

# Building Drawing Line Art with High Pass Filtering and Image Morphology

Teady Matius Surya Mulyana<sup>1\*</sup>, Destriana Widyaningrum<sup>2</sup>, Jusia Amanda Ginting<sup>3</sup>,  
Theophilus Louie Surya Mulyana<sup>4</sup>

<sup>1,2,3</sup>Faculty of Engineering and Design, Department of Informatics, Universitas Bunda Mulia, Jakarta, Indonesia

<sup>4</sup>Primary School, Pelita Bangsa, West Jakarta, Indonesia

**Abstract:** Coloring activities can be done conventionally using paper and colored pencils, can also be done with facilities and media based on technological advances. One of them is by using computers in coloring activities, so that apart from being a stimulus for motor sensor visual abilities, it can also be a means for computer recognition and learning for children and mentally retarded children. The provision of line-art drawings for coloring media that are suitable for the skill level of students that are available free of charge do not always match the needs of students. It will be easier if line-art can be generated from images from cartoons, animated images or photos that can be easily obtained. To overcome this problem, we need an application that can automatically process an input image into a line-art image. High pass filtering 0 can produce the edges of the image object. The results of edge detection can be combined with a binary image morphology process to produce the desired line-art thickness. The purpose of this research is to develop an algorithm to create line-art automatically based on the input image with parts of the object that are more closed so that when colored with fill color operation, the color does not overflow out of the part that should be. The result of this research will be a sequence of processes to produce lines from an image object in the form of line-art to be applied to a computer-based coloring program.

**Keywords:** Edge detection, high pass filtering, image, line-art, morphology.

## 1. Introduction

Improving the visual abilities of motor sensors can be done through coloring pictures. Coloring activities can be done conventionally using paper and colored pencils, can also be done with facilities and media based on technological advances. One of them is by using a computer in coloring activities, so that apart from being a stimulus for visual motor sensor abilities, it can also be a means for computer recognition and learning for children and mentally retarded children [1], [2]. Coloring activities also increase students' creativity [3]. This coloring activity is carried out on a line-art drawing, a drawing in the form of a line dividing the parts of the image.

Line-art or Line drawing is an image that only contains lines forming objects in the image. An image containing line-art is presented for various purposes, such as aesthetics or for coloring activities [4]-[6].

Line-art can be formed by performing point detection with a

high pass filtering process using a high pass filter with a total coefficient of 0. [7], [8] However, the resulting line-art is often interrupted because there are parts of the image that have very thin grayscale values compared to adjacent parts [9], [4].

In cartoon images that have color gradations will produce noise. These spots need to be cleaned. Another problem that arises is that in cartoon images that previously had a dividing line, it would produce a firm double line because of the difference in value between the dividing line and its part. The process for this type of image cannot be done only by doing binary thresholding, because there are dark parts of the image that must still be made white so that they can be colored. [10]

The provision of pictures for coloring media according to the skill level of students is very necessary. The teacher should look for line-art drawings for the coloring activity. The line-art images available for free do not always meet these needs. The efforts of the teachers to make their own line-art drawings from scratch are also rather difficult, because not all teachers also have good drawing competence.

Images from cartoons, animated images or photos can be easily obtained and are widely available for free. The image can be processed into a line-art image so that the teacher can freely choose the image that will be used as a line-art image. The obstacle faced is that not all teachers are also skilled at using image processing applications. To solve this problem, we need an application that can automatically process an input image into a line-art image. The application only needs to enter an image into the application to produce the desired line-art image, the line-art image can be distributed to students to be colored.

Based on the background of the problem that has been described previously, the formulation of the problem is how to arrange a process that can produce an outline of an object in the image so that it can produce line-art with an unbroken dividing line of object parts and a single dividing line.

## 2. Methods

### A. High Pass Filtering Theory

Linear spatial filtering is filtering that is performed by linear operations in the form of convolution or correlation between the local area of a bitmap cell and the kernel. In the filtering

\*Corresponding author: tmulyana@bundamulia.ac.id

process, a filter  $g(x,y)$  is needed in the form of a matrix measuring  $m \times n$ , in general the matrix used is a  $3 \times 3$  matrix in which each cell contains a weight value. This matrix is referred to as a filter, mask, or kernel, some call it a window. Every point  $(x,y)$  in the  $f(x,y)$  image is filtered with a  $g(x,y)$  filter to produce  $h(x,y)$ [11].

