Image Segmentation

Wanrong Xu
S1636119

Year 4 Project
School of Mathematics
University of Edinburgh
April 14, 2022

Abstract

Image segmentation is an important research topic in the field of digital image processing. In this project we introduce a range of topics to demonstrate how a combination of image manipulation, topology, mathematics and other knowledge to perform image segmentation before deep learning. This project first introduces various colour spaces, then describes clustering methods and some typical traditional image segmentation techniques to correct the colours of old maps and separate the land and sea of satellite images. The land and sea segmentation of satellite images is very important for research on detecting coastlines and nearshore targets. The colour correction of historical maps can be used to restore an old map and promote the research of historical geography.

Declaration

I declare that this thesis was composed by myself and that the work contained therein is my own, except where explicitly stated otherwise in the text.

(*Wanrong Xu*
S1636119)

Contents

Abstract ii

Contents v

1 Introduction 1
- 1.1 RGB colour image . . . 2
- 1.2 Grayscale image . . . 2
- 1.3 How images are stored in a computer? . . . 3
  - 1.3.1 How image pixels are stored? . . . 3
  - 1.3.2 Bitmap image & Vector image . . . 4
- 1.4 Colour spaces . . . 4
  - 1.4.1 RGB colour space . . . 4
  - 1.4.2 CMYK colour space . . . 5
  - 1.4.3 HSV colour space . . . 5
- 1.5 Colour quantization . . . 6
- 1.6 Historical maps . . . 6
  - 1.6.1 Historic map making . . . 7
- 1.7 Satellite images . . . 7
  - 1.7.1 Sentinel-2 Satellite . . . 8
- 1.8 Normalised Difference Vegetation Index . . . 9
  - 1.8.1 NDVI calculation . . . 10

2 Clustering 12
- 2.1 K-means clustering . . . 12
  - 2.1.1 K-means clustering for colour image . . . 13
  - 2.1.2 Advantages and disadvantages . . . 14
- 2.2 Mean shift clustering . . . 14
  - 2.2.1 Kernel density estimation . . . 15
  - 2.2.2 Mean shift vector . . . 15
  - 2.2.3 Gaussian kernel . . . 16
  - 2.2.4 The Mean Shift Procedures . . . 16
  - 2.2.5 Mean Shift in Image Segementation . . . 17

3 Principal Component Analysis 18
- 3.1 PCA algorithm . . . 18
- 3.2 Covariance matrix . . . 19
- 3.3 Singular Value Decomposition . . . 20

3.4 Dimensionality reduction . . . 21
3.5 Dimensionality reduction before clustering . . . 22

4 Image Segmentation Techniques **24**
4.1 Threshold-based technique . . . 24
4.1.1 Otsu's method . . . 24
4.2 Edge detection techniques . . . 26
4.2.1 Differential operators . . . 26
4.2.2 Filters and convolutions. . . . 27
4.2.3 Comparison of different filters. . . . 28
4.2.4 Canny detector . . . 29
4.2.5 Edge-based segmentation . . . 31
4.3 Watershed transformation technique . . . 32
4.3.1 Watershed transformation segmentation . . . 33
4.4 Region-based technique . . . 34
4.4.1 Region growing . . . 34
4.4.2 Region splitting and merging . . . 35
4.4.3 Region-based method using the watershed transformation . 36
4.5 Active contour model . . . 37
4.5.1 Parametric active contour model . . . 37
4.5.2 Geometric active contour model . . . 39
4.5.3 Active contour segmentation . . . 39

5 Conclusion **40**

Contents **46**

Chapter 1

Introduction

With the increasing power of computer information technology, more and more people obtain information and solve problems by computer in our daily life and work. Therefore, the processing of image information has become an important research direction.

Image segmentation is a typical problem in the research for computer vision and has become crucial in the field of image understanding [29]. It is the first step in image analysis, the basis of many computer vision tasks, and an important part of understanding an image. However, it is also one of the most challenging problems in image processing.

Image segmentation can be divided into three distinct types of task: ordinary segmentation, semantic segmentation and instance segmentation [26]. Ordinary segmentation is to separate the pixel areas that belong to different objects. For example, to segment the foreground and background. Semantic segmentation is normally performed to classify the semantic content of each area on the basis of ordinary segmentation. For example, point out all the objects in the picture to their respective categories. Finally, instance segmentation is often performed to find and number the objects on the basis of semantic segmentation, such as number the 3 girls in an image as girl 1, girl 2 and girl 3.

In this project, I will first explain some important concepts about the image data and then introduce various techniques which I use to achieve ordinary segmentation.

One of the important tools of image segmentation is watershed transformation technique. The concept of this technique was first applied by Beucher and Lantuejoul in 1979 [32]. They divided lines to segmentation problems. The idea of watershed transformation technique is to regard the image as a topological topography. In that paper, they illustrate the technique with two examples: bubbles detection in a radiographic plate, and display of facets in a metallic fracture.

There are two purposes of this project. The first purpose is to correct the colours of historical maps by researching the clustering methods. The second purpose is to segment the sea and land areas of satellite images. To achieve better segmentation effect, different techniques are tested on satellite images.

