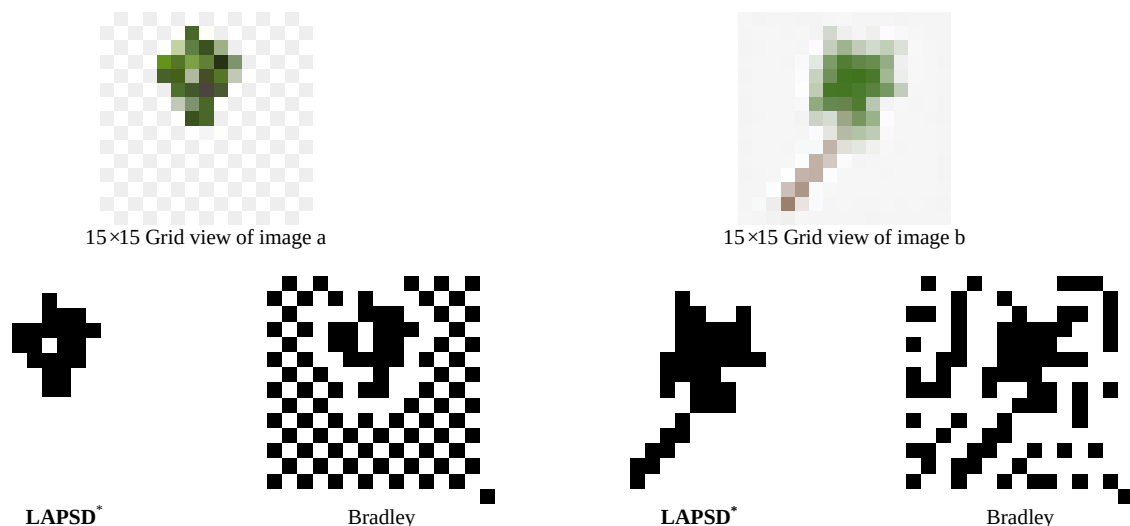


Fig. 9. Binarization results of Experimental Images with window size '3'

From the output images, it has been observed, for the given window size '3', the proposed LAPSD method precisely performs image binarization, removes water marking and clearly separates the foreground from background of the experimental images. Other experimental methods such as Bradley, Niblack and Sauvola could be seen struggling with image binarization, removal of water marking and separation of foreground from background images. The binarization output of 15×15 Grid view of two images 'a' and Image 'b' of the experimental images is shown in Fig.10. The best value for the SSIM is indicated with asterisks and in bold. The images present different types of perturbations that affect accurate binarization. The results clearly show that the proposed LAPSD binarization method overcomes the impediments and perturbations with accurate binarization than the other experimental methods.



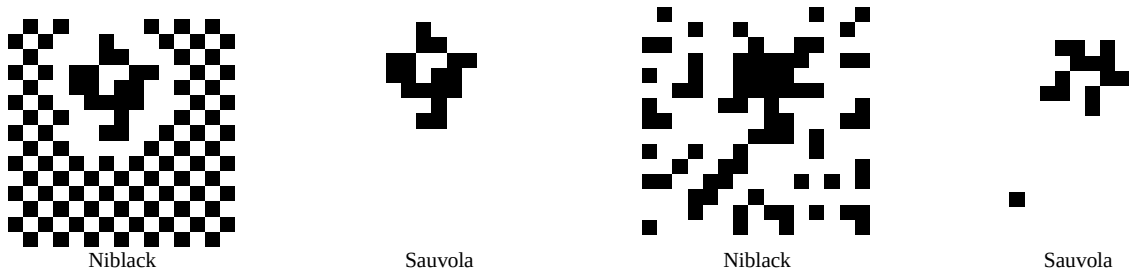


Fig. 10. Binarization result of LAPSD, Bradley, Niblack's and Sauvola's method over 15×15 Grid View of Images a and b

Though LAPSD, Bradley's, Niblack's and Sauvola's methods work based on integral sum of input image, threshold wise adjustments are possible only in proposed LAPSD and Bradley's methods. One can choose different threshold values with varying window size 'w' in both methods. On the other hand, Niblack's and Sauvola's methods use the standard threshold value 0.5 with varying window sizes 'w'. Thus, the comparisons of the binarization methods are done using a standard threshold value 0.5 with different window sizes. Tables 3-5 shows the conventional experimentation results of LAPSD, Bradley's, Niblack's and Sauvola's methods with varying window sizes such as 3, 13 and 23. The pictorial representations of Tables 3-5 have been presented in Figures 11 – 13 respectively.