Kernel high pass filtering is a kernel for filtering values with high intensity and reducing them with low intensity values, so as to produce an image sharpening effect.[7] Kernel is designed to increase the brightness of a pixel based on the share of neighboring values. [12].

Edge detection has direction and some has no direction. The direction referred to here is the existence of a higher or lower value on one side compared to the other side [8], [9], [13]. The high pass filtering kernel used in this edge detection is a high pass filtering kernel which has no direction, so this kernel can only be operated with correlation. [14]

Kernel high pass filtering with a positive kernel center is intended to hold high-intensity values that are bordered by low-intensity values, so as to detect edges of images with bright objects. [15], [16]

Kernel high pass filtering with a coefficient of zero has a very large difference in the center cell with its neighbors. This kernel has a central cell that has the highest value, while its neighboring cells have a negative value. [7]

High pass filtering kernels with a coefficient of zero can also have negative core values. This kernel has very low cell center values with positive neighboring values. The difference between the center of the kernel and its neighbors is also large like a kernel with positive cores. This kernel will filter pixels with low values compared to neighboring pixels. [7]

The filtering process is carried out by performing a convolution operation. which is the recursive sum of linear functions. Convolution for the purposes of optical processes, is useful in many ways for the purposes of image sharpening, as well as for the process of smoothing image gradations. For programming purposes, this process is formulated according to formula (1) [12],[17].

$$h(x,y) = \sum_{k=1}^M \sum_{l=1}^N f(k,l) \cdot g(x - (k - m), y - (l - n)) \quad (1)$$

#### Information:

- $x, y, k, l$ : independent variables that have discrete values, where  $x$  and  $y$  are the coordinates of the pixel being processed,  $k$  and  $l$  are the coordinates of the pixels in a local area that affect the result  $h(x,y)$
- $h(x,y)$  : result at coordinate  $x,y$
- $f(x,y)$  : function  $f$  that processes pixels  $x,y$  and its neighbors
- $g(x,y)$  : filter function to process  $x,y$  . pixels
- $m, n$ : neighboring point boundaries that affect the point being processed

High pass filtering which has a total coefficient of zero with a positive core in a binary image with a black object and a white background will produce an edge on the outside of the object.

The same effect occurs in filters with negative cores, in binary images with white objects on a black background.

High pass filtering which has a total coefficient of zero with a positive core in a binary image with a white object and a black background will produce an edge on the inside of the object. The same effect occurs in filters with negative cores, in binary images with black objects on a white background.

All edge pixels resulting from high pass filtering that have a total coefficient of zero with negative cores and positive cores, both in binary images with black objects on a white background and binary images with white objects on a black background, will be white, with all pixels other than edges on objects or black backgrounds.[7], [18], [19]

The line-art creation process is carried out using sequential processes like grayscaling, contrast stretching, edge detection, binarization with thresholding, noise cleaning with thresholding, lastly a closing operation is performed to close broken lines or to merge two separate lines. due to edge detection in bold lines.

#### B. Grayscale Process and Contrast Stretch

Grayscale process is carried out as the initial stage of processing, so that the image can be easily processed with edge detection.[19], [20] The next process is the Contrast Stretching Process, this process is needed to emphasize the boundaries of parts of the image. [21] Contrast stretching is carried out according to formula (2). The contrast gain value  $G$  is taken as 2. While the contrast stretch center value is taken from the contrast stretch center trackbar input An example of the contrast stretching result and its effect on enhancing edge detection can be seen in Fig. 1. [22].

$$H(x,y) = (C(x,y) - P) * G + P \quad (2)$$

#### Information:

- $x$ : bitmap column address
- $y$ : bitmap line address
- $H(x,y)$  result pixels at address  $x,y$
- $C(x,y)$  original pixel at address  $x,y$
- $P$ : center of contrast stretch
- $G$ : Contrast Gain

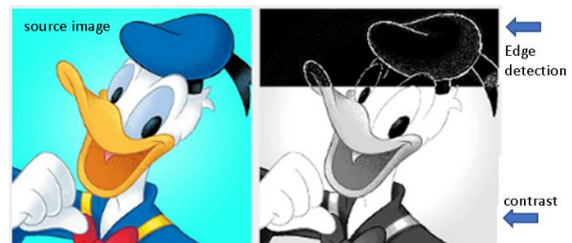


Fig. 1. Contrast and edge detection

#### C. Edge Detection Process

For the edge detection process, the edge detection process for black pixel objects is chosen to produce edges in the object. The choice of this process is to ensure the distance between the edges of a thick line is not too far, also so that the thin line does

not produce double lines. To realize this process, a filter from high pass filtering is used with a total coefficient of 0, where the center of the filter matrix is less than 0 (negative) so that the neighboring cells will be positive.