1.1 RGB colour image

In an RGB image, the three primary colours red, green and blue represent the colour values of each pixel. They are directly stored in the image matrix, because the colour of each pixel needs to be represented by three components of R, G, and B.

By splitting the three channels of the RGB image, I got the R, G, B single-channel images which are all grayscale images as shown in Figure 1.1.

(a) RGB (b) R (c) G (d) B

Figure 1.1: RGB channels splitting

Then by merging the splitted single channel image with zero components, The red, green and blue channel colour images obtained(Figure 1.2).

(a) Red (b) Green (c) Blue

Figure 1.2: RGB channels merging

1.2 Grayscale image

Grayscale image is a degenerate version of colour image. The information saved is not as much as colour image. It only contains channel, while colour image usually contains three channels, like RGB colour image.

A grayscale image has only one sampled colour per pixel. Usually, this type of image can be shown as a grayscale from the darkest black(0) to the brightest white(1). They are different from binary images. In the field of computer graphics, binary images contain only two colours: black and white; however, grayscale images contain many levels of colour depth between black and white.

1.3 How images are stored in a computer?

Images are stitched together by pixels, which are small square regions with constant colour or brightness. The resolution of an image indicates the number of pixels in the image. A picture with a resolution of 1008×700 has 1008 pixels in length and 700 pixels in height, and there are total 705600 pixels in this image.

1.3.1 How image pixels are stored?

The information stored by each pixel in the computer is its RGB values, and these RGB values $(0 - 255)$ are stored in the hard disk in binary form.

The pixel values are stored in a matrix, and the size of the matrix depends on the colour model adopted by the image.

If it is a grayscale image, then the image is single-channel, and each pixel in the image only needs to store one matrix element, usually a value from 0 to 255.

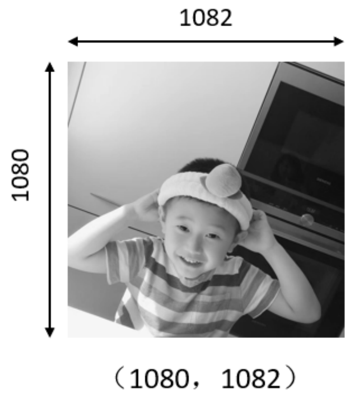


Figure 1.3: Pixels in grayscale image.

If it is a colour image, the image is multi-channel, and a pixel needs to store multiple matrix elements. The columns in the matrix will contain multiple sub-columns, and the number of sub-columns is equal to the number of colour channels.

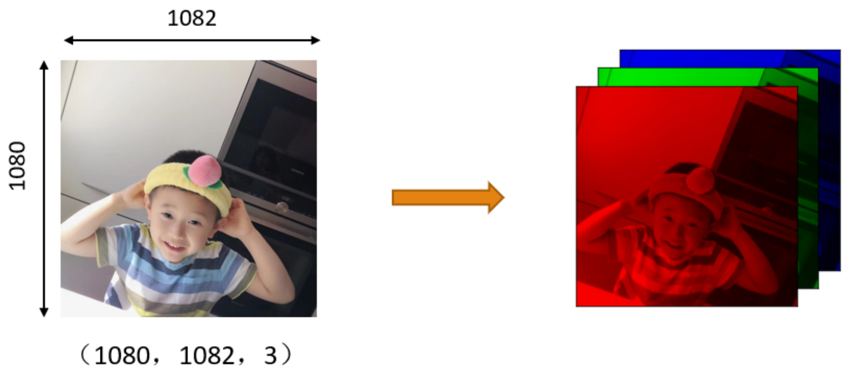


Figure 1.4: Pixels in RGB image.

As the examples shown above, the shape of the grayscale image is $(1080, 1082)$, which shows that there are 1080×1082 pixels in this image. Each pixel only stores one value$(0-255)$ since grayscale is single-channel. However, in the RGB image, the shape is $(1080, 1082, 3)$, which shows that there are 1080×1082 pixels and each pixel stores 3 values$(0-255)$ since RGB image contains 3 channels, so there are total $1080 \times 1082 \times 3$ values stored in this image.

1.3.2 Bitmap image & Vector image

There are two storage methods for digitized image data: bitmap and vector.

A bitmap image is an identifiable image which consists of a series of pixels. The advantage of bitmap storage is that it is simple and close to the real image. The disadvantage is that once the resolution is too small, the image will be jagged and blurred when zooming in.

A vector image does not describe every point of image data directly, it describes the process and method of generating these points. Mathematical equations are used to describe the edges and internal fillings to create the image. The advantage of vector storage is that it is affected by the size of images and can be scaled arbitrarily without distortion. In general, vector storage is more advanced than bitmap storage.