Table 3. Conventional Experimentation - PSNR Analysis

Window Size	Method	Experimental Images									
		a	b	c	d	e	f	g	h	i	j
3	LAPSD	20.69	19.26	28.33	24.37	13.28	16.18	18.49	17.52	17.21	23.92
	Bradley	3.40	3.15	2.41	3.27	3.92	3.57	3.43	3.26	3.51	3.93
	Niblack	4.82	3.37	4.50	4.57	3.64	4.51	3.95	4.52	3.43	4.014
	Sauvola	11.09	8.51	10.16	10.21	6.97	10.47	13.87	9.70	9.69	7.05
13	LAPSD	22.36	20.88	28.83	26.85	15.78	18.79	18.56	18.02	17.68	24.43
	Bradley	14.15	11.27	13.27	13.19	11.61	16.29	17.30	11.68	10.77	11.30
	Niblack	15.46	12.19	10.29	10.66	6.57	10.31	13.53	10.19	10.39	6.59
	Sauvola	10.42	8.284	10.22	10.73	6.40	10.16	13.36	10.42	10.64	6.39
23	LAPSD	26.59	24.37	32.38	29.86	16.34	19.08	24.92	21.50	22.93	28.41
	Bradley	9.40	7.72	9.46	9.34	10.25	11.32	15.01	12.24	13.17	12.97
	Niblack	4.38	4.01	4.14	5.54	6.58	4.44	4.57	6.79	7.63	7.19
	Sauvola	18.50	11.91	15.81	24.32	11.89	14.39	19.48	17.70	17.83	12.55

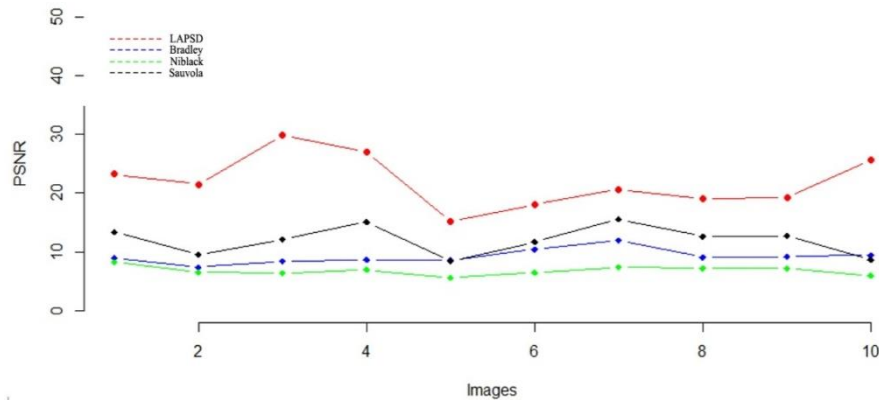


Fig. 11. Conventional Experimentation - PSNR Analysis

Table 4. Conventional Experimentation – SSIM Analysis

Window Size	Method	Experimental Images									
		a	b	c	d	e	f	g	h	i	j
3	LAPSD	0.99984	0.99978	0.9999748	0.9999364	0.9989566	0.999775	0.9997381	0.999665	0.9996373	0.99993
	Bradley	0.9752776	0.977535	0.9611171	0.9773707	0.9870307	0.9770305	0.9742748	0.9791494	0.9813596	0.9871097
	Niblack	0.9882139	0.9801978	0.9918483	0.9865445	0.9778839	0.9900567	0.9866147	0.9921865	0.9919702	0.9907123
	Sauvola	0.998046	0.9955018	0.9973988	0.997432	0.9922381	0.9976351	0.9991267	0.9969527	0.9969353	0.9924356
13	LAPSD	0.9999	0.9998598	0.99997	0.99996	0.99949	0.99977	0.99976	0.99971	0.99969	0.9999384
	Bradley	0.9984344	0.9966634	0.9980878	0.9980806	0.9963003	0.9989477	0.9992861	0.9973992	0.9965186	0.9959852
	Niblack	0.9990999	0.9975748	0.9975018	0.9977659	0.9910684	0.9975202	0.9990384	0.997389	0.9975439	0.9911109
	Sauvola	0.9976182	0.9951486	0.9974454	0.9978333	0.9905564	0.9974213	0.9990032	0.9975907	0.9977459	0.9904707
23	LAPSD	0.9999626	0.9999376	0.9999902	0.9999824	0.9995587	0.9995372	0.9999442	0.9998742	0.9999109	0.9999755
	Bradley	0.9967153	0.9941428	0.996749	0.996652	0.9974102	0.9981768	0.9993579	0.9985983	0.9989296	0.9988561
	Niblack	0.978816	0.9797296	0.9766225	0.9864338	0.9916478	0.9791252	0.9804407	0.9917603	0.9939367	0.9931768
	Sauvola	0.9997386	0.9984569	0.9994802	0.9999357	0.9984327	0.9992406	0.9997948	0.9996798	0.9996899	0.9987073