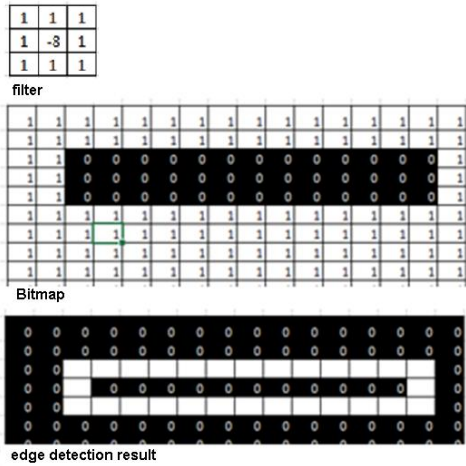


Fig. 2. Outline result from edge detection process

In theory, high pass filtering with a total coefficient of 0 with a negative matrix center will detect pixels with low values adjacent to pixels with high values. This matrix can be seen in Fig. 2. The center of the matrix is worth -8 while the neighboring cells each have a value of 1.

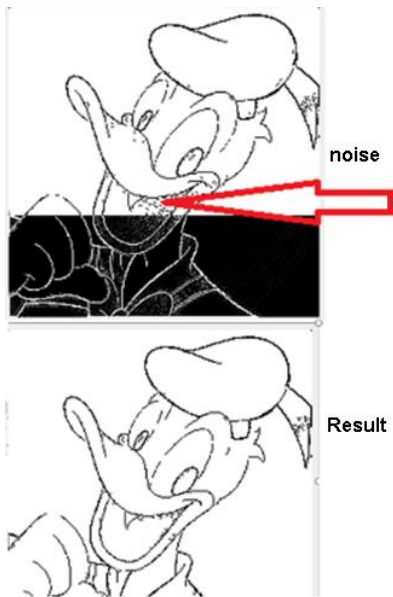


Fig. 3. Noise reducing

As an illustration, we can see in Fig. 2, an object with a pixel value of 0 is shown and a background with a pixel value of 1 in the image on the left. The edge detection results produce the edges of the object that borders the background in the image on the right. The resulting edges are contained within the object. Because the object is 0.

D. Noise Reduction

The spot cleaning process is done by looking for where the

spots are. This is done on a line-by-line scan basis of the image pixels. The Process is shown in Fig. 3. Once found, one of the pixels that contains a black value, then from the position of that pixel the number of black pixels connected to that pixel is calculated [23], [24]. If the resulting value is greater than the noise limit value, then the pixel with the connected pixels is considered noise [25], but if not, then it is considered an art-line. The bitmap is marked as processed so that it is not recalculated again. This process relies on Backtracking algorithms which can be seen in the source code of the program in Fig. 4.

```

procedure searchNoise(x, y : integer);
.
.
.
880   if (x >= 0)
.       and (x<=lebar-1)
.       and (y>=0)
.       and (y<=tinggi-1) then
.   begin
.       if (tempBiner[x+1,y] = 0) then
.           begin
.               searchNoise(x+1, y);
.           end;
.       if (tempBiner[x,y+1] = 0) then
.           begin
.               searchNoise(x, y+1);
.           end;
.       if (tempBiner[x-1,y] = 0) then
.           begin
.               searchNoise(x-1, y);
.           end;
.       if (tempBiner[x,y-1] = 0) then
.           begin
.               searchNoise(x, y-1);
.           end;
.       //diagonal
.       if (tempBiner[x+1,y+1] = 0) then
.           begin
.               searchNoise(x+1, y+1);
.           end;
.       if (tempBiner[x-1,y+1] = 0) then
.           begin
.               searchNoise(x-1, y+1);
.           end;
.       if (tempBiner[x-1,y-1] = 0) then
.           begin
.               searchNoise(x-1, y-1);
.           end;
.       if (tempBiner[x+1,y-1] = 0) then
.           begin
.               searchNoise(x+1, y-1);
.           end;
.       end;
.   end;
. end;
    
```

Fig. 4. Noise searching

As a basic rule, the cell will be given a value of 2 as a sign that the cell has been searched. While doing the search, the population value will be added every time a function call occurs. As a condition to perform a recursive search, the value of the cell to be searched must be 0. Recursion is carried out if the value of the cell being searched is 0, if it is not 0, then the recursive process will not be called. Giving a value of 2 to the cell to be searched will result in the cell that has been searched no longer being searched recursively.