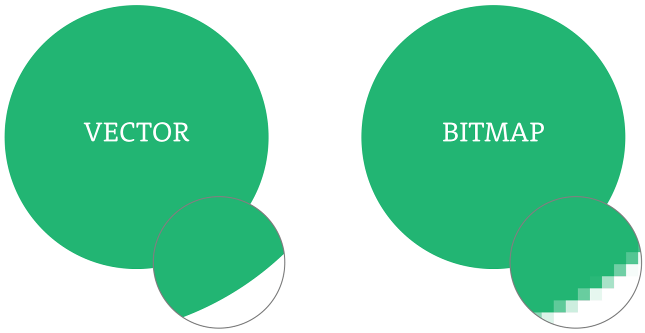


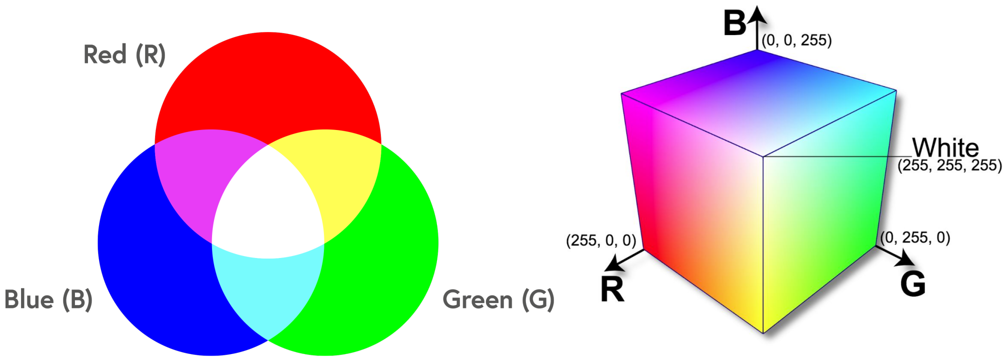
Figure 1.5: The difference between vector and bitmap/raster images. [4]

1.4 Colour spaces

Colour space can be used to describe colours. There are several common colour spaces such as RGB colour space, CMYK colour space and HSV colour space.

1.4.1 RGB colour space

The RGB colour space is based on the principle of additive colour mixing. Red, Green and Blue are the three colour channels of the RGB colour space. These three colour channels are continuously superimposed from black, and finally a bright white colour obtained.



(a) RGB colours (b) RGB colour space

Figure 1.6: The RGB colours [23] and RGB colour space [5].

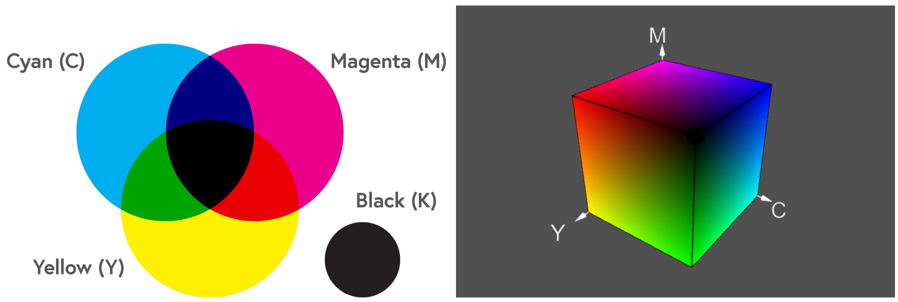
Taking the three channels of R, G, and B as the X, Y, and Z axes in the Cartesian coordinate system, this forms a cube called the RGB colour space (Figure 1.6).

The RGB colour space has high compatibility to machine, which is suitable for display on a screen.

1.4.2 CMYK colour space

The CMYK are the four inks **C**yan, **M**agenta, **Y**ellow and blac**K** which are normally used for colour printing. In contrast to RGB colour space, the CMYK colour space is based on the principle of subtractive colour mixing. It starts from white, and by repeatedly superimposing on the four inks, the colour becomes darker until obtain a pure black is obtained.

Taking the Yellow, Cyan and Magenta as the X, Y, and Z axes in the Cartesian coordinate system, this forms a cube called the CMYK colour space (Figure 1.7).



(a) CMYK colours (b) CMYK colour space

Figure 1.7: The CMYK colours [23] and CMYK colour space [35].

1.4.3 HSV colour space

The HSV colour space (Figure 1.8) is based on human visual perception.

The **H** in HSV represents the range of colours that the human eyes can perceive. These colours are distributed on a flat colour ring. The range is from 0°

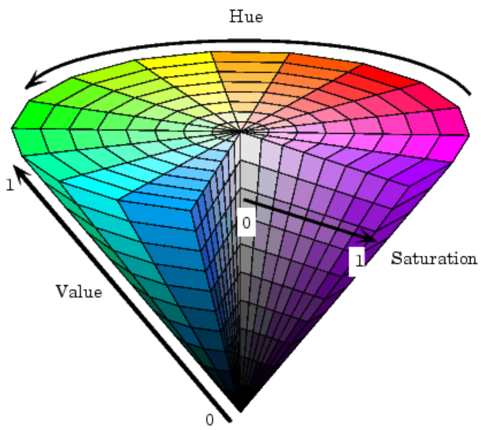


Figure 1.8: HSV colour space [10].

to 360°, and each angle represents a colour.