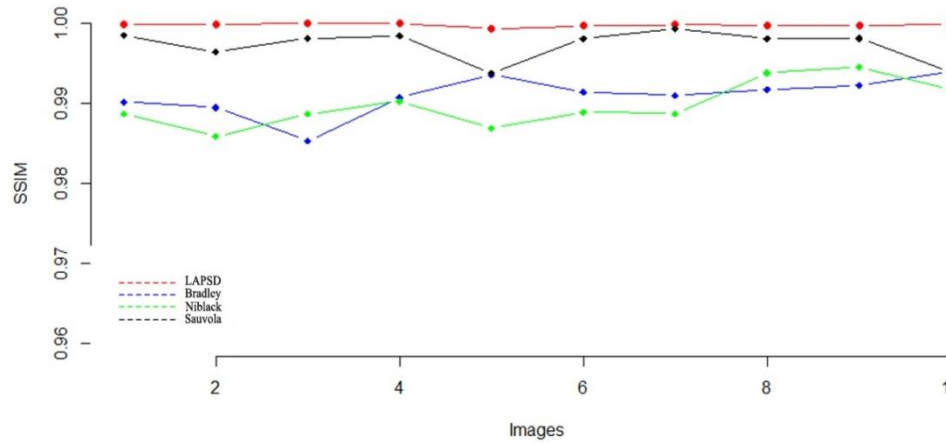


Fig. 12. Conventional Experimentation – SSIM Analysis

Table 5. Conventional Experimentation – MSE Analysis

Window Size	Method	Experimental Images									
		a	b	c	d	e	f	g	h	i	j
3	LAPSD	0.008	0.011	0.002	0.003	0.046	0.013	0.014	0.017	0.018	0.004
	Bradley	0.457	0.483	0.573	0.470	0.405	0.439	0.453	0.471	0.445	0.403
	Niblack	0.329	0.459	0.354	0.348	0.431	0.353	0.401	0.353	0.453	0.396
	Sauvola	0.077	0.140	0.096	0.095	0.200	0.089	0.040	0.106	0.107	0.197
13	LAPSD	0.005	0.008	0.001	0.002	0.026	0.012	0.013	0.015	0.017	0.003
	Bradley	0.038	0.074	0.047	0.047	0.068	0.023	0.018	0.067	0.083	0.074
	Niblack	0.028	0.060	0.093	0.085	0.219	0.092	0.044	0.095	0.091	0.218
	Sauvola	0.090	0.148	0.094	0.084	0.228	0.096	0.046	0.090	0.086	0.229
23	LAPSD	0.002	0.003	0.000	0.001	0.023	0.024	0.003	0.007	0.005	0.001
	Bradley	0.114	0.168	0.113	0.116	0.094	0.073	0.031	0.059	0.048	0.050
	Niblack	0.364	0.397	0.385	0.279	0.219	0.359	0.348	0.209	0.172	0.190
	Sauvola	0.014	0.064	0.026	0.003	0.064	0.036	0.011	0.016	0.016	0.055

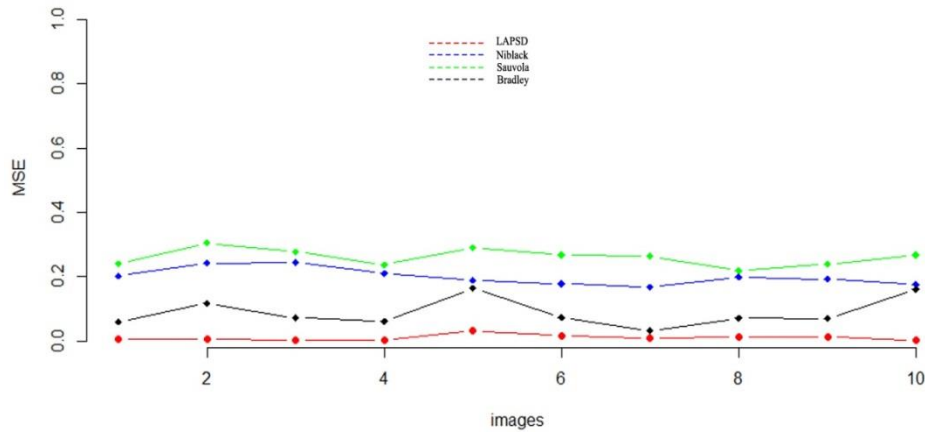


Fig. 13. Conventional Experimentation – MSE Analysis

From the results it has been observed that the proposed LAPSD method does not introduce additional noise during binarization and does not get affected by low contrast and illumination issues. But, other methods such as Bradley, Niblack and Sauvola methods suffer from the above said issues. In addition, the proposed LAPSD method achieves desirable values in all metrics of the conventional experimentation than the other binarization methods. It has also been noticed that the performance of LAPSD, Bradley, Niblack and Sauvola methods are improving with ascending window sizes.

