Morphology on Label Image Using Multilayer Approach

MSc Thesis

Swaiz Ahmed

Master's Thesis

UNIVERSITY OF EASTERN FINLAND

School of Computing
Computer Science

April 2025

UNIVERSITY OF EASTERN FINLAND, Faculty of Science and Forestry, Joensuu
School of Computing
Computer Science

Swaiz Ahmed: Morphology on label images using multilayer approach
Master's Thesis, 49p.

Supervisor of the Master's Thesis: Professor Pasi Fränti

April 2025

Abstract: In image processing, morphology relates to a collection of processes that change images based on their shapes. A structuring element is exercised to an input image, resulting in an output image of a similar size. Label morphology using a multilayer approach deals with three common steps. The input image is divided into different layers, and each layer has unique features of the Image, such as edges or textures. Every individual layer is then processed using morphological operations like opening, closing, dilation, and erosion. The operation is applied to all layers, and the choice depends on the application. These processed layers are merged to create an output image identical to the input image. The final Image is more precise and detailed, making this a helpful process for improving the quality of label images. The technique is useful to remove noise from the images.

Keywords: Mathematical Morphology, Structuring Element, Label Images, Multilayer Approach, Map Imaging

Acknowledgments

First and foremost, I would like to extend my deepest gratitude to the University of Eastern Finland and its faculty. I am particularly thankful to my supervisor, Professor Pasi Fränti, for his invaluable guidance and motivation throughout my thesis journey. It has been an immense pleasure to have him as an instructor. I appreciate his support and direction for my thesis, other courses, and career-related matters. I also want to thank my teacher sincerely, Adil Waheed, and my friends and family for their unwavering support and encouragement, which was crucial in completing my studies. This achievement would not have been possible without them. Lastly, I am profoundly grateful to my parents for allowing me to study in Finland. May God bless them with success and honor in their lives.

List of abbreviations

MO	Morphological Operations
RGB	Red, Green, and Blue
SE	Structuring Element
JBIG	Joint Bi-level Image Experts Group

Contents

List of Figures

List of Tables

1 Introduction

Morphology is a set of procedures in image processing that deal with the shape and structure of objects in an image. Image analysis often employs morphological processes such as picture augmentation and object extraction. Image labeling entails assigning a distinct identification (label) to each connected part or region. Connected components are collections of pixels linked together based on specific criteria, such as pixel intensity or color.

A multilayer approach in image processing refers to the use of multiple layers or levels of processing to achieve a specific goal. In the context of morphology and labeling images, a multilayer approach may involve applying morphological operations at various stages or scales to capture multiple levels of detail in the Image. A multilayer approach allows you to apply MO at different scales. This helps capture both small and large structures in the Image. Everyday morphological operations include erosion, dilation, opening, and closing. These operations can be applied using structuring elements that define the neighborhood relationships between pixels.

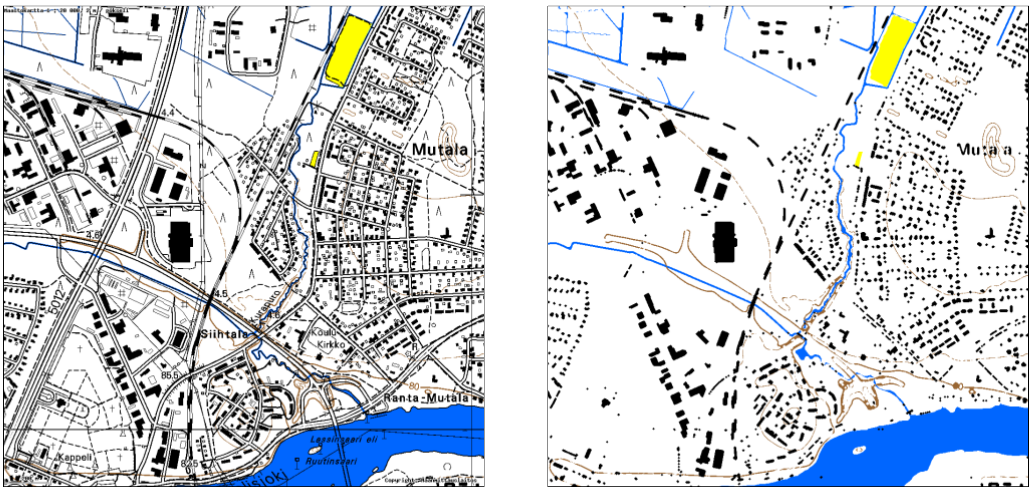

Figure 1: *Morphological Filtering Image*

1.1 RGB Images

An RGB image consists of M*N pixels, where M stands for width, and N stands for height. When displayed on a screen, each pixel corresponds to a specific color. When an RGB image is broken down, it is divided into three layers. These layers have values associated with red, green, and blue. In an 8-bit representation, each channel's values typically range from 0 to 255, with 0 being the absence of color and 255 being the maximum intensity. RGB images are commonly used to depict photographs, digital images, and other visual content that requires color information. The figure below illustrates an RGB image.

Figure 2: *RGB Images of Parrots*

1.2 Map Images

In image processing, map images refer to grayscale or single-channel images. In an 8-bit grayscale image, each pixel is represented by a single number showing the intensity of light or darkness, which commonly ranges from 0 (black) to 255 (white). In contrast to RGB images, which have three color channels, map pictures only have one channel, representing the Image's brightness or intensity. The figure below shows a map image.

Figure 3: *Greyscale Image of Ships*

The main difference lies in the way color information is represented. RGB images use three color channels to describe color, while map images, often referring to grayscale images, use a single channel to represent intensity or brightness. The selection between RGB and grayscale depends on the specific requirements of the image processing task and whether color information is essential for the analysis.

The table below shows some statistical differences between RGB and grayscale.

Table 1: *Statistical Difference between RGB and Map Images*

	RGB	**Map Image**
Color Channels	3 (Red, Green, Blue)	1
Channel Bit Depth	8 bits per channel (0 to 255)	8 bits (0 to 255)
Total Bit Depth	24 bits per pixel (8 bits per channel * 3 channels)	8 bits per pixel
Total Colors	256^3 = 16,777,216 possible colors	256 (from black to white)
Common File Formats	JPEG, PNG, TIFF	BMP, PNG, TIFF

2 Morphology on label images

A structuring element is applied to an input image, producing an output image of the same size. Morphology is a powerful image processing technique used to manipulate the shapes and sizes of label images. Morphological operations refine and improve the quality of label images. Four primary morphological operations include dilation, erosion, opening, and closing.

2.1 Mathematical Morphology

Mathematical morphology is an effective image-processing technique based on set theory, algebra, and mathematical operations concepts. It controls the shape and structure of objects in digital photographs using a powerful and versatile framework. Fundamental operations, erosion, and dilation are localized within the core of mathematical morphology. Operations are used to adjust the added properties owned by objects in the interior of a singular via the systematic method of applying a single preset structuring element, which works as a singular probe rather than a filter.

Combining these two basic operations can create more complex morphological transformations, such as opening (erosion followed by dilation) and closing (dilation followed by erosion). These operations benefit noise removal, edge enhancement, and object segmentation. Beyond these fundamental operations, mathematical morphology also includes a range of advanced techniques, such as top-hat and bottom-hat transformations, watershed segmentation, and morphological filtering. These techniques can be tailored to specific applications and image processing requirements, making mathematical morphology a versatile and powerful tool in digital image analysis and processing.

Overall, mathematical morphology provides a robust and flexible framework for manipulating the shape and structure of objects within digital images, with a wide range of applications in fields such as computer vision and image processing.

2.2 Label images

Label images in morphology refer to digital images where each distinct object or region of interest is assigned a unique label or identifier. These labels are often used to segment and analyze the different components within an image, enabling quantitative analysis of morphological features, such as size, shape, and spatial distribution. Label images are specialized annotations that assign a unique label to each pixel in an image. Think of it as a detailed map where every region or object within the Image has its distinct identifier. This mapping is crucial for various image-processing tasks, including image morphing.

In the context of image morphing, label images serve as blueprints. They guide the transformation process, ensuring that specific areas within the Image are manipulated accurately. By providing a clear outline of the boundaries and characteristics of different objects or regions, label images help keep the integrity of the visual transformation. Our research uses label images to create a new image that is morphed, cleared, and enhanced. For this purpose, we used dilation and erosion to remove that label from geographically and map images; thus, the output makes the Image more advanced and understandable.

Label images in morphology are a crucial tool for researchers across various disciplines. By assigning unique identifiers to different regions within an image, label images ease detailed analysis and quantification of structures. Researchers can use label images with advanced tools and techniques to gain deeper study insights.

- CellProfiler for high-throughput cell image analysis.
- Fiji's MorphoLibJ for morphological analysis.
- Machine learning and deep learning frameworks for automated segmentation and classification.

By applying these morphological techniques, label images can be transformed and enhanced, making them more legible, visually appealing, and accessible to process and analyze. This can be particularly useful in product packaging, industrial automation, and document digitization applications.