The **V** in HSV refers to the brightness of the colour, and the range is from 0% to 100%. The smaller the percentage, the darker the colour, and the closer to black(0); the larger the percentage, the brighter the colour, and the closer to white(1).

The **S** in HSV refers to the saturation of the colour, which uses a value from 0% to 100% to describe the change of colour purity under the same hue and brightness. The larger the percentage, the less gray in the colour, and the more vivid the colour will be, showing a change from gray to pure colour.

1.5 Colour quantization

Colour quantization is a compression process which reduces the number of colours in an image. Obviously all compression will have information loss. In this case, the generated image may be different from the original image. Therefore, the task of colour quantization is to compressed an image and keep as much information as possible from the original image, and the key factor in achieving this is to choose the colour palette by choosing the colour that best summarizes the original image.

The most common technique is to simplify the colour quantification to points clustering, where each point represents the colour of a pixel. It creates a colour palette by selecting the points represented by each cluster. After that, it simply remaps all the colours into the corresponding cluster centres. The colour palette obtained depends on the colour space and distance metric used.

1.6 Historical maps

Maps have the unique functions of expressing the real world in many ways. They allows you to identify spatial distributions, relationships and trends that cannot be reflected in other ways. They can analyze spatial issues by combining or superimposing data. With the continuous development of human society, there are more and more varieties of maps. There are many types of classification, such as by scale, by mapping area, by map function, by content and other classification indicators.

A historical map may record the environment, politics, ethnicity, culture, militarisation, and national boundaries of certain periods in the history of a region. Historical maps show great value for history, geography, geology, and historical geography. As shown in Figure 1.9, this is a historical map image of 'Lothian and Linlitquo' on the National Library of Scotland website[31].

Figure 1.9: Historical map[31].

1.6.1 Historic map making

The origin of maps and the development of cartography have a long history. Traditional cartography is to use surveying, statistical and other methods to obtain topographical, professional or topical data, and to compile maps expressed in the form of graphics, symbols, and colours through manual processing, sorting, design, and decoration. Through a series of process such as clearing and drawing, and colour separation, the final printing is completed. Every process cannot be separated from manual operation, the process is complicated, and the standard is difficult to be unified.

1.7 Satellite images

Satellite images use satellites to give us a true representation of surface image on the earth. Unlike traditional maps, the topography seen on satellite maps is real and real-time. Therefore, satellite maps is widely used in many applications, such as detecting ground information and geographic location and terrain. Satellite mapping can also be applied to military applications,navigation systems, urban planning, rural planning, disaster monitoring and disaster relief deployment.

Using the GPS navigation system of the satellite mapping like Google Maps, you can tell where you are, which way to go and other information. Furthermore, if the monitoring satellite mapping is real-time, it can also be used by the police to hunt down wanted suspects.

1.7.1 Sentinel-2 Satellite

The satellite images used in this project are taken by the Sentinel-2 satellite [1]. It consists of two sensors, Sentinel-2A and Sentinel-2B. Alternatively, Sentinel-2 (Figure 1.10) is a multi-spectral imaging satellite with high-resolution. It carries a Multi-Spectral Instrument (MSI) for land monitoring which can provide images of vegetation coverage, also soil and water coverage. It can not only be used for improving forestry and agricultural planting, predicting the production of food, and ensuring food security, but also for monitoring natural disasters for instance floods, volcanic eruptions, landslides, and providing assistance for humanitarian relief.

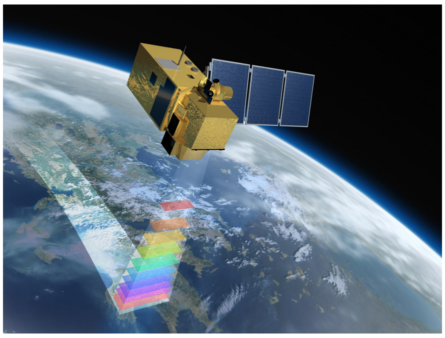


Figure 1.10: Sentinel-2 satellite, European Space Agency [9]

Multi-spectral imaging refers to an image that contains 3 or more bands, sometimes even hundreds. Each band is a grayscale image that represents the brightness of the scene based on the sensitivity of the sensor which used to generate the band. In such an image, each pixel is related to a string of values in different bands by the pixel. This number string is called the spectral marker of the pixel.

There are different types of spectral bands based on the different wavelengths of light. The spectral bands with wavelength between $380nm$ and $780nm$ are called visible light bands. The spectral bands with wavelength shorter than $380nm$ are called ultraviolet light bands. The spectral bands with wavelength longer than $780nm$ are called infrared light bands. These can be further divided into near infrared, mid infrared and far infrared,etc.

The MSI for Sentinel-2 has a height of 786km and can cover 13 spectral bands with different resolutions 10m, 20m, and 60m [1]. The revisit period of one satellite is 10 days, and the revisit period of two satellites is 5 days. From visible light and near infrared to shortwave infrared, it has different spatial resolutions.