Recursive rules are arranged based on the 8-way neighbors of the cell being searched. backtracking occurs when all

recursive tracing rules have been passed. At that time the process will be popped from memory to backtrack to the recursive process that called it. The object being searched will be marked as noise if the total population of black cells representing the object is less than the threshold value.

```

procedure removeNoise(x, y : integer);
Image3.Canvas.Pixels[x,y] := clWhite;
Sketch[x,y] := False;
923 if sketch[x+1,y] then
begin
925   removeNoise(x+1, y);
end;
if (tempBiner[x,y+1] = 0) then
begin
   removeNoise(x, y+1);
930 end;
if (tempBiner[x-1,y] = 0) then
begin
   removeNoise(x-1, y);
end;
935 if (tempBiner[x,y-1] = 0) then
begin
   removeNoise(x, y-1);
end;
//diagonal
940 if sketch[x+1,y+1] then
begin
   removeNoise(x+1, y+1);
end;
if (tempBiner[x-1,y+1] = 0) then
945 begin
   removeNoise(x-1, y+1);
end;
if (tempBiner[x-1,y-1] = 0) then
begin
950   removeNoise(x-1, y-1);
end;
if (tempBiner[x+1,y-1] = 0) then
begin
955   removeNoise(x+1, y-1);
end;
end;

```

Fig. 5. Remove noise

If it is marked as noise then noise cleaning is done. This cleaning process also uses the backtracking algorithm shown in Fig. 5.

The basic rule of this purge recursive process is to assign false or nonobject values to the cells being searched for. So that objects that are recognized as noise will be removed from the image.

Just like the noise tracing process, this noise cleaning process will also recursively in 8 directions of neighboring cells. Recursion is carried out in the direction of the neighbor by checking the position of the cell according to the direction of the neighbor. Then, if a cell is found to be a noise cell, a search will be carried out while cleaning the cells that are considered noise. The arrangement of recursive rules according to the neighboring direction will result in a backtracking process when all the rules of the recursive direction are passed.

### E. Closing Morphology

In making line-art based on cartoon images that have outline lines on the image, the resulting outline will have double lines. This is caused by an edge detection process that will detect both

edges of the outline of the original image. This double outline can be seen in Fig. 6 in the middle image. the image on the left shows the original image.

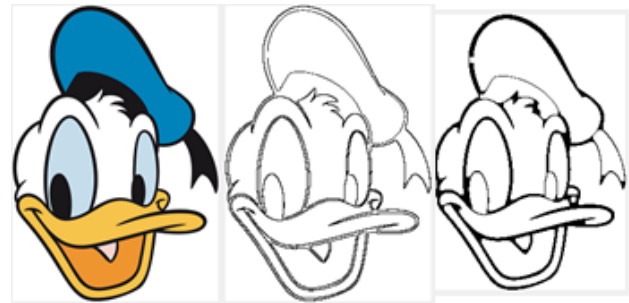


Fig. 6. Closing the double outline

The result of making line-art with the closing process can be seen in the image on the right. Closing process is a sequence of n times dilation process then followed by n times erosion process to restore the size back to its original size or at least close to its original size [13], [26], [27], [28]. In the picture, you can see several outline lines as thick as the outline of the original image. This is because the position of the outline line resulting from the edge detection is right on the edge of the original outline line. So that after the closing process is carried out, the two lines unite during the dilation process, so when the erosion process is carried out, the gap between the two lines is closed and does not erode again, while on the edges outside the gap the two edge detection lines remain eroded, so the remaining lines are as thick as the line in the original image. Thin outline sections are line sections that do not produce double lines after the edge detection process.

## 3. Result and Discussion

The test in this study is intended to test the success of the process of making Art-Line images. Success is judged by how many points of line leakage should not occur and cartoon images that already have a boundary line can still have a single boundary line.

Table 1  
Testing Result

Image Type	Sample Number	Outline Broken		
		Yes	No	Success
Without Outline	100	22	78	78%
With Outline	100	0	100	100%

The testing procedure is carried out based on two images, namely cartoon images without borders and cartoon images with border lines. Line leakage testing was carried out on both types of images. Testing the success of a single line is carried out only on cartoon images that have a dividing line between their parts.