Following are the examples of label images

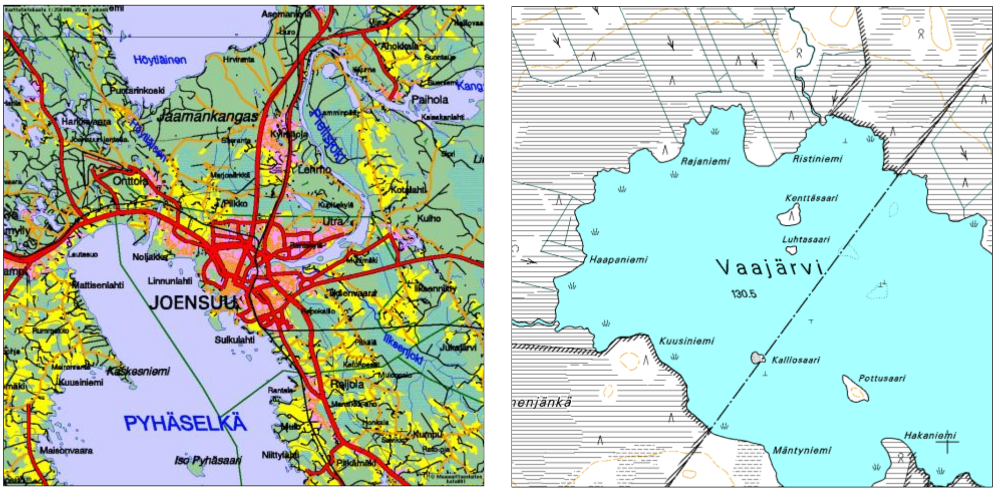

Figure 4: *Label Images of Maps of different cities of Finland*

2.3 Related Work

Morphological operations are essential techniques in image processing used to analyze and manipulate the structure of objects within an image. Below, we outline some significant related works that use morphological operations such as closing, dilation, and opening for label image analysis.

Jean Serra's (1982) seminal work in morphological image processing laid the foundation for many modern applications. His book, "Image Analysis and Mathematical Morphology," details various morphological operations, including dilation, erosion, opening, and closing. The techniques described have been widely applied to preprocessing steps in label image analysis, particularly for enhancing image quality and removing noise.

Rui, Huang, and Chang's (1999) paper on shape and texture-based image retrieval explores the applications of morphological operations for extracting features from images. Their research shows how opening and closing operations can separate text from the background in label images, making them particularly useful for document and label image analysis.

Liu and colleagues (2006) study on license plate recognition incorporates morphological operations to preprocess and segment characters. Dilation is used to merge broken character parts, while closing helps to fill small holes and gaps. These techniques apply to label images where text clarity is crucial for the next OCR (Optical Character Recognition) tasks.

Haralick(1987) et al.'s "Image Analysis Using Mathematical Morphology" work focuses on using morphological operations for noise reduction in binary images. Opening and closing operations are highlighted as practical means of removing minor artifacts and smoothing boundaries, which are vital for ensuring the accuracy of label segmentation and recognition.

The DIBCO (2009) contests have consistently highlighted ultramodern techniques for document image binarization, many of which employ morphological operations. Gatos and his team explain how dilation and closing can be used to enhance the legibility of text in scanned documents, which is highly relevant for the analysis of label images.

Smith and Kanade's (2005) research on automatic text extraction from label images combines morphological operations to isolate text regions. Dilation and opening are employed to connect fragmented parts of characters and remove small noise elements, respectively. This approach improves the efficiency and accuracy of text extraction processes in label image analysis.

This study [1] presented an optimized model for enhancing brain MRI images to improve tumor detection using morphological filters with a disk-shaped structuring element. The research [2] introduced an unsupervised method using figure-ground maps and SimCLR to generate and analyze morphology features of urban form types. This paper [3] presented an automated approach using machine learning and computer vision to efficiently characterize grain morphology in TEM images of nanocrystalline UO2 during in-situ ion irradiation, achieving results comparable to human experts with reduced analysis time. The monograph [4] proposed a mathematical morphology-based algorithm for preprocessing and feature extraction in side-scan sonar images, achieving efficient and correct underwater target detection and recognition. This report [5] compared five ultramodern superpixel algorithms and introduced the Simple Linear Iterative Clustering (SLIC) algorithm, which efficiently generates superpixels that adhere to image boundaries, improve segmentation performance, and are faster and more memory efficient.

The paper [6] analyzed the random swap algorithm for clustering, showing its efficiency and expected time complexity, highlighting its linear dependency on data size, quadratic dependency on cluster number, and inverse dependency on neighborhood size, with experiments confirming its superiority over k-means. This article [7] presented a simple and efficient multiresolution approach to grayscale and rotation invariant texture classification using local binary patterns and nonparametric discrimination, showing robustness to grayscale variations and computational simplicity. The research [8] presented a technique for restoring semantic layers from color-separated map images, enhancing compression performance and visual quality for various lossless compression algorithms. The study [9] explored adapting binary layer context-tree-based compression techniques, initially designed for color map images, to photographic and palette images by evaluating various bit-plane separation methods and extending the approach to multilayer contexts for improved performance. The report [10] assessed the NIH ImageJ Java-based framework for implementing and experimenting with morphological operations in image processing, highlighting its ease of use and computational efficiency.

The article [11] explained a method for compressing color map images using interlayer correlations through context tree modeling and arithmetic coding, achieving significant improvements over JBIG and single-layer context tree modeling. The paper [12] explored enhancing binary image compression by combining hierarchical block coding with predictive coding and bit row reordering to increase the number of white pixels and all-white blocks, using Huffman coding for local dependencies. A storage system [13] for digital map images was introduced, using binary layer division, context-based statistical modeling, and arithmetic coding for compact storage, supporting partial decompression and smooth scale transitions for real-time applications on low-resource portable devices. The research [14] presented a method for improving context templates in context-based image compression, tailored to multi-part map images and using the JBIG2 standard for efficient layer compression. This study [15] evaluated the most suitable compression methods for storing digital map images, comparing binary image standards (G4, JBIG2) and limited-color computer graphics standards (GIF, PNG) through experiments on both clean and noisy map images.

The monograph [16] proposed a method for color quantization of scanned, lossy compressed map images to reduce their color count despite the unknown number of original colors. A novel [17] multilayer image filtering approach for raster map images transformed filtering into the binary domain, significantly improving efficiency and preserving crucial details. The analysis [18] presented an extended context-based filtering method for denoising raster map images corrupted by additive Gaussian noise, efficiently preserving specific local contexts using a probability-based fusion procedure. The proposed context tree filtering method [19] for indexed-color raster map images significantly reduces impulsive and content-dependent noise while preserving spatial structures better than existing filters, achieving up to 30% improvement in noise reduction. The paper [20] evaluated the performance of Multilayer Context Tree (MCT) modeling for lossless compression of grayscale images using three-bit-plane separation methods and compared it with existing algorithms.

Overall, Morphological operations such as closing, dilation, and opening have been widely studied and applied in various fields of image processing, particularly for enhancing and analyzing label images. These operations are crucial for noise reduction, text extraction, and improving image quality, making them valuable tools for anyone involved in label image analysis.

3 Multilayer Approach

Morphology analysis is a powerful technique used in image processing to extract meaningful structures and details. This guide will use a multilayer approach to apply morphology analysis to map images. Analyzing map label images through a multilayer morphological approach can reveal many features often hidden in a single-layer analysis. The process involves decomposing the Image into separate layers based on color, applying various morphological techniques, and then recombining these layers to produce a refined output. The Image below describes dividing the layers according to their color and then applying morphological operations. Furthermore, the next step involves recombing the process image to create a new output image that is identical to the original Image.

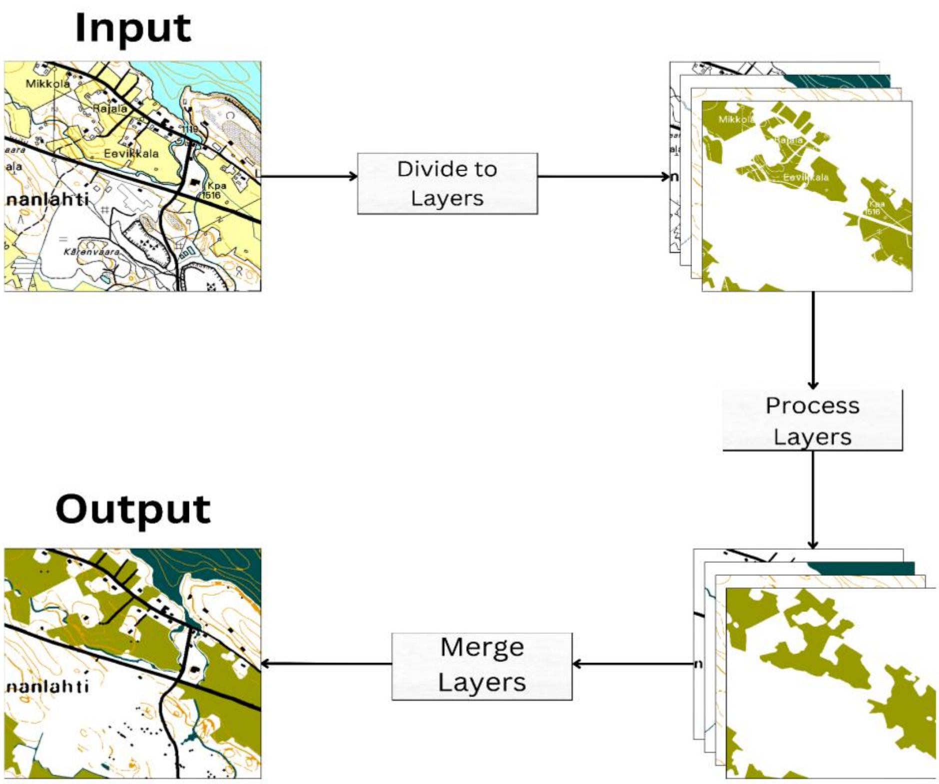

Figure 5: *Steps involved in Morphological Filtering of Map Image*

The above Image has been created through MATLAB, and the Image has been input using the imread function, e.g.,

Img=imread('img_name');

The following steps are decomposing, filtering, and then recombining the layers.