The 13 spectral bands are shown in the table (Figure 1.11).Band 1 refers to the coastal and aerosol band, which is used to monitor nearshore water bodies and aerosols in the atmosphere. Bands $2, 3, 4$ are the blue, green and red visible light bands; bands $5, 6, 7$ are the bands within the red edge which are very effective for monitoring vegetation health information; band 8 is the near infrared band and band 8A is the narrow near infrared band, which is also within the red edge and effective for monitoring vegetation health information.; the band 9 is the water vapour band. The remaining 3 bands $10, 11$ and 12 are shortwave infrared bands.

Sentinel-2 Bands	Central Wavelength (µm)	Resolution (m)
Band 1 - Coastal aerosol	0.443	60
Band 2 - Blue	0.490	10
Band 3 - Green	0.560	10
Band 4 - Red	0.665	10
Band 5 - Vegetation Red Edge	0.705	20
Band 6 - Vegetation Red Edge	0.740	20
Band 7 - Vegetation Red Edge	0.783	20
Band 8 - NIR	0.842	10
Band 8A - Vegetation Red Edge	0.865	20
Band 9 - Water vapour	0.945	60
Band 10 - SWIR - Cirrus	1.375	60
Band 11 - SWIR	1.610	20
Band 12 - SWIR	2.190	20

Figure 1.11: Spectral bands for the Sentinel-2 sensors [8]

As a MSI, the 13 spectral bands produce 13 different grayscale images. We can produce a true colour remote sensing image based on the images of blue, green and red bands.

1.8 Normalised Difference Vegetation Index

Vegetation index is a method of remote sensing to monitor the growth and distribution of ground plants. Different green plants have different absorption rates of light of different wavelengths.

Normalized difference vegetation index (NDVI) [13] is a parameter that reflects nutritional information and crop growth. From this parameter, we can know the nitrogen demand of crops in various seasons. This plays an important guiding role in the rational application of nitrogen based fertilizer.

In the field of remote sensing applications, it is widely used to quantitatively evaluate vegetation coverage and its growth vitality. It is applied to detect vegetation growth status, vegetation coverage and eliminate some radiation errors. It utilizes the high reflectivity of the cell structure of plant leaves in the near-infrared band, and the strong absorption of chlorophyll in the red light band, which can turn multi-spectral data into a single image band for displaying vegetation distribution. A higher NDVI value indicates more green plants.

According to research findings [22], healthy plants are relatively green, with high chlorophyll content. Most of the red and blue light in sunlight is absorbed, while green light and near-infrared are largely reflected back. If the plant is unhealthy, it may die, and the chlorophyll content will decrease, it will also decrease when it matures, and it will look less green. Then the absorption capacity of green light is insufficient, the emission of red light increases, and it will absorb relatively more near-infrared light. Therefore, this relationship can be used to find a way to define the health of vegetation, which is NDVI.

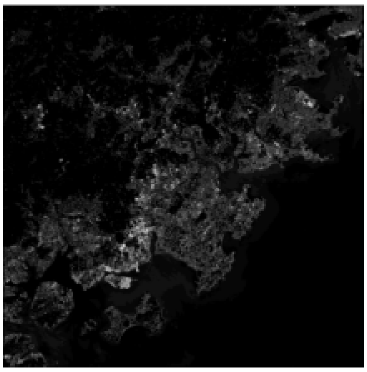
1.8.1 NDVI calculation

Normalized difference vegetation index can be calculated with the function [13]:

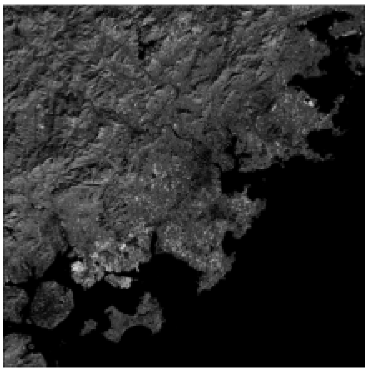
$$NDVI = \frac{NIR - R}{NIR + R}, \tag{1.1}$$

where NIR refers to the reflection value of the near infrared band, R refers to the reflection value of red light band. The result of this function should always be a value between -1 and $+1$, which avoids the inconvenience caused by data which is too small or too large.

If the reflection value of the red light band is low and the reflection value of the near infrared band is high, it will result in a high NDVI value and vice versa. Therefore, we can use NDVI as a standardized method for measuring healthy vegetation. When the NDVI value is positive, it shows that there is vegetation coverage, and the NDVI value increases as the coverage increases. When the NDVI value is negative, then the ground is covered with clouds, water, snow, etc. When NDVI is close to zero, NIR and R are approximately equal. In this case, there will be no green leaves, and it may even become an urbanized area.