5.2. Convolutional experimentation

Convolutional experimentation is conducted using four VGG16 based CNN models, as they are most common and widely used image classifiers. Each CNN model takes 180 images of training data and 60 images of testing and validation data created by each of the experimental methods such as LAPSD, Bradley, Niblack and Sauvola. The dataset contains colored clipart images of ‘COCONUT’ and ‘FIG’ Trees. The objective of the CNN model is to classify the ‘COCONUT’ and ‘FIG’ trees in the respective classes. The images are binarized using the experimental methods

and are given as input to the corresponding CNN model to measure the performance of those methods in terms of image classification. The performance of CNN models is measured using accuracy and loss metrics and shown in Table. 6, followed by the graphical representation in Fig.14.

Table 6. Convolutional Experimentation – Accuracy and Loss Analysis

Epochs	LAPSD		Bradley		Niblack		Sauvola	
	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
Epoch 1	0.9667	3.0798	0.7333	4.4556	0.5000	37.210	0.9167	6.5341
Epoch 2	0.9333	3.0348	0.7333	6.3767	0.5000	20.629	0.9167	7.4981
Epoch 3	0.9667	2.0556	0.5667	9.6576	0.6500	8.6052	0.9167	2.3434
Epoch 4	0.9833	0.0256	0.4833	16.025	0.7500	8.6448	0.8833	5.0113
Epoch 5	0.9833	0.1305	0.6500	6.7212	0.5333	18.8265	0.9500	2.5936
Epoch 6	0.9833	0.8554	0.5333	24.866	0.8167	4.3464	0.9667	1.1832
Epoch 7	0.9833	0.9739	0.8667	1.8440	0.9000	2.6268	0.9500	2.2214
Epoch 8	0.9833	0.9931	0.7333	5.0723	0.8833	2.9397	0.8833	5.0113
Epoch 9	0.9833	0.9962	0.8500	3.2287	0.6333	11.761	0.9667	0.5206
Epoch 10	0.9833	0.9966	0.8167	5.4042	0.6333	14.0985	0.9500	3.2019

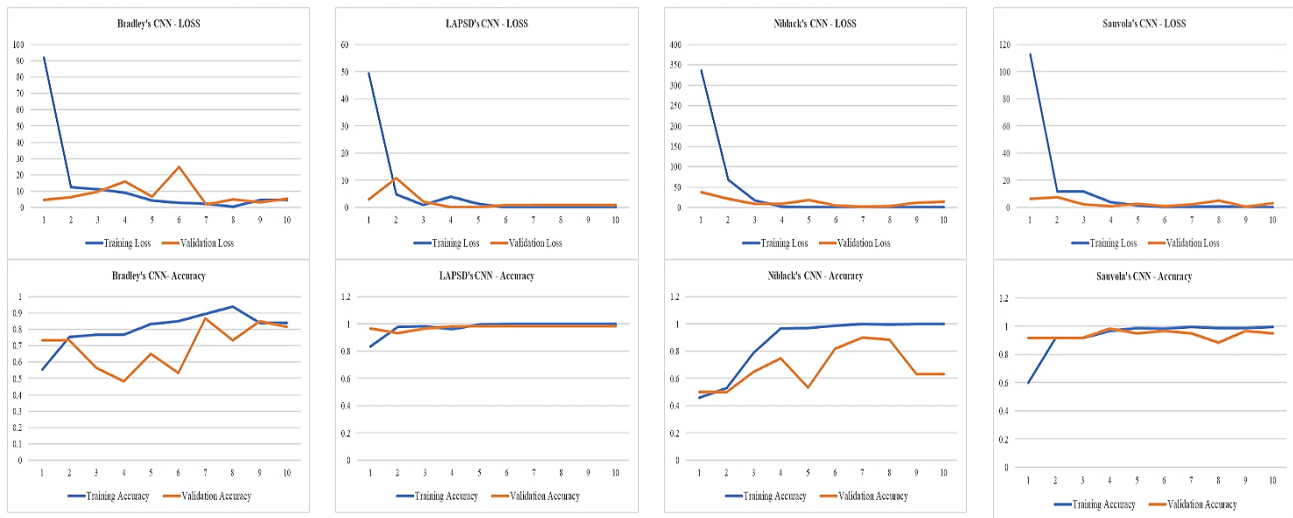
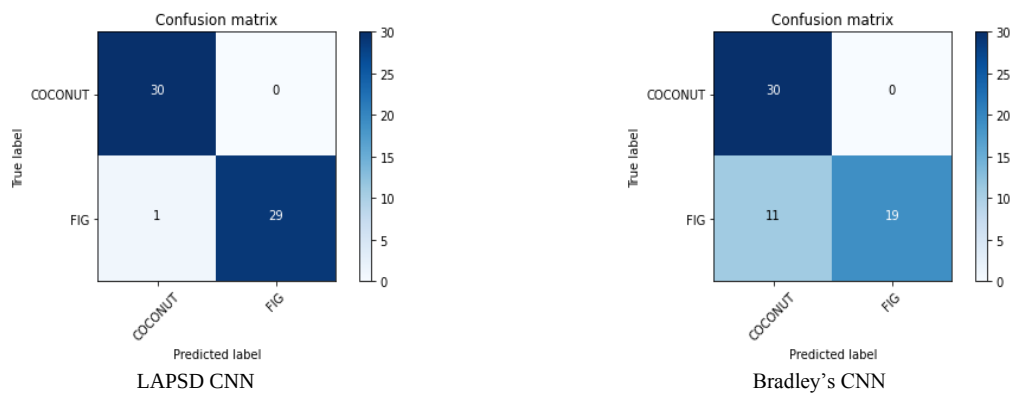