3.1 Decomposition

The first step in the multilayer approach is dividing an input image into different layers using their specifications. In our case, we have divided the layer using color values. Each color has a specific range in the RGB spectrum. Each layer of distinct colors in the original Image is separated by giving the color range of that particular color and performing color thresholding to extract the color layer using that range. The example code for removing one of the layers from the original Image is below. Sometimes, the original color in the Image is too light to be adequately distinguished. Hence, the function below also resolves this issue by applying a technique to darken the color of this layer. This is done by converting the layer into HSV and changing the hue and saturation values.

```
function [BW,maskedRGBImage] = yellowMask(RGB)
   % Convert RGB image to chosen color space
   I = RGB;
   % Define thresholds for channel 1 based on histogram settings
   channel1Min = 255.000;
   channel1Max = 255.000;
   % Define thresholds for channel 2 based on histogram settings
   channel2Min = 255.000;
   channel2Max = 255.000;
   % Define thresholds for channel 3 based on histogram settings
   channel3Min = 0.000;
```

```
channel3Max = 169.000;
% Create mask based on chosen histogram thresholds
sliderBW = (I(:,:,1) >= channel1Min ) & (I(:,:,1) <= channel1Max) & ...
 (I(,2) >= channel2Min ) & (I(:,:,2) <= channel2Max) & ...
 (I(:,:,3) >= channel3Min ) & (I(:,:,3) <= channel3Max);
%BW = morphMask(sliderBW);
BW = sliderBW;
% Initialize output masked Image based on the input image.
maskedRGBImage = RGB;
% Set background pixels where BW is false to zero.
maskedRGBImage(repeat(~BW,[1 1 3])) = 0;
%maskedRGBImage = morphMask(maskedRGBImage);
% Make the identified yellows darker and more saturated
darkenFactor = 1; % Reduce brightness by 20%
saturationIncrease = 1; % Increase saturation by 20;
% Convert RGB to HSV
hsvImage = rgb2hsv(maskedRGBImage);
hueChannel = hsvImage(:,:,1);
saturationChannel = hsvImage(:,:,2);
valueChannel = hsvImage(:,:,3);
% Adjust the value and saturation channels
valueChannel = valueChannel * darkenFactor;
saturationChannel = saturationChannel * saturationIncrease;
% Ensure the intensity does not exceed 1
valueChannel(valueChannel > 1) = 1;
saturationChannel(saturationChannel > 1) = 1;
% Combine the adjusted HSV channels back
hsvImage(:,:,1) = hueChannel;
hsvImage(:,:,2) = saturationChannel;
hsvImage(:,:,3) = valueChannel;
% Convert HSV back to RGB
maskedRGBImage = hsv2rgb(hsvImage);
% Multiply by 255 to convert to uint8
maskedRGBImage = uint8(maskedRGBImage * 255);
```

end

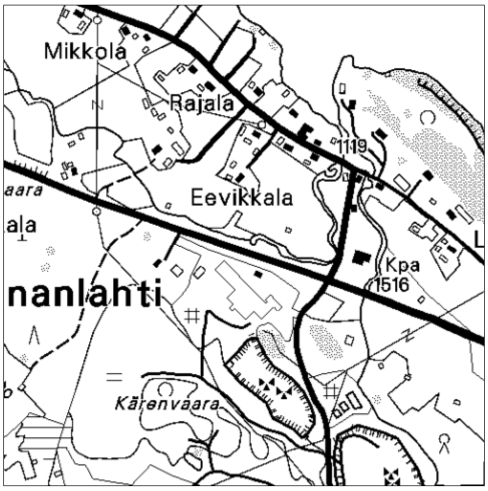

Figure 6: *Black Layer Separated*

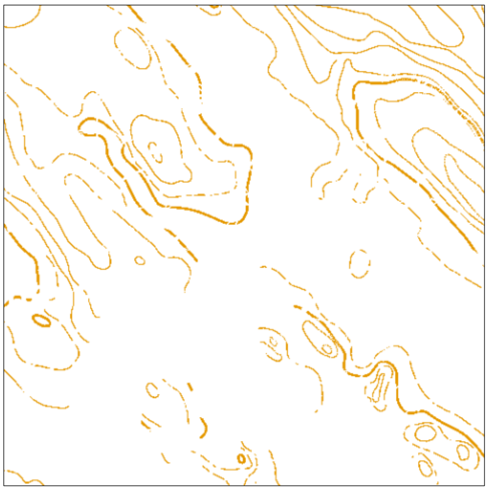

Figure 7: *Brown Layer Separated*

Figure 8: *Yellow Layer with increased darkness*

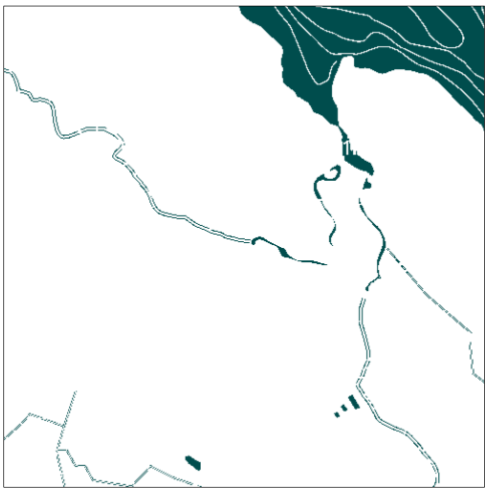

Figure 9: *Blue Layer with increased darkness*

3.2 Filtering the layer

In this step, the decomposed layers are further processed using morphological techniques such as dilation, opening, erosion, and closing.

We used the morphMask function to filter in MATLAB.

```
layer1 = morphMask(layer1_up,5);
layer2 = morphMask(layer2_up,5);
layer3 = morphMask(layer3_up,2);
layer4 = morphMask(layer4_up,2);
```

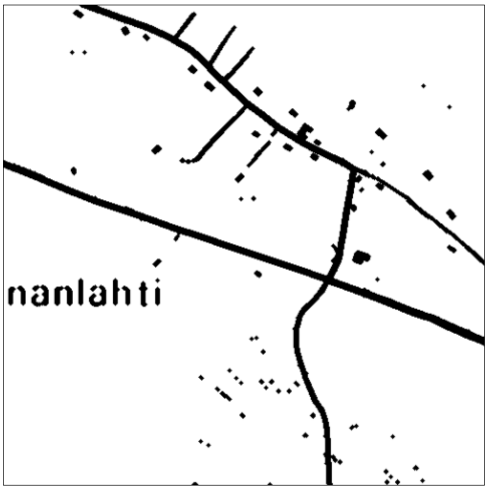

Figure 10: *Black Layer Morphed*

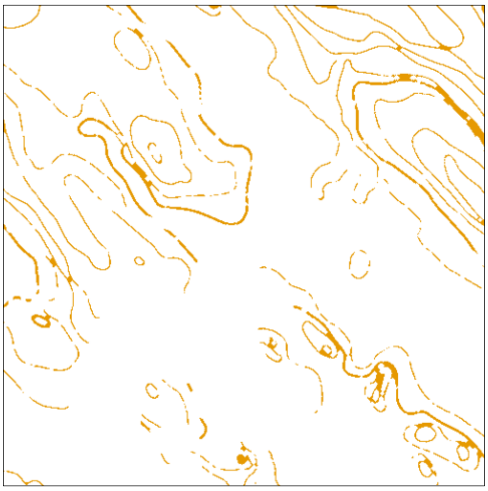

Figure 11: *Brown Layer Morphed*

Figure 12: *Yellow Layer Morphed*

Figure 13: *Blue Layer Morphed*

3.3 Reconstruction of the map image

The last and decisive step of the multilayer approach combines multiple layers of an image into a single RGB image, ensuring that areas without any information (background) are kept white. The process involves checking each layer for color information and merging them to prioritize non-background pixels. Here's a step-by-step explanation of what the MATLAB code does.

The function 'segmentedRGBImage' is initialized to an all-white image of the same size as the input image. Moreover, a mask (whiteBackgroundMask1 to whiteBackgroundMask4) is created for each layer to find white background pixels (where all RGB values are greater than or equal to 240). Next, the inverse of white background masks is created (colorMask1 to colorMask4) to find where each layer has color.