The results of the test recap can be seen in Table 1. The test was carried out on 100 cartoon images without borders and 100 cartoon images with thick borders. The recap of the leak test results on cartoon images without borders resulted in 78% of images with unbroken lines. 22% experienced leaks (broken line), with the part that leaked at least 1 and at most 7 parts. This



recap can be seen in Table 2. The results of this recap will be used as an evaluation for the next research.

Table 2  
Image number that has broken line

Broken Line	Image Number
0	78
1	6
2	0
3	5
4	7
5	2
6	0
7	2
Total	100
Broken	22%

Testing on cartoons that have thick dividing lines, 100% no leakage. Tests on cartoon images that already have dividing lines all work 100% to make the art-line into a single line.

The test was carried out with a center of contrast value of 127 and a contrast gain of 2, a binary threshold value of 127, and a value for the number of noise pixels 5. The test was carried out with the morphological value of erosion and dilation for closing adjusted to the thickness of the boundary line. This value ranges from 3 to 5. As a limitation of the problem, this test is not tested for the influence of these values, because it will be a separate study.

Based on temporary observations of the image source, the leakage in the image without borders is all due to the very slight difference in the grayscale image value due to the color gradation, even though the counter stretch has not succeeded in separating the colors. This argument requires further research.

#### 4. Conclusion

The formation of line-art from the image can be done by composing the gray scaling process to facilitate edge detection, stretching the contrast to emphasize the edge detection results, edge detection to find the boundaries of the color area, inverse the image to produce line-art with a black outline with a white background, tracing as well as noise cleaning with backtracking and closing processes to maintain a single outline in images that have color borders.

The cartoon image without borders has a success of 78%, while the cartoon image with a border outline produces 100% success. The failure is caused because the grayscale values generated from the gray scaling process generally produce grayscale values that are not much different in adjacent parts.