(a) Band 4 (NIR)

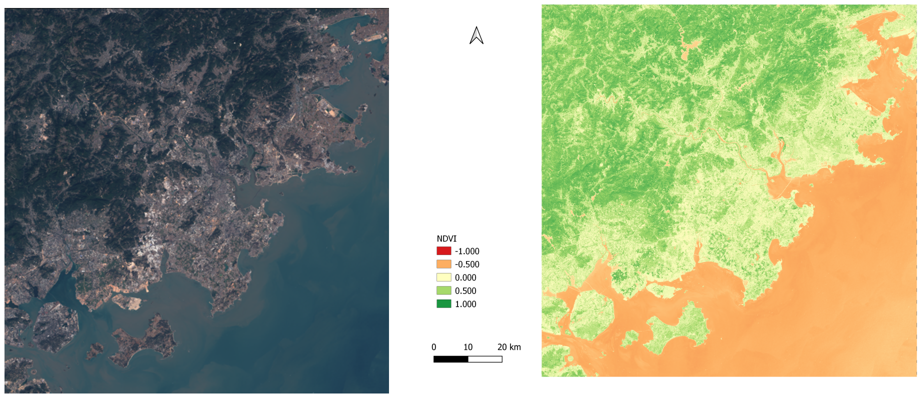


(b) Band 8 (Red)

Figure 1.12: NIR, Red bands images.

This NDVI image(Figure 1.13(b)) is produced based on the near infrared band image and red light band image (Figure 1.12), which are band 4 and band 8. In order to see how the near infrared band and red light band images are like, the contrast of the images are enhanced, since the images were nearly pure black.

In this NDVI image, according to the legend bar, the green parts represent the positive NDVI values, which shows the vegetation coverage corresponding to the vegetation area in the original image; the yellow parts represent the NDVI value



(a) Original [7] (b) NDVI

Figure 1.13: NDVI analysis.

of zero, which have no green leaves, it may be rock or soil, and this corresponds to the cities where do not have vegetation in the original image; and the orange parts represent the negative NDVI values, which is the area of the sea and corresponding to the sea in the original image.

Chapter 2

Clustering

Clustering [37] is a type of unsupervised learning, meaning that it does not use a training set of data to train a model. Instead it only uses the one set of data, and aims to find the natural groupings within this set of data. Unsupervised learning in general means only the original data set is to be analyzed, without any labels in advance.

Basically, clustering is to separate a data set into different clusters based on a certain criterion, so that the data objects in the same cluster is as similar as possible, and the data objects that are not in the same cluster is as different as possible. That is, after clustering, data of the same type are grouped together as much as possible, and different data are separated as much as possible.

There are many kinds of clustering algorithms. In this project, K-means clustering and Mean Shift clustering are introduced.

2.1 K-means clustering

K-Means clustering is a sort of an entry level clustering algorithm, which is a centroid-based clustering [24]. It uses distance to measure the similarity between data objects. Usually, Euclidean distance is used to calculate the distance between data objects:

$$d(x_i, y_j) = \sqrt{\sum_{d=1}^{D}(x_{i,d} - x_{j,d})^2}, \tag{2.1}$$

where D represents the number of attributes of the data object. The smaller distance between the objects gives us the higher similarity of the objects.

When processing K-means algorithm, we first need to determine a value of K and choose K centres of clusters randomly, which are the initialized centroids; then, we need to calculate the distance from all objects in the data set to the centroids with Euclidean distance equation, and the objects are assigned into the cluster which is closer to the centroids; after that, calculate the center point of each cluster as new centroids; and repeat the previous steps until the distances between the updated centroids and the corresponding original centroids are less than a certain set threshold.

Algorithm 1 K-Means algorithm [36]

Input: $D = \{d_1, d_2, \cdots, d_n\}$ // Set of n elements
K // Number of desired clusters
Output: A set of K clusters
Initialize K centroids;
repeat
Assign each element to the most similar cluster;
Calculate the new mean for each cluster;
until Convergence criteria is met

2.1.1 K-means clustering for colour image

A common method of performing image segmentation is to use a clustering algorithm. Each pixel in a colour image is a point in a three-dimensional space. The three dimensions are the intensity of the three primary colours of red, green, and blue. For image segmentation, the pixels are regarded as data points in clustering, and gather these data points according to the specified number of clusters. Then, replace each pixel with its corresponding cluster center to reconstruct the image.