Finally, the merging process starts from the bottom layer (layer 1) and moves upwards. For each layer, it assigns pixels to segmentedRGBImage where the layer has color, ignoring white background pixels. Moreover, it ensures that any remaining pixels in segmentedRGBImage that are still white (255 in all channels) stay white. The final merged Image is displayed and saved as "FinalImage_Morphed.png."

```
%% recombining layers
% Initialize the segmented RGB image with a white background
segmentedRGBImage = uint8(255 * ones(size(image)));
% Define the white background mask for each layer
whiteBackgroundMask1 = all(layer1 >= 240, 3);
whiteBackgroundMask2 = all(layer2 >= 240, 3);
whiteBackgroundMask3 = all(layer3 >= 240, 3);
whiteBackgroundMask4 = all(layer4 >= 240, 3);
% Create a mask for where each layer has a color
colorMask1 = ~whiteBackgroundMask1;
colorMask2 = ~whiteBackgroundMask2;
colorMask3 = ~whiteBackgroundMask3;
colorMask4 = ~whiteBackgroundMask4;
```

```
% Merge the layers with alpha blending (where each layer's colors are prioritized)
% Start from the bottom layer (layer1) and move upwards
segmentedRGBImage(repmat(colorMask1, [1, 1, 3])) = layer1(repmat(colorMask1, [1, 1, 3]));
% Only add layer2 where it has color (not white)
layer2Mask = colorMask2 ;
segmentedRGBImage(repmat(layer2Mask, [1, 1, 3])) = layer2(repmat(layer2Mask, [1, 1, 3]));
% Only add layer3 where it has color (not white)
layer3Mask = colorMask3 ;
segmentedRGBImage(repmat(layer3Mask, [1, 1, 3])) = layer3(repmat(layer3Mask, [1, 1, 3]));
% Only add layer4 where it has color (not white)
layer4Mask = colorMask4 ;
segmentedRGBImage(repmat(layer4Mask, [1, 1, 3])) = layer4(repmat(layer4Mask, [1, 1, 3]));
% Ensure the background is white where no layers are present
finalBackgroundMask = all(segmentedRGBImage == 255, 3);
segmentedRGBImage(repmat(finalBackgroundMask, [1, 1, 3])) = 255;
figure;
imshow(segmentedRGBImage), title("Final Image")
imwrite(segmentedRGBImage,['FinalImage_Morphed',' .png'])
```

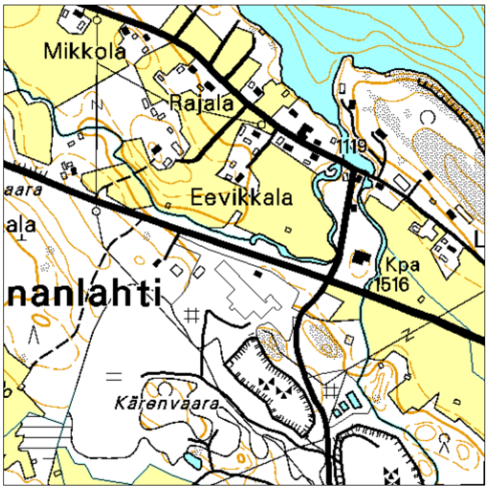

Figure 14: *Input Image of a city map*

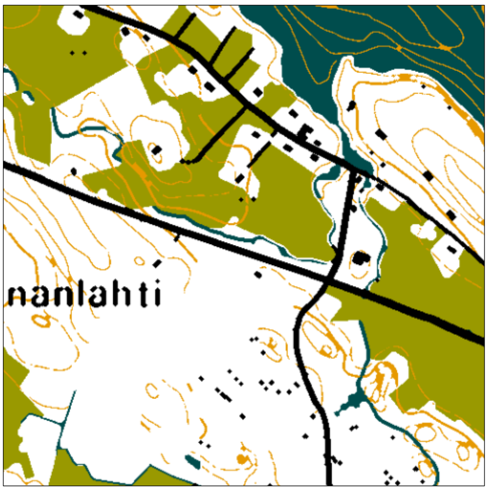

Figure 15: *Reconstructed Image after filtering*

3.4 Flow chart working diagram

The following flow chart diagram explains how mathematical morphology works on map images.

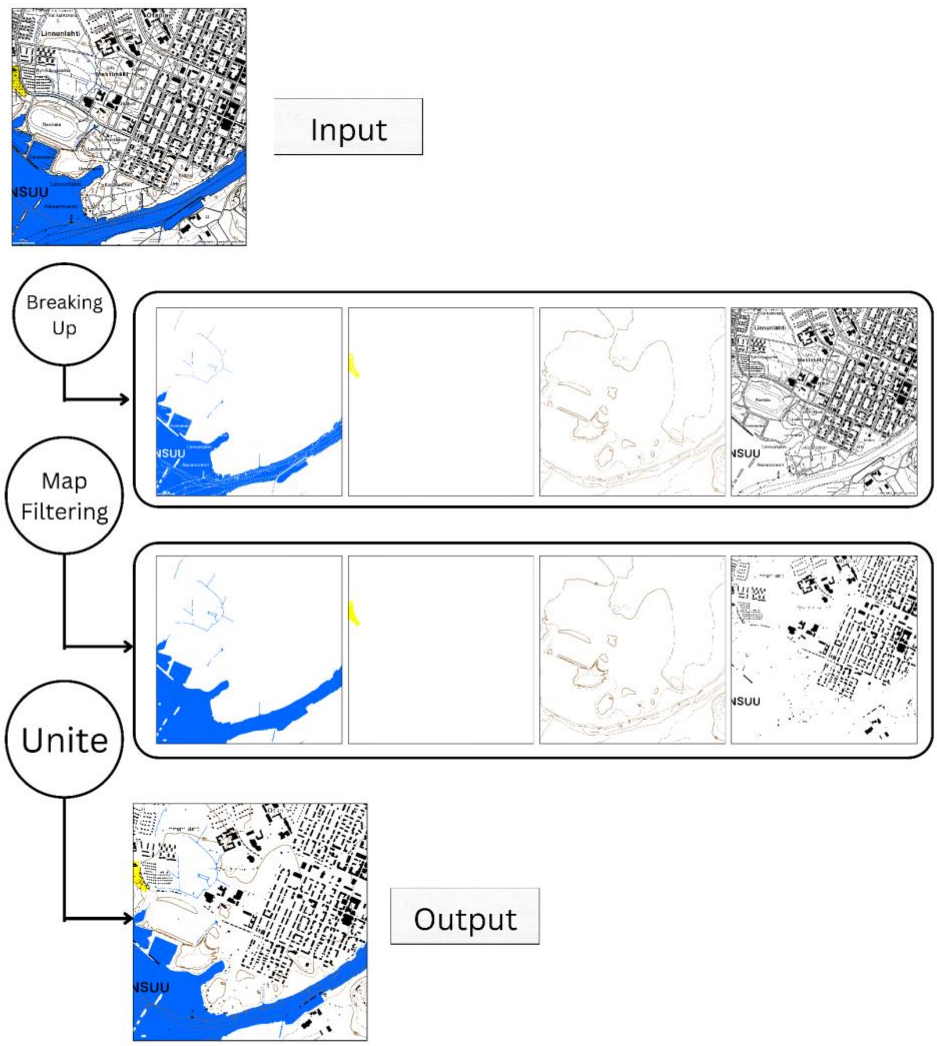

Figure 16: *Flowchart diagram of morphological filtering of a map image of Joensuu*

4 Filtering

Morphological filtering is a common technique in image processing for labeling images and is often used in segmentation tasks. Label images are typically made up of regions marked with unique labels (integers) that find different objects or areas. Morphological operations can help clean up and refine label images by removing noise, filling gaps, and improving the structure of labeled areas. Here are some everyday morphological operations used to label images.

4.1 Dilation

Morphological operations are a critical aspect of image processing, providing techniques to probe and change the geometric structure within an image. One such operation is dilation. When applied correctly, dilation can significantly enhance an image's features, making it easier to analyze and interpret. In simple terms, dilation is a process that adds pixels to the boundaries of objects in an image.

The extent and shape of this addition depend on the structuring element used. The primary purpose of dilation is to expand the shapes within an image, which can help bridge gaps or connect disjointed elements. Dilation works by placing the center of the structuring element on each Image pixel and then comparing the overlapping pixels. If any overlapping pixels are set (i.e., have a value of 1), the pixel beneath the center of the structuring element is also set to 1. This process results in the expansion of the object boundaries.

Applications of dilation include enhancing object boundaries, making objects more prominent by thickening their boundaries, filling small gaps, and connecting broken parts of an object, making it more cohesive. Furthermore, its smoothest object contours and eases the rough edges of objects, providing a more precise shape.

Dilation is applied using the following code

dilatedImage = imdilate(image, se);

where se is the structuring element

While dilation of a set A by structuring element B: all z in A such that B hits A when origin of B=z

$$A \oplus B = \{ z \mid (\hat{B}) \cap A \neq \emptyset \}$$

The following images show the effect of dilation on the coast of a river.

Figure 17: *Image before dilation*

Figure 18: *Image after dilation*

4.2 Erosion

Morphological filtering is a common technique in image processing for labeling images and is often used in segmentation tasks. Label images are typically made up of regions marked with unique labels (integers) that show different objects or areas. Morphological operations can help clean up and refine label images by removing noise, filling gaps, and improving the structure of labeled areas. Here are some everyday morphological operations used to label images. Image erosion is a fundamental process in digital image processing, specifically within the morphological operations typically applied to binary images, such as label images. This technique diminishes the boundaries of foreground (white) regions in a binary image.

The term "erosion" aptly describes the process, as it effectively "erodes" away the edges of foreground pixels, reducing the overall size and shape of objects within the Image. The erosion process involves sliding a structuring element into a small matrix or kernel over the Image. The structuring element defines the neighborhood of each pixel for the erosion operation. First, it places the structuring element on the Image so that its center aligns with a pixel in the input image. The process determines if all pixels under the structuring element match the Image's white (foreground) pixels. If the structuring element fills the white region, the output image's center pixel is set to white. The pixel is converted to black (background) if it does not fit. In basic terms, a pixel in the output image is assigned the lowest value of the pixels in its neighborhood as the structuring element decides. This causes the white areas to shrink.