Fig. 14. Convolutional Experimentation – Accuracy and Loss Analysis

The confusion matrix presented in Fig.15 denotes the performance of the CNN models in terms of classifying the two classes of trees such as 'COCONUT' as 'COCONUT', 'FIG' as 'FIG', 'COCONUT' as 'FIG' and 'FIG' as 'COCONUT'. From the figure, it has been observed, the CNN model built using the binarized images of LAPSD method is precise and accurate than the other models by correctly classifying the 'COCONUT' and 'FIG' trees with the accuracy of 0.98. Thus, it is clearly evident the proposed LAPSD binarization methods can be used as a pre-processing method for CNN models.



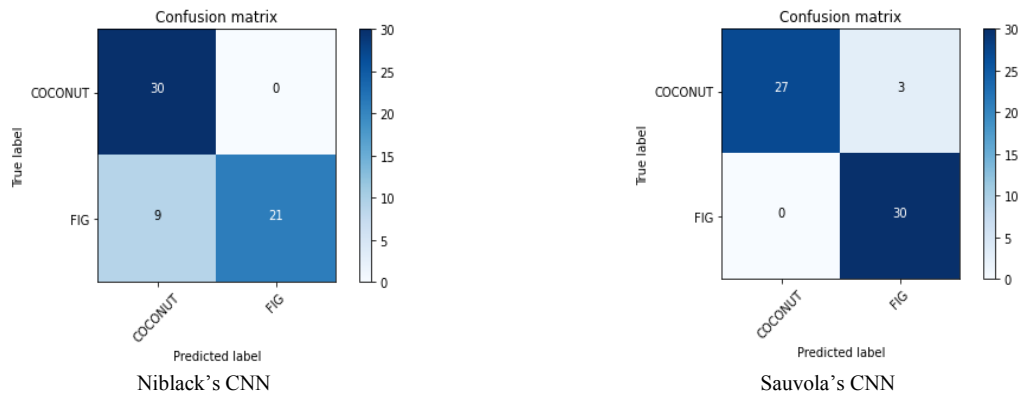


Fig. 15. Convolutional Experimentation - Confusion Matrix of VGG16 CNN Models

6. Conclusions

The paper proposes a new local adaptive thresholding method LAPSD for image binarization, which is a continuation of our global thresholding method 'PSD' [46]. The method uses a novel calculation upon integral sum for computing efficient thresholds for image binarization. The proposed LAPSD method has two tuning parameters such as threshold percentage 'tp' and window size 'w' that help to obtain optimum local adaptive thresholds. It has been shown that, compared to the other local adaptive image binarization methods, our method improves image binarization, by obtaining best results in PSNR, SSIM, MSE, Accuracy and Loss metrics without any additional computational complexity. Thus, the proposed method turns out to be very stable and consistent method that obtains correct binarization thresholds for real-world images which are disturbed with lighting conditions, color contrasts and background and foreground noises. Due to its impressive time and accurate performances, the new LAPSD method could be installed in deep learning models for obtaining better accuracies in image classification. The results obtained by the proposed LAPSD method, both from theoretical and experimental points of view are found to be promising. We also expect that our new local adaptive thresholding method, with its ease of implementation and wide range of use, gives new research opportunities in real-time image binarization applications. In this work, we brute force the tuning parameters externally and varies from image to image. In future, this work may be extended with appropriate optimization technique to identify the best tuning parameters for the image.