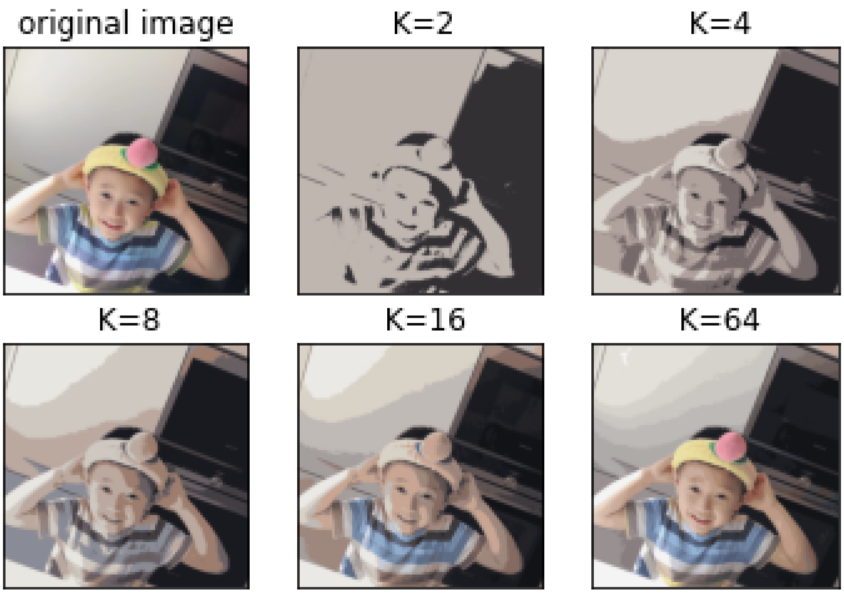


Figure 2.1: K-means on image

K-means clustering is applied on the image with different K values. By choosing different number of centroids, the image is clustered into different number of colours. When we set $K = 2$, we obtained an image with only 2 colours. When we set $K = 16$, we obtained an image with 16 colours. As can be seen in Figure 2.1, different number of clusters have different colour characteristics.

K-means clustering algorithm can also be used for map colour quantization. As we can see, the original map image has obvious colour difference at the splicing part. By using K-means algorithm, the colour of the image can be corrected by clustering the colours into 2 kinds. This method can be used to restore the colour of old maps.

(a) Original (b) Kmeans

Figure 2.2: Map colour quantization by K-means algorithm.

This map image is taken from the *National Library of Scotland* website[31]. There are a few obvious colours in this image: the black edges and the light colours background. Although the background has an obvious colour difference at the splicing part, the colours will still have similar intensity of the three primary colours of red, green, and blue. In K-means clustering we set $K = 2$, the pixels in the image are assigned to the cluster with the shortest colour distance. This colour distance can be calculated as follow:

$$d = \sqrt{(R - R')^2 + (G - G')^2 + (B - B')^2}. \tag{2.2}$$

Then, calculate the mean values of the colour intensity of the pixels in each cluster. Let the mean values be the new clusters. Keep doing this until there is no new cluster appears. Finally, the pixels of the background are assigned to the same cluster with the mean colour intensity of these background colours.

2.1.2 Advantages and disadvantages

The advantage of K-means algorithm is that it is very simple and fast. For processing large data sets, the algorithm is relatively applicable. The effect is better with resulting clusters which are dense and have obvious difference between them.

The most significant disadvantage of this algorithm is that we must specify the value of K before using the algorithm. However, in some situation, it is difficult to estimate the proper value of K. Using the iterative method, the results may only obtain local optimal solutions but not global optimal solution.

2.2 Mean shift clustering

Mean Shift [2] algorithm is also a common clustering algorithm, and it is very similar to K-means, both of which are constantly adjusting the position to approach the center of the region with the highest sample density. As one of the representative methods, mean shift clustering is based on pixel clustering and a feature space analysis method. Density estimation and mode search are the two core points of mean shift.

Mean shift algorithm is a sliding-window algorithm, which is also a centroid-based algorithm. Mean shift clustering is to find clustering points along the direction of increasing density, which uses a kernel function to estimate the density of the sample.

2.2.1 Kernel density estimation

The Gaussian kernel is the most commonly used kernel function that can be used to estimate the density of the sample. It is to set a kernel function for each sample point on the data set, and then add all the kernel functions to obtain the kernel density estimation of the data set.

Suppose we have a d-dimensional data set $\{x_i\}$ of size n, the bandwidth of the kernel function K is h, which gives us the function of the kernel density estimation[21] of the data set:

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K(\frac{x - x_i}{h}). \tag{2.3}$$

In here, $K(x)$ is the radially symmetric kernel:

$$K(x) = c_{k,d} k(\|x\|^2) \tag{2.4}$$

where $c_{k,d}$ is the normalization constant which makes the integral of $K(x)$ equal to 1.

2.2.2 Mean shift vector

For image data, the distribution has no fixed pattern to follow, so the density estimation must use non-parametric estimation, and the kernel density estimation with smoothing effect is selected.

The basic goal of the mean shift algorithm is to move the centroids in the direction where the local density increases, and the mean shift vector refers to the direction where the local density increases fastest. Therefore, the gradient direction of the data set density is the direction where the density increases fastest.

The gradient function[21]:

$$\begin{aligned} \nabla f(x) &= \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (x_i - x) g(\|\frac{x - x_i}{h}\|^2) \\ &= \frac{2c_{k,d}}{nh^{d+2}} [\sum_{i=1}^{n} g(\|\frac{x - x_i}{h}\|^2)][\frac{\sum_{i=1}^{n} x_i g(\|\frac{x-x_i}{h}\|^2)}{\sum_{i=1}^{n} g(\|\frac{x-x_i}{h}\|^2)} - x] \end{aligned} \tag{2.5}$$

where $g(s) = -k'(s)$ and the mean shift vector is:

$$m_h(x) = \frac{\sum_{i=1}^{n} x_i g(\|\frac{x-x_i}{h}\|^2)}{\sum_{i=1}^{n} g(\|\frac{x-x_i}{h}\|^2)} - x \tag{2.6}$$

2.2.3 Gaussian kernel

The core idea of the Gaussian kernel function is to map each sample point to an infinite-dimensional feature space, thereby making the originally linearly inseparable data linearly separable.