#### References

- [1] W. Warnida, "Upaya Meningkatkan Kemampuan Motorik Halus Melalui Kegiatan Mewarnai di Kelompok B1 TK Berkah Kota Jambi Tahun 2016/2017," *J. Ilm. Dikdaya*, vol. 9, no. 1, p. 132, Apr. 2019.
- [2] I. Kurnia, "Pengaruh Kegiatan Mewarnai Gambar terhadap Kemampuan Motorik Halus Anak Kelompok B di Pendidikan Anak Usia Dini Bukit Selanjut Kecamatan Kelayang Kabupaten Indragiri Hulu," *Kindergarten J. Islam. Early Child. Educ.*, vol. 2, no. 1, p. 67, Feb. 2020.
- [3] L. D. Larasati, N. Kurniah, and D. D., "Peningkatan Kreativitas Dalam Kegiatan Mewarnai Dengan Menggunakan Metode Demonstrasi," *J. Ilm. Potensia*, vol. 1, no. 2, pp. 62–66, 2016.
- [4] G. Papari and N. Petkov, "Edge and line oriented contour detection: State of the art," *Image Vis. Comput.*, vol. 29, no. 2–3, pp. 79–103, Feb. 2011.
- [5] Y. Kim, J. Yu, X. Yu, and S. Lee, "Line-art illustration of dynamic and specular surfaces," in *ACM SIGGRAPH Asia 2008 papers on - SIGGRAPH Asia '08*, 2008, p. 1.
- [6] A. Hertzmann, "Why Do Line Drawings Work? A Realism Hypothesis," *Perception*, vol. 49, no. 4, pp. 439–451, Apr. 2020.
- [7] T. M. S. Mulyana, "Efek High Pass Filtering Dengan Koefisien Nol Pada Citra Biner," *J. Muara Sains, Teknol. Kedokt. dan Ilmu Kesehat.*, vol. 1, no. 1, May 2017.
- [8] V. Kurlin and G. Muszynski, "A persistence-based approach to automatic detection of line segments in images," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11382 LNCS, no. August, pp. 137–150, 2019.
- [9] D. Chudasama, T. Patel, S. Joshi, and G. I. Prajapati, "Image Segmentation using Morphological Operations," *Int. J. Comput. Appl.*, vol. 117, no. 18, pp. 16–19, 2015.
- [10] M. M. Sebatubun and E. H. Pujiarini, "Pengenalan Varietas Kopi Arabika Berdasarkan Fitur Bentuk," *JIKO (Jurnal Inform. dan Komputer) STMIK AKAKOM*, vol. 3, no. 2, pp. 60–69, 2018.
- [11] P. Tarabek, "Morphology Image Pre-Processing For Thinning Algorithms," *J. Information, Control Manag. Syst.*, vol. 5, no. 1, pp. 131–138, 2007.
- [12] L. kabbai, A. Sghaier, A. Douik, and M. Machhout, "FPGA implementation of filtered image using 2D Gaussian filter," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 7, pp. 514–520, 2016.
- [13] Y. Chugh, R. Gupta, and R. Kaushik, "Image Enhancement Using Morphological Operators," *Int. J. Eng. Technol.*, vol. 3, no. May, pp. 2349–4476, 2015.
- [14] C. Wu, Q. Wu, C. Yuan, P. Li, Y. Zhang, and Y. Xiao, "Multimeter digital recognition based on feature coding detection," in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Oct. 2017, pp. 1–6.
- [15] R. Sejal and P. Mitul, "Removal of the Fog from the Image Using Filters and Colour Model," *Int. J. Eng. Res. Technol.*, vol. 3, no. 1, pp. 553–557, 2014.
- [16] K. Abhinav, J. Singh Chauhan, and D. Sarkar, "Image Segmentation of Multi-shaped Overlapping Objects," in *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Nov. 2018, vol. 4, pp. 410–418.
- [17] P. Rosyani, S. Saprudin, and R. Amalia, "Klasifikasi Citra Menggunakan Metode Random Forest dan Sequential Minimal Optimization (SMO)," *J. Sist. dan Teknol. Inf.*, vol. 9, no. 2, p. 132, Apr. 2021.
- [18] T. M. S. Mulyana, "Reduce noise in the binary image using non linear spatial filtering of mode," in *2016 International Conference on Information & Communication Technology and Systems (ICTS)*, 2016, pp. 135–139.
- [19] F. F. C. Ayu Novitasari, Endina Putri Purwandari, "Tidentifikasi Citra Daun Tanaman Jeruk Dengan Local Binary Pattern Dan Moment Invariant," *JIKO (Jurnal Inform. dan Komputer) STMIK AKAKOM*, vol. 3, no. 2, pp. 76–83, 2018.
- [20] L. Novamizanti, H. Gymnovriza, and E. Susatio, "Pengenalan Wajah Individu Berbasis 3D Biometrik," *JIKO (Jurnal Inform. dan Komputer)*, vol. 6, no. 1, p. 41, 2022.
- [21] M. Qomaruddin, D. Riana, and A. Anton, "Segmentasi K-Means Citra Daun Tin Dengan Klasifikasi Ciri Gray Level Co Occurance Matrix," *J. Sist. dan Teknol. Inf.*, vol. 9, no. 2, p. 223, 2021.
- [22] R. Firoz, S. Ali, M. N. U. Khan, and K. Hossain, "Medical Image Enhancement Using Morphological Transformation," *J. Data Anal. Inf. Process.*, vol. 4, no. February, pp. 1–12, 2016.
- [23] P. Rosyani, R. Amalia, and I. H. Iksari, "Deteksi Objek dengan Model Warna Ycber dan Similarity Distance," *J. Sist. dan Teknol. Inf.*, vol. 9, no. 2, p. 98, Apr. 2021.
- [24] D. Sur, S. Rudra, S. Mitra, and S. Mitra, "A Case Study on Effects of Low Pass Filtering over Facial Image Processing and Proper Edge," *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 5, no. 7, pp. 13394–13402, 2016.
- [25] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern Recognit.*, vol. 33, no. 2, pp. 225–236, 2000.
- [26] S. Singh and S. K. Grewal, "Role of Mathematical Morphology in Digital Image Processing: A Review," vol. 2, no. 4, pp. 3–5, 2014.
- [27] A. . Raid, W. . Khedr, M. . El-dosuky, and M. Aoud, "Image Restoration Based on Morphological Operations," *Int. J. Comput. Sci. Eng. Inf. Technol.*, vol. 4, no. 3, pp. 9–21, Jun. 2014.
- [28] K. Sreedhar, "Enhancement of Images Using Morphological Transformations," *Int. J. Comput. Sci. Inf. Technol.*, vol. 4, no. 1, pp. 33–50, Feb. 2012.