The following equation defines erosion for Sets A and B in Z^2:

$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$

$$A \ominus B = \{ z \mid (B)_z \cap A^t = \emptyset \}$$

The erosion of A by B is the set of all points z such that B, translated by z, is contained in A. However, the following code is used in MATLAB to apply erosion on an input image.

```
erodedImage = imerode(image, se);
```

where se is the structuring element

The following pictures show the effect of erosion on an input image

Figure 19: *Image before Erosion*

Figure 20: *Image after Erosion*

4.3 Opening

Opening is a morphological operation that is especially useful for cleaning up label map images by eliminating tiny objects or noise while retaining the shape and size of more significant objects. It is a two-step process that involves erosion following dilation. Erosion removes pixels from the borders of objects. As a result, small objects are entirely removed, while larger objects' sizes are reduced.

Dilation brings the pixels to the outer edges of objects, increasing their size. Opening combines these two operations to smooth the edges of objects, break narrow islands, and cut tiny components. It helps reduce noise by removing small noise elements from the Image and keeping the integrity of larger structures within it. It also enhances the clarity of labeled regions for better interpretation.

Opening helps reduce noise by removing small noise elements from the Image and maintaining the integrity of larger structures within the Image. Also, it enhances the clarity of labeled regions for better interpretation.

Erosion followed by dilation, denoted by $\circ$

$$A \circ B = (A \ominus B) \oplus B$$

A stand for the primary Image,

B is the structuring element

$\ominus$ denotes erosion and $\oplus$ dilation.

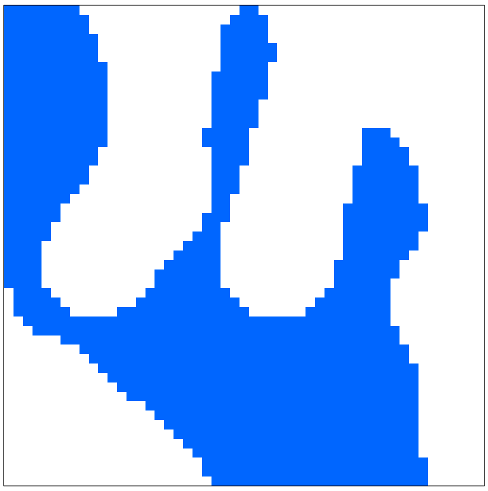

Figure 21: *50*50 Image before opening*

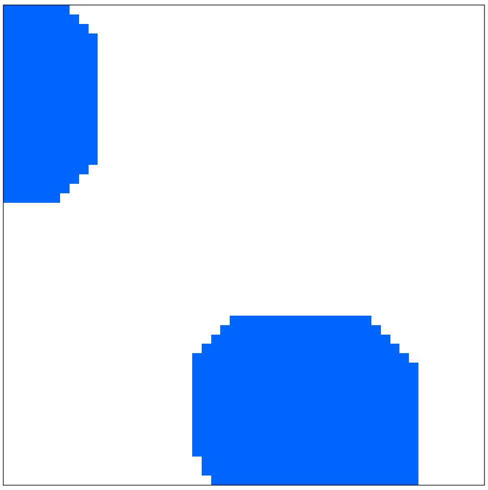

Figure 22: *50*50 Image after opening*

4.4 Closing

Closing is a morphological operation that aims to close small gaps and holes in an image's foreground objects while keeping those objects' overall shape and size intact. By dilating and then eroding the Image with the same structural element, closing effectively smooths object boundaries, fills in gaps, and can help merge objects that are close to each other. This operation is particularly useful in Image preprocessing to prepare images further.

The following code is applied in MATLAB for closing.

```
% Apply the dilation
dilatedImage = imdilate(image, se);
% Apply the erosion
closedImage = imerode(dilatedImage, se);
```

The mathematical equation of closing is as follows:

The closing of set A by structuring element B, expressed by A • B, is

$$A \bullet B = (A \oplus B) \ominus B$$

Here, A stands for the original Image, while B is the structuring element

$\ominus$ denotes erosion and $\oplus$ dilation.

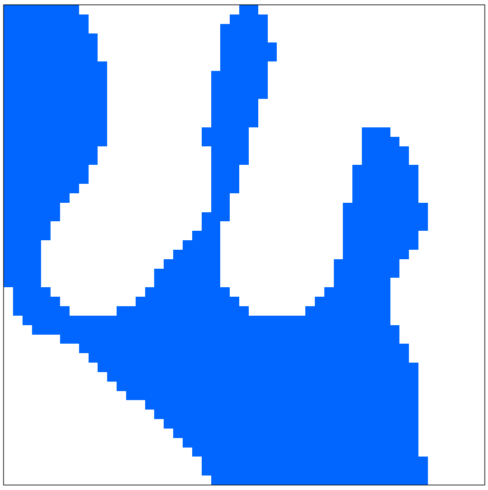

Figure 23: *Image before closing*

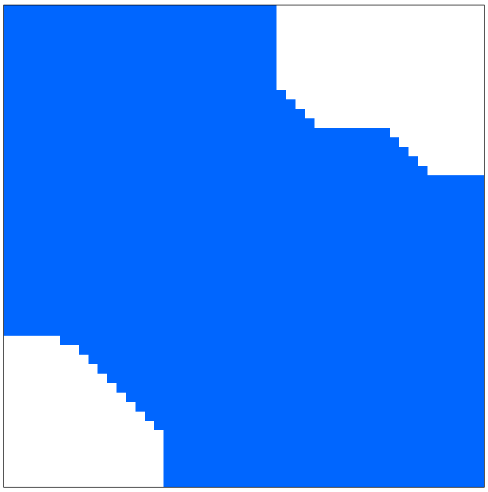

Figure 24: *Image after closing*

5 Frequency

The frequencies of patterns have been calculated by storing the values that the 3x3 window covers in the binary as a key to the HashMap, and the count of how many times that pattern appears is stored as the value of that key. This process allows you to analyze the frequency of different patterns within the 3x3 windows. Patterns with higher counts may say areas of interest or specific features in the dataset.

Here are the calculated values printed on the console for each binary and the patterns it owns.

5.1 Frequency for 3x3 images

Table 2: *Top frequencies of patterns (3x3 window)*

Key	Layer 1	Layer 2	Layer 3
0 0 0 1 1 1 1 1 1	215	864	1251
0 1 1 0 1 1 0 1 1	201	654	1290
1 0 0 0 0 0 0 0 0	97	703	1039
1 0 0 1 1 0 1 1 1	103	573	2726
1 0 0 1 1 1 1 1 1	67	1155	2145

5.2 Frequency for 8x8 images

Table 3: *Top Frequencies of Patterns (8x8 window)*

Key	Layer 1	Layer 2	Layer 3
0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1	129	740	2360
0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1	129	258	788
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	242	603	3305
0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1	129	285	3337
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	69	309	802
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	253	808	682

6 Experiments

Several experiments are performed using the same multilayer approach on label images to determine the applications of the process. The cases differ in terms of label removal, selected layer removal, two layers removal, noise removal, types of noises, entropy, etc. Some experiments are as follows.

6.1 Experiment 1 (Label Removal)

The first experiment is a simple test that shows the morphology on different layers of an image and removes the labels to clean the image. The test includes an image with four different colors i-e Black, Blue, Yellow, and Brown. In the separation stage, the black layer contains the map data and the labels as well because both have the same color. While the other three images have gaps between them.

In the next stage, the closing operation is performed on every layer to fill the gaps and remove the maximum labels in the black layer. A structuring element of 3x3 is used to implement the closing operation. Moreover, the layers are rejoined to get a final output. The recombining of the layers is done by adding the extracted layers on a white layer and the sequence is blue, yellow, brown, and black. The layers are combined one by one, and the sequence is the same for extracting and recombining the layers.

The resultant image shows the filling in three layers while the black layer has removed some useful data like roads instead of just removing labels. This is the limitation of this experiment because the labels have the same color as roads and bridges. Removing labels results in the removal of the data having the same color. The stepwise process diagram is attached below.