References

- [1] Lu, T. and Dooms, A., 2021. Probabilistic homogeneity for document image segmentation. *Pattern Recognition*, 109, p.107591.
- [2] Mittal, M., Arora, M., Pandey, T. and Goyal, L.M., 2020. Image segmentation using deep learning techniques in medical images. In *Advancement of machine intelligence in interactive medical image analysis* (pp. 41-63). Springer, Singapore.
- [3] Kaur, S., Kaur, N. and Kaur, K., 2020. Image segmentation: techniques and its applications. *Journal of Natural Remedies*, 21(2), pp.79-87.
- [4] Tomchuk, K.K. and Volkov, V.Y., 2020, June. Analysis of the Quality of Thresholding for Object Detection in Images. In *2020 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF)* (pp. 1-5). IEEE.
- [5] Mehta, N., Braun, P.X., Gendelman, I., Alibhai, A.Y., Arya, M., Duker, J.S. and Waheed, N.K., 2020. Repeatability of binarization thresholding methods for optical coherence tomography angiography image quantification. *Scientific reports*, 10(1), pp.1-11.
- [6] Minaee, S., Boykov, Y.Y., Porikli, F., Plaza, A.J., Kehtarnavaz, N. and Terzopoulos, D., 2021. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- [7] Adegun, A. and Viriri, S., 2021. Deep learning techniques for skin lesion analysis and melanoma cancer detection: a survey of state-of-the-art. *Artificial Intelligence Review*, 54(2), pp.811-841.
- [8] Aslam, Y., Santhi, N., Ramasamy, N. and Ramar, K., 2020. Localization and segmentation of metal cracks using deep learning. *Journal of Ambient Intelligence and Humanized Computing*, pp.1-9.
- [9] Yamni, M., Daoui, A., Karmouni, H., Sayyouri, M., Qjidaa, H. and Flusser, J., 2020. Fractional Charlier moments for image reconstruction and image watermarking. *Signal Processing*, 171, p.107509.
- [10] Volkov, V.Y., Bogachev, M.I. and Kayumov, A.R., 2020. Object selection in computer vision: from multi-thresholding to percolation based scene representation. In *Computer Vision in Advanced Control Systems-5* (pp. 161-194). Springer, Cham.
- [11] Nichele, L., Persichetti, V., Lucidi, M. and Cincotti, G., 2020. Quantitative evaluation of ImageJ thresholding algorithms for microbial cell counting. *OSA Continuum*, 3(6), pp.1417-1427.
- [12] Michalak, H. and Okarma, K., 2020. Optimization of Degraded Document Image Binarization Method Based on Background Estimation.
- [13] Abdusalomov, A., Mukhiddinov, M., Djuraev, O., Khamdamov, U. and Whangbo, T.K., 2020. Automatic salient object extraction based on locally adaptive thresholding to generate tactile graphics. *Applied Sciences*, 10(10), p.3350.
- [14] Rani, N.S., Karthik, U. and Ranjith, S., 2020. Extraction of Gliomas from 3D MRI Images using Convolution Kernel Processing and Adaptive Thresholding. *Procedia Computer Science*, 167, pp.273-284.