The Gaussian kernel function is[21]:

$$k(x, x') = \exp(-\frac{\|x - x'\|^2}{2\sigma^2}), \tag{2.7}$$

where σ controls the local scope of the function. When the Euclidean distance between x and x' is within a certain range, assume we have a fixed x', the Gaussian kernel function changes significantly as x changing.

The function can be rewritten in following form:

$$K(\frac{x - x'}{h}) = \frac{1}{\sqrt{2\pi}h} \exp(-\frac{(x - x')^2}{2h^2}), \tag{2.8}$$

where h is the bandwidth.

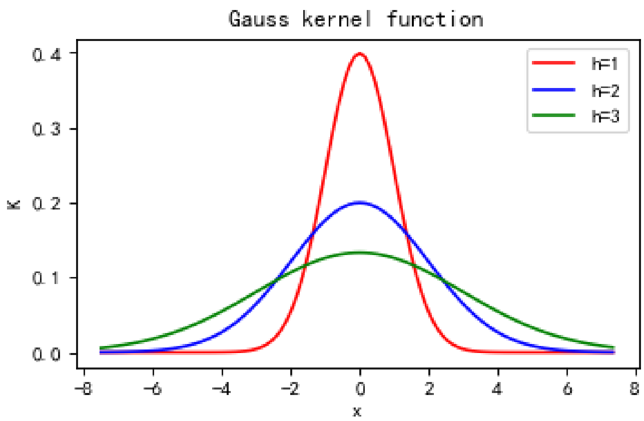


Figure 2.3: Gauss kernel function with different bandwidths

The Gauss kernel function has different images with different values of h. When h is determined, the closer the distance between the sample points, the greater the value of the kernel function; when the distances between the sample points are equal, as the bandwidth h of the Gaussian kernel function increases, the value of the kernel function decreases.

2.2.4 The Mean Shift Procedures

Procedures for mean shift algorithm:

1. Randomly choose a point from the unmarked data points as the centroid;

2. Find all points whose distance from the centroid is within the bandwidth, record them as set M, and consider these points belong to cluster c;

3. Calculate the vectors from the centroid to each element in the set M, add these vectors together to obtain the mean shift vector;
4. The centroid moves along the shift direction, and the moving distance is the modulus of the mean shift vector;
5. Repeat the steps 2, 3, and 4 until the size of the mean shift vector meets the threshold requirement, then record the centroid;
6. Repeat the steps 1, 2, 3, 4, 5 until all points are classified.

In contrast to K-means clustering, mean shift can determine the number of clusters by itself. Furthermore, mean shift clustering does not have a local minimum.

Mean shift algorithm have poor performance under high-dimensional data. Also, we can not choose the number of clusters for mean shift algorithm, however in some cases we need a specific number of clusters.

2.2.5 Mean Shift in Image Segementation

Mean shift can continuously segment the image to find the peak of the spatial colour distribution, and then merge the similar regions according to the peak to solve the problem of over-segmentation and obtain the final segmented image. Strictly speaking, mean shift is not to segment the image, but to smooth the image based on colours. It can neutralize colours with similar colour distribution, smooth colour details, and erode smaller colour areas.

During the process of performing mean shift on the image, the spatial window radius and the colour window radius are two key parameters. By changing these two radius, different results will be obtained.

(a) original (b) $sp = 20$, $sr = 20$ (c) $sp = 50$, $sr = 50$

Figure 2.4: Mean shift on image

To compare the results, mean shift algorithm is applied to the image with different spatial window radius and colour window radius. As can be seen, the image with small radius is more detailed, and the image with large radius has lost a lot of details. Therefore, the larger the value two radius, the more obvious the smoothing effect on the colour of the image, and hence the more time cost of the algorithm.

Chapter 3

Principal Component Analysis

Principal Component Analysis(PCA) is an unsupervised data analysis method [14]. The original data can be transformed into a set of linearly independent representations of each dimension through linear transformation by PCA algorithm. It is to extract the most valuable information based on the variance of the data, and can be used to extract the main feature components of the data, and for dimensionality reduction of high-dimensional data.

3.1 PCA algorithm

The procedure of PCA algorithm can be described with following algorithm box:

Algorithm 2 PCA algorithm

Input: Sample set $D = \{x_1, x_2, \cdots, x_m\}$; low-dimensional space dimension d'
Output: Projection matrix $W = (w_1, w_2, \cdots, w_{d'})$
  Compute the mean $x_i \leftarrow x_i - \frac{1}{m}\sum_{i=1}^{m} x_i^2$;
  Compute the covariance matrix of the samples $\frac{1}{m}XX^T$;
  Find the eigenvalues and corresponding eigenvectors of the covariance matrix;
  Take the eigenvectors corresponding to the largest d' eigenvalues $w_1, w_2, \cdots, w_{d'}$ to form the projection matrix M.

As shown in Figure 3.1, the white points are the sample points, projecting these points onto the green 'average' line gives