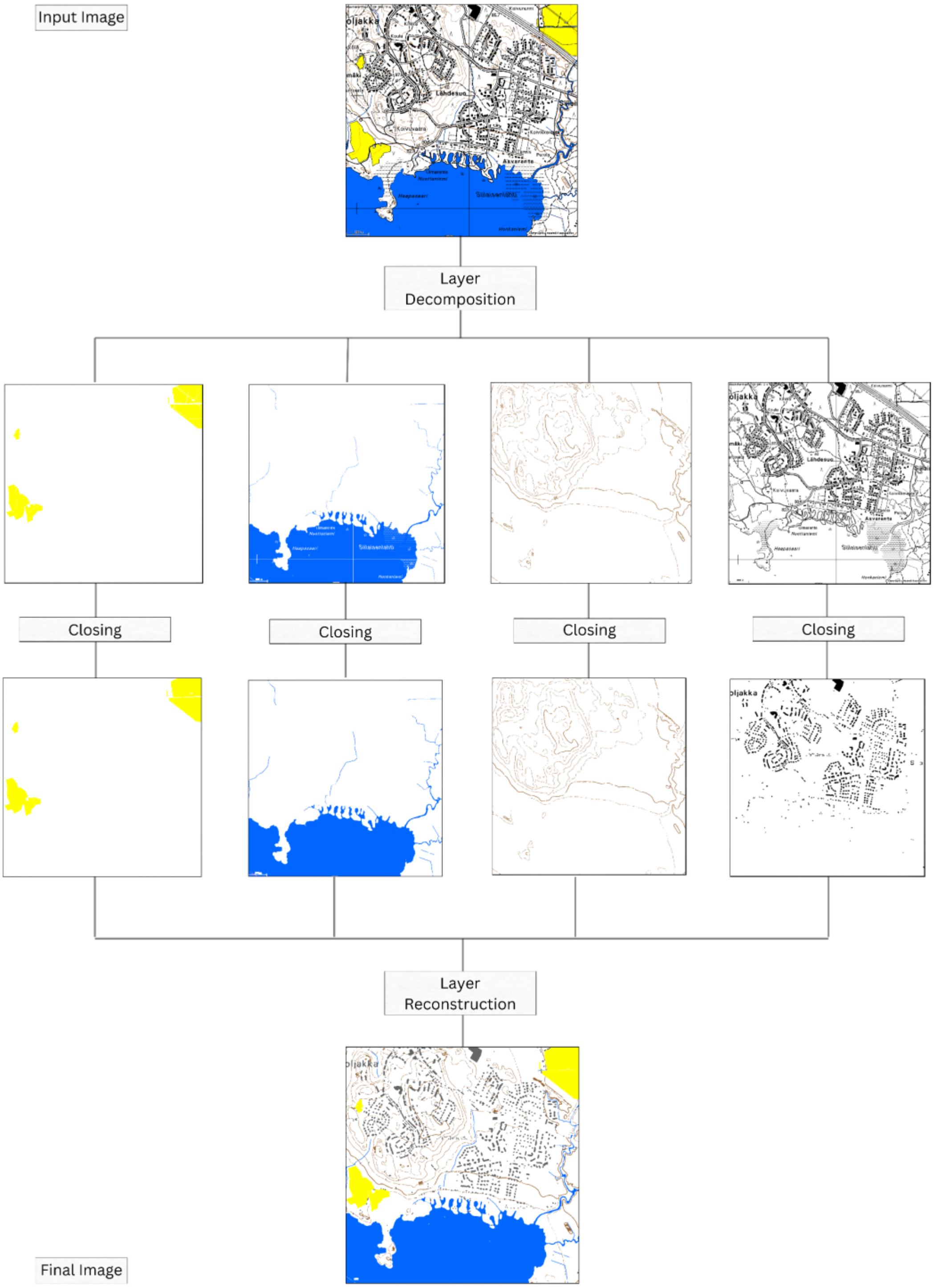

Figure 25: *Experiment of morphological filtering on a city map*

6.2 Experiment 2 (Comparison between two sizes of an image)

The following Image compares two images of varied sizes: one is 5000x5000 pixels, and the other is 50x50 pixels. The input images are the same, but one is a subpart of another. Both are processed with the same operation and window. The structuring element in both cases is the same (3x3).

The first input (5000x5000) is morphed using the closing function, and the black layer is removed. The remaining layers are rejoined to get an output image. In the next step, a sub-part (0 675 50 50) of the resultant Image is taken out as a final image. This part is compared with the result of the second scenario.

The second input (5000x5000) is cropped, and the subpart (0 675 50 50) is taken out for further processing. The cropped part is then processed using the same closing function as used in the first scenario. Thus, the black layer is removed from the cropped part as well. The resultant image is compared with the final image of the first case.

The resultant images from both cases mentioned above are identical and it shows the success in the removal of a required layer. Figure 26 shows the comparison among both cases and numbering is given to ease the understanding. For instance, images 3 and 6 are the results of input 1 and input 2 respectively, and are identical.

Overall, the process of morphological filtering is the same as the experiment 6.1 but the limitations are the least.

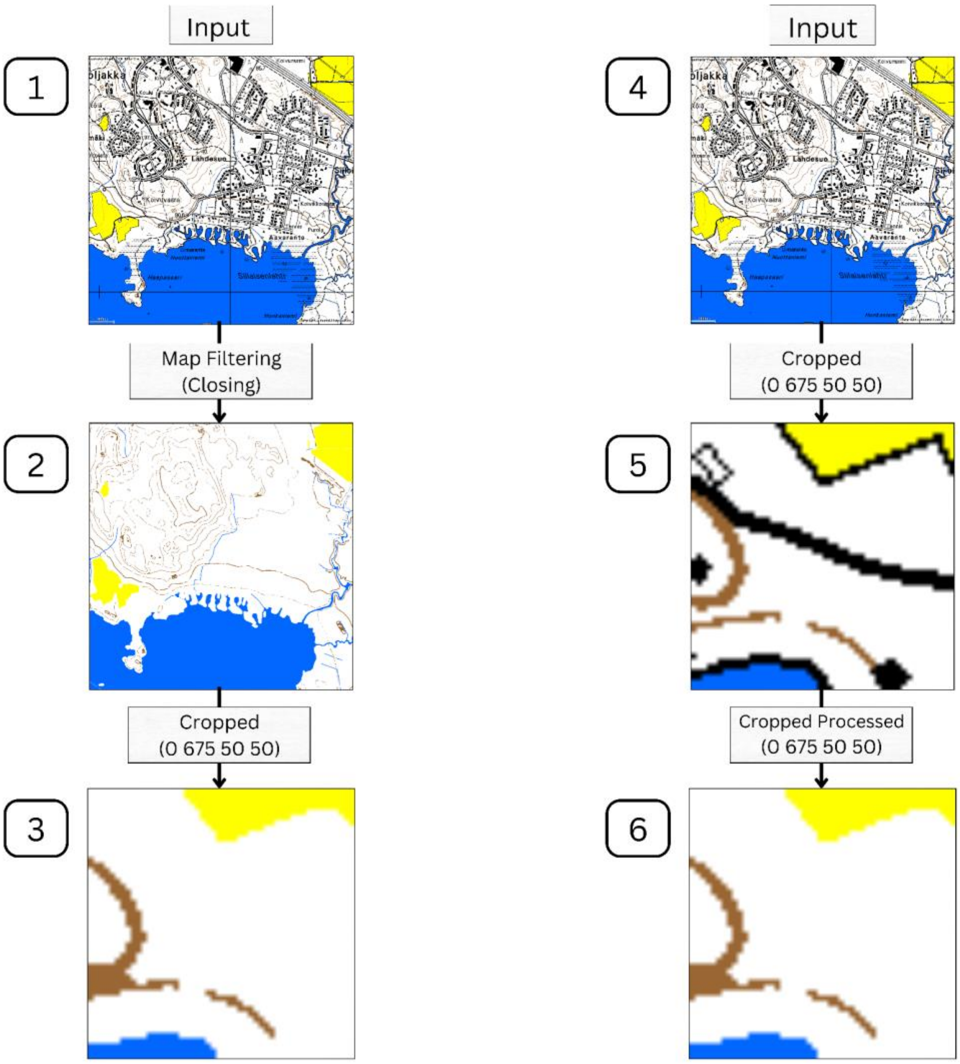

Figure 26: *Experiment diagram of two different scenarios*

6.3 Experiment 3 (Layer Removal)

This experiment involves the removal of a selected layer and reconstruction of the image by processing water and field layers by closing operation. The test shows the reconstruction of an image without using all the layers present in the original image. The first step is to decompose the images into several layers according to their color. The decomposed unprocessed layers are as follows:

Figure 27: *Decomposition of image into three layers*

This image contains three colors cyan, brown, and black. Cyan represents water while the contours are brown. The labels and map lines are black, and they are removed as part of our experiment. The remaining two layers of water and contours are further processed by closing operation to fill in the gaps that occurred due to the removal of the black layer. The structuring element of 3x3 is applied and it helped to recover the water layer.

In the last step, the merging of the processed layers is performed. The sequence of layer merging does not affect the final image due to a smaller number of layers. The input and output images are attached below, and they show the success of the experiment.

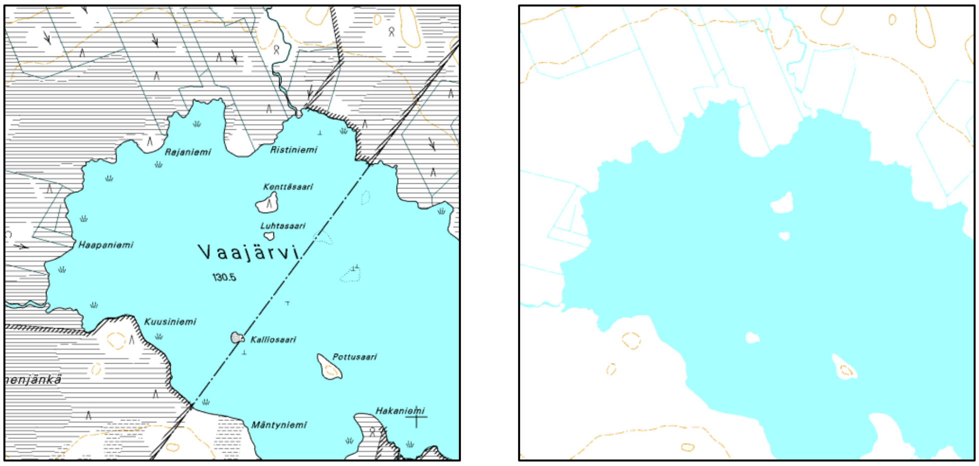

Figure 28: *Input Image and Reconstructed Image without Black Layer*

6.4 Experiment 4 (Layer Removal)

This experiment is the same as the previous experiment, but the input image is changed to compare the results. The number of layers in this scenario is four while the final image is reconstructed by only three layers. Among four, the black layer from the input image is removed.