- [15] Li, J., Hu, B. and Cao, Z., 2020, August. A new QR Code Recognition Method using Deblurring and Modified Local Adaptive Thresholding Techniques. In 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE) (pp. 1269-1274). IEEE.
- [16] Liang, J. and Liu, D., 2020. A local thresholding approach to flood water delineation using Sentinel-1 SAR imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 159, pp.53-62.
- [17] Ganem, J., Thureau, S., Gardin, I., Modzelewski, R., Hapdey, S. and Vera, P., 2018. Delineation of lung cancer with FDG PET/CT during radiation therapy. *Radiation Oncology*, 13(1), pp.1-8.
- [18] Rahimi-Nasrabadi, H., Jin, J., Mazade, R., Pons, C., Najafian, S. and Alonso, J.M., 2021. Image luminance changes contrast sensitivity in visual cortex. *Cell reports*, 34(5), p.108692.
- [19] Zhou, F., Jia, Z., Yang, J. and Kasabov, N., 2017. Method of improved fuzzy contrast combined adaptive threshold in NSCT for medical image enhancement. *BioMed research international*, 2017.
- [20] Singh, P., 2020. A neutrosophic-entropy based adaptive thresholding segmentation algorithm: A special application in MR images of Parkinson's disease. *Artificial Intelligence in Medicine*, 104, p.101838.
- [21] Yahya, A.A., Tan, J., Su, B., Liu, K. and Hadi, A.N., 2019. Image noise reduction based on adaptive thresholding and clustering. *Multimedia Tools and Applications*, 78(11), pp.15545-15573.
- [22] Wang, C., Chen, Z., Wang, Y. and Wang, Z., 2018. Denoising and 3D reconstruction of CT images in extracted tooth via wavelet and bilateral filtering. *International Journal of Pattern Recognition and Artificial Intelligence*, 32(05), p.1854010.
- [23] Cheremkhin, P.A. and Kurbatova, E.A., 2019. Comparative appraisal of global and local thresholding methods for binarisation of off-axis digital holograms. *Optics and Lasers in Engineering*, 115, pp.119-130.
- [24] Buayai, P., Kantanukul, T., Leung, C.K. and Runapongsa Saikaew, K., 2017. Boundary detection of pigs in pens based on adaptive thresholding using an integral image and adaptive partitioning.
- [25] Thanh, D.N., 2020. An adaptive local thresholding roads segmentation method for satellite aerial images with normalized HSV and lab color models. In *Intelligent Computing in Engineering* (pp. 865-872). Springer, Singapore.
- [26] Gan, H.S., Salam, B.A.J.A., Khaizi, A.S.A., Ramlee, M.H., Mahmud, W.M.W., Lee, Y.S., Sayuti, K.A. and Musa, A.T., 2019. Local mean based adaptive thresholding to classify the cartilage and background superpixels. *Indonesian Journal of Electrical Engineering and Computer Science*, 15(1), pp.221-230.
- [27] Nandy, M. and Banerjee, M., 2017, October. A comparative analysis of application of niblack and sauvola binarization to retinal vessel segmentation. In 2017 3rd International Conference on Computational Intelligence and Networks (CINE) (pp. 105-109). IEEE.
- [28] Saddami, K., Munadi, K., Muchallil, S. and Arnia, F., 2017, November. Improved thresholding method for enhancing Jawi binarization performance. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) (Vol. 1, pp. 1108-1113). IEEE.
- [29] Michalak, H. and Okarma, K., 2018, April. Fast adaptive image binarization using the region based approach. In *Computer Science On-line Conference* (pp. 79-90). Springer, Cham.
- [30] Michalak, H. and Okarma, K., 2018, May. Region based adaptive binarization for optical character recognition purposes. In 2018 International Interdisciplinary PhD Workshop (IIPhDW) (pp. 361-366). IEEE.
- [31] Michalak, H. and Okarma, K., 2019. Improvement of image binarization methods using image preprocessing with local entropy filtering for alphanumeric character recognition purposes. *entropy*, 21(6), p.562.
- [32] Mustafa, W.A., Yazid, H. and Jaafar, M., 2018. An improved sauvola approach on document images binarization. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 10(2), pp.43-50.
- [33] Kurniawan, M.R., 2021. Segmentasi dan Perhitungan Luas Bercak Putih Citra Rontgen Paru Penderita Covid-19 dengan Metode Sauvola Thresholding (Doctoral dissertation, Universitas Jenderal Soedirman).
- [34] Kiran, M., Ahmed, I., Khan, N. and Reddy, A.G., 2019. Chest X-ray segmentation using Sauvola thresholding and Gaussian derivatives responses. *Journal of Ambient Intelligence and Humanized Computing*, 10(10), pp.4179-4195.
- [35] Mustafa, W.A., Abdul-Nasir, A.S. and Mohamed, Z.E.E.H.A.I.D.A., 2018. Malaria parasites segmentation based on Sauvola algorithm modification. *Malaysian Appl. Biol.*, 47(2), pp.71-76.
- [36] Panagiotakis, C. and Argyros, A.A., 2018, October. Cell segmentation via region-based ellipse fitting. In 2018 25th IEEE International Conference on Image Processing (ICIP) (pp. 2426-2430). IEEE.
- [37] Misra, N.N., Phalak, R. and Martynenko, A., 2018. A microscopic computer vision algorithm for autonomous bubble detection in aerated complex liquids. *Journal of food engineering*, 238, pp.54-60.
- [38] Jamaluddin, M.I., Mustafa, W.A., Idrus, S.Z.S., Jamlos, M.A. and Khairunizam, W., 2020, April. Thresholding Comparison Display Using Graphical User Interface (GUI) with Mobile Notification. In *Journal of Physics: Conference Series* (Vol. 1529, No. 4, p. 042022). IOP Publishing.
- [39] https://favpng.com/png_search/tree/
- [40] Huynh-Thu, Q. and Ghanbari, M., 2008. Scope of validity of PSNR in image/video quality assessment. *Electronics letters*, 44(13), pp.800-801.
- [41] Hore, A. and Ziou, D., 2010, August. Image quality metrics: PSNR vs. SSIM. In 2010 20th international conference on pattern recognition (pp. 2366-2369). IEEE.
- [42] Wang, S., Rehman, A., Wang, Z., Ma, S. and Gao, W., 2011. SSIM-motivated rate-distortion optimization for video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(4), pp.516-529.
- [43] Moll, M.A., Baird, H.S. and An, C., 2008, September. Truthing for pixel-accurate segmentation. In 2008 The Eighth IAPR International Workshop on Document Analysis Systems (pp. 379-385). IEEE.
- [44] Garnavi, R., Aldeen, M., Celebi, M.E., Varigos, G. and Finch, S., 2011. Border detection in dermoscopy images using hybrid thresholding on optimized color channels. *Computerized Medical Imaging and Graphics*, 35(2), pp.105-115.
- [45] Garnavi, R., Aldeen, M., Celebi, M.E., Bhuiyan, A., Dolianitis, C. and Varigos, G., 2010. Automatic segmentation of dermoscopy images using histogram thresholding on optimal color channels. *International Journal of Medicine and Medical Sciences*, 1(2), pp.126-134.
- [46] Christy, A. Joy, and A. Umamakeswari. "A Novel Percentage Split Distribution Method for Image Thresholding." *Optik* 218 (2020): 164953.