The remaining three layers (Blue, contour, Yellow) are processed by applying the closing operation and a structuring element of 3x3 is used. The resultant image is as follows.

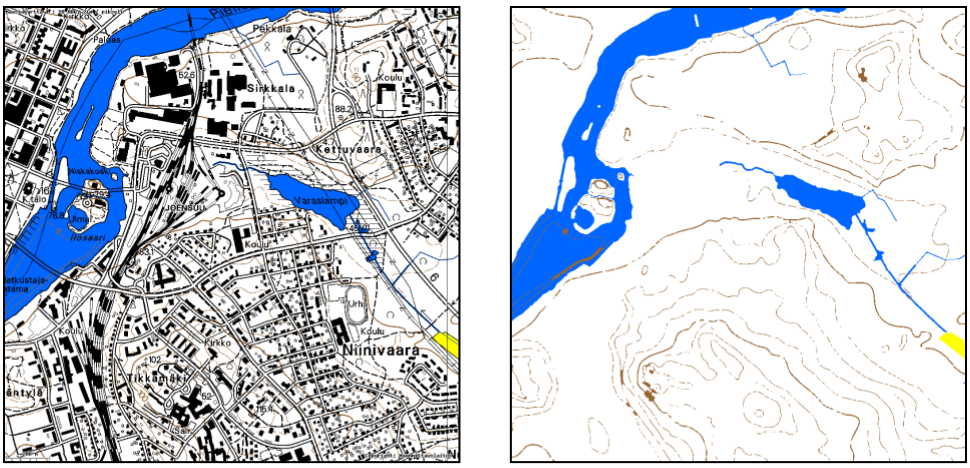

Figure 29: *Input Image and Reconstructed Image without Black Layer*

6.5 Experiment 5 (Layer Removal)

Removal of selected layers and reconstruction of the image by processing water and field layers by closing operation.

In this experiment, the image that is used has four different color layers. The first scenario removes the contours and reconstructs the image without using contours. Two layers (water, and yellow) are processed by applying the closing operation. The black layer is not processed in this case. The process starts by decomposing the layers according to their colors.

The layer with brown color is the contours and those are removed. The black layer remains the same to avoid any loss in the labels of the image. A structuring element of 5x5 is used in this case. In the merging phase, the black layer is put on the top while at the back is the water layer. The input image and the resultant image are shown in Figure 30.

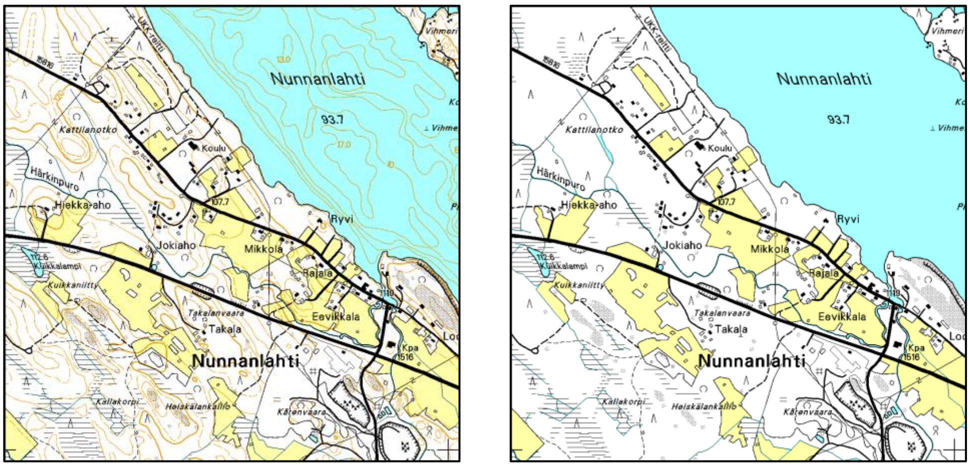

Figure 30: *Input Image and Reconstructed Image without Contours*

Furthermore, a new operation is applied to the input image where two layers (contours and black) are removed from the input image. So, two out of four are removed while the other two recombine to form a final image. The remaining layers are processed using a 5x5 structuring element and a closing operation is performed. The input image and their results are as follows:

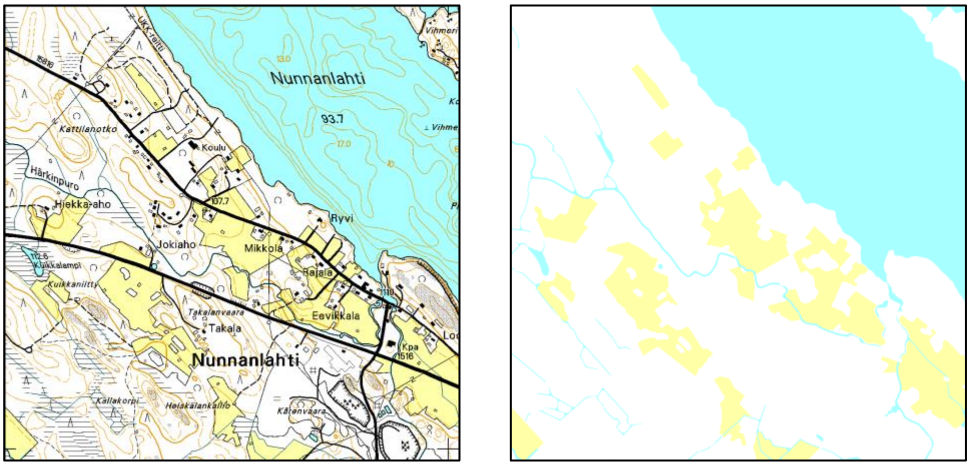

Figure 31: *Input Image and Reconstructed Image without Contours and Black Layer*

6.6 Experiment 6 (Noise Layer Removal)

In the previous experiments, a multilayer approach is used to remove labels and selected layers from an input image. Now this approach is performed on an image with noise and the goal is to remove the noise layer. The input image consists of nine distinct colors and the decomposition step is applied to separate each layer from the image. Layer separation also helps to identify the noise type and layer.

In this image dithering noise is present and the focus is to eliminate it. After successfully identifying the noise layer, it is then removed. The water layer is also reconstructed by applying a closing operation with a 5x5 window. All other layers are processed by a structuring window of 3x3. The image contains two types of labels i-e black and blue. So, these two layers are not processed to avoid any loss of data. In the merging step, the label layers are put on the top and the rest are ordered at the bottom of the labels.

The following order is used to merge the layers of this image:

{@greenMask, @silverMask, @yellowMask, @brownMask, @redMask, @cyanMask, @magentaMask, @purpleMask, @pinkMask, @whiteMask, @lightGrayMask, @goldMask, @darkGrayMask, @DarkGreenMask, @orangeMask, @blueMask, @blackMask};

The resultant image has all the required features the noise layer removed, labels restored, water reconstructed, and filled the rest of the gaps in the image. This process involved the use of if-else conditions at various stages of the code. The process also included the changes in the order of layers merging. In the merging phase, the order of layers is swapped accordingly to get a good final image. If there is no color range present for a layer, a blank white layer is produced in the separation stage.

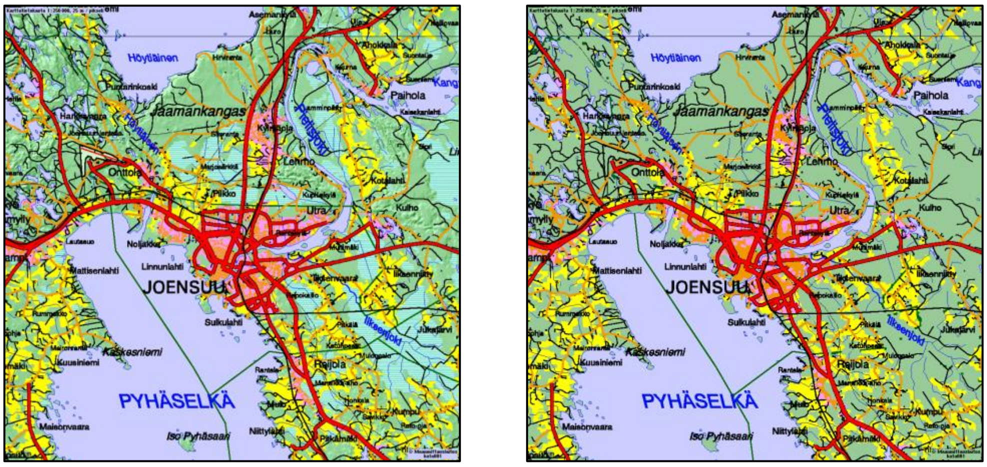

Figure 32: *Input Image and Reconstructed Image without Noise*

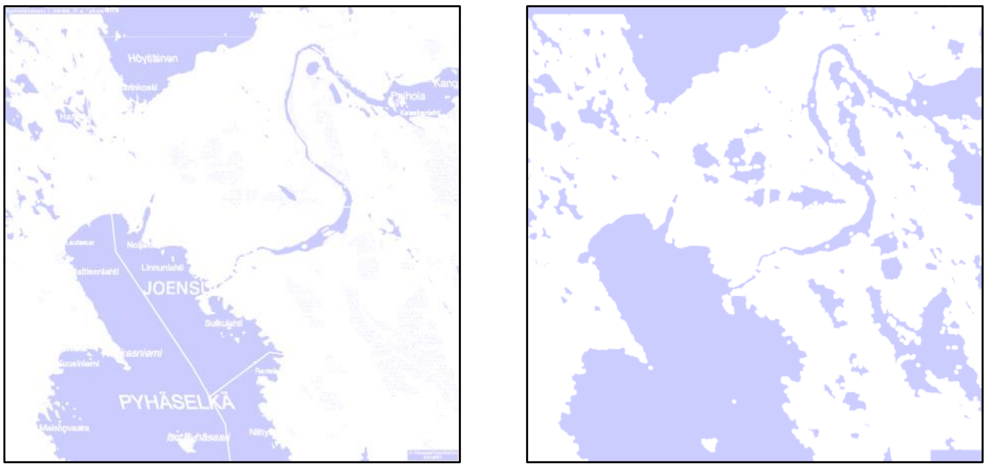

Figure 33: *Unprocessed and Processed Water Layer*

Overall, the merging algorithm is an important part where the number of layers is more. Merging always has an impact on the visibility of layers and thus determining the right order results in a better image. This experiment is performed by applying several orders to find a perfect combination for the merger.

6.7 Experiment 7 (Noise Layer Removal)

Like the previous experiment, this case also removes the noise layer and reconstructs the image by processing the water layer. The input image contains twelve distinct colors while the back layer (water) contains most of the noise. The water layer is reconstructed with a 5x5 window, and the rest of the layers are not processed. The experiment successfully removes the noise, and a new processed image is generated.

The merging is applied by using a white layer as a start layer. Then the water layer comes over it and after that are purple and red layers. The topmost layer is black as it has labels on it. The input image and the resultant image are as follows:

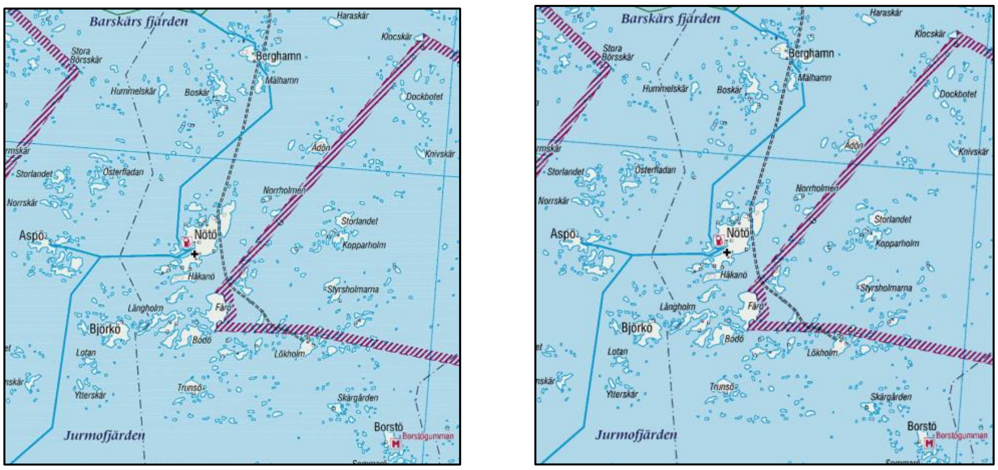

Figure 34: *Input Image and Reconstructed Image without Contours and Black Layer*

7 Conclusions

This thesis research investigated an innovative technique for processing labeled images using a multilayer approach and morphological operations. The basic concept was to decompose images into individual layers, apply targeted morphological operations to each layer, and then recombine these layers to create a new, morphed image according to the desire. It helps to remove labels, preserve labels, remove selected layers, and remove noise from an image. This method aimed to improve image quality and utility by effectively removing noise while keeping and enhancing key image features.

The division of images into multiple layers enabled a more specific and focused use of morphological operations. The multilayer approach was highly effective at removing labels and noise from images. Labels were removed by isolating them into separate layers and applying proper morphological operations without interfering with the underlying image content. Carefully using dilation, erosion, opening, and closing operations across different layers ensured that critical details were preserved or even improved in the recombined Image. The recombined morphed images showed better overall quality.

Overall, it is a useful technique to modify color layers in an image by using morphological operations.

8 REFERENCES

[1] V. Bhateja, M. Nigam, A. S. Bhadauria, A. Arya, and E. Y. D. Zhang, "Human visual system based optimized mathematical morphology approach for enhancement of brain MR images," *J. Ambient Intell. Humans. Comput.*, vol. 15, no. 1, pp. 799–807, 2024, doi: 10.1007/s12652-019-01386-z.

[2] J. Wang, W. Huang, and F. Biljecki, "Learning visual features from figure-ground maps for urban morphology discovery," *Comput. Environ. Urban Syst.*, vol. 109, 2024, doi: 10.1016/j.compenvurbsys.2024.102076.

[3] X. Xu, Z. Yu, W. Y. Chen, A. Chen, A. Motta, and X. Wang, "Automated analysis of grain morphology in TEM images using a convolutional neural network with CHAC algorithm," *J. Nucl. Mater.*, vol. 588, 2024, doi: 10.1016/j.jnucmat.2023.154813.

[4] J. Y. Jue Hou, "A mathematical morphology-based detector for underwater target detection using side-scan sonar images," in *International Conference on Remote Sensing, Mapping, and Image Processing (RSMIP 2024)*, 2024, p. 131670D. Doi: https://doi.org/10.1117/12.3029651.

[5] K. S. Radhakrishna Achanta, Appu Shaji, and S. S. Aurelien Lucchi, Pascal Fua, "SLIC superpixels compared to state-of-the-art superpixel methods," *J. Latex Cl. files*, vol. 6, no. 1, 2011.

[6] P. Fränti, "Efficiency of random swap clustering," *J. Big Data*, vol. 5, no. 1, 2018, doi: 10.1186/s40537-018-0122-y.

[7] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution grayscale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, 2002, doi: 10.1109/TPAMI.2002.1017623.

[8] A. Podlasov, "Morphological reconstruction of semantic layers in map images," *J. Electron. Imaging*, vol. 15, no. 1, p. 013016, 2006, doi: 10.1117/1.2178188.

[9] A. Podlasov, "Lossless image compression via bit-plane separation and multilayer context tree modeling," *J. Electron. Imaging*, vol. 15, no. 4, p. 043009, 2006, doi: 10.1117/1.2388255.

[10] A. Podlasov and E. Ageenko, "Java Tools for Morphological Image Analysis and Research," *Univ. Joensuu - Dept Comput. Sci. - Rep. Ser. A*, vol. A-2004-2, 2004, [Online]. Available: ftp://www.cs.joensuu.fi/pub/Reports/A-2004-2.pdf

[11] P. Kopylov and P. Fränti, "Compression of map images by multilayer context tree modeling," *IEEE Trans. Image Process.*, vol. 14, no. 1, pp. 1–11, 2005, doi: 10.1109/TIP.2004.838694.

[12] P. Fränti and O. Nevalainen, "Compression of binary images by composite methods based on block coding," *J. Vis. Commun. Image Represent.*, vol. 6, no. 4, pp. 366–377, 1995, doi: 10.1006/jvci.1995.1030.

[13] P. Fränti, E. Ageenko, P. Kopylov, S. Gröhn, and F. Berger, "Compression of map images for real-time applications," *Image Vis. Comput.*, vol. 22, no. 13, pp. 1105–1115, 2004, doi: 10.1016/j.imavis.2004.05.009.

[14] E. Ageenko, P. Kopylov, and P. Fränti, "On the size and shape of multi-level context templates for compression of map images," *IEEE Int. Conf. Image Process.*, vol. 3, pp. 458–461, 2001, doi: 10.1109/icip.2001.958150.

[15] Fränti, P., Kopylov, P., & Ageenko, E. (2002), "Evaluation of compression methods for digital map images," *IASTED International Conference on Automation, Control, and Information Technology (ACIT 2002), Novosibirsk, Russia*, pp. 418–422.

[16] P. Kopylov and P. Fränti, "Color quantization of map images," *Proc. Fourth IASTED Int. Conf. Vis. Imaging, Image Process.*, pp. 837–842, 2004.

[17] M. Chen, M. Xu, and P. Fränti, "Multilayer filtering approach for map images," *Proc. - Int. Conf. Image Process. ICIP*, pp. 3953–3956, 2009, doi: 10.1109/ICIP.2009.5414038.

[18] M. Chen, M. Xu, and P. Fränti, "Statistical filtering of raster map images," *2010 IEEE Int. Conf. Multimedia. Expo, ICME 2010*, pp. 394–399, 2010, doi: 10.1109/ICME.2010.5583054.

[19] A. Podlasov, P. Kopylov, and P. Fränti, "Statistical filtering of raster map images using a context tree model," *Proc. - Int. Conf. Signal Image Technol. Internet Based Syst. SITIS 2007*, pp. 501–508, 2007, doi: 10.1109/SITIS.2007.121.

[20] A. Podlasov, P. Fränti, and P. Kopylov, "Multilayer context tree modeling for lossless compression of grayscale images," *10th Finnish Artificial Intelligence Conference (STeP), 2005*, pp. 39–41.