Authors' Profiles



A. Joy Christy, Assistant Professor in School of Computing, SASTRA Deemed to be University. Her research areas include Data Analysis and Data Mining, Big Data, Image Segmentation and Convolutional Neural Networking. She has published good number of articles in reputed and refereed journals in the areas of statistical analysis and data mining, machine learning, medical imaging, image segmentation and software metrics.



A. Umamakeswari, Professor and Dean, School of Computing, SASTRA Deemed University, Thanjavur has research experience in the areas of Wireless Sensor Networks, Cloud Computing, Internet of Things, Machine Learning, Embedded Systems, and Block chain. She has been an integral part of completing projects funded by the Government Institutions on content development for Software Testing, Microprocessors & Interfacing (NMEICT-MHRD) and Design of Logic tools for Process Modelling (IGCAR). She has guided research works that feature the development of Intrusion Detection Systems (IDS) for Wireless Sensor Networks & Industrial Cyber-Physical Systems (CPS), computation offloading for Edge environments, Cloud Security, Block chain-based solutions for the Internet of Things and Predicting Air quality in Urban environments using Machine Learning. She has published good number of papers in reputed journals in the areas of Internet of Things, Machine Learning and Cyber Physical Systems.



P. Shanthi, Assistant Professor, SASTRA Deemed to be University – PhD in Computer Science and Engineering, SASTRA. Bio – Shanthi graduated with a Masters in Industrial Engineering from Anna University, Chennai and obtained her PhD from SASTRA University. She completed her Bachelors in Computer Science and Engineering at Shanmugha College of Engineering, Thanjavur. She has 16 years of teaching experience. Her research interests include Cloud computing, Information security, High performance computing and Quantum computing.



A. srilakshmi working in sastra deemed to be university in the department of computer science. Her research area is deep learning with image processing. Attended conference and workshops related to deep learning. She has more than 11 years of teaching experience and 2 years if industry experience. She is an enthusiastic, self-motivated, reliable, responsible and hard-working person. Currently she has also started working in IoT with deep learning to improve the technical skills.



Siva Chandrasekaran is a Senior Lecturer (Computer and Software Engineering), Academic Senate (elected), and Major Discipline Co-Ordinator (Software Engineering) in the School of Science, Computing, and Engineering Technologies at the Swinburne University of Technology. Siva has made significant contributions to engineering education research and published more than 140 scholarly articles in national/international peer-reviewed journals and conference proceedings. Siva is an active researcher, and his research is in Artificial Intelligence, Augmented Reality, Industrial digital transformation, and Engineering practice. Siva conducts world-class research and fosters cross-disciplinary collaboration, and his research is supported by Australian government funding agencies and commercial partners (DFAT, CSIRO, Australian Academy of Science, and the Department of Industry, Science, Energy, and Resources) around Australia.

How to cite this paper: Joy Christy A., Umamakeswari A., Shanthi P., Srilakshmi A., Siva Chandrasekaran, "A Novel Local Adaptive Percentage Split Distribution Method for Image Binarization and Classification", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.17, No.2, pp. 29-46, 2025. DOI:10.5815/ijigsp.2025.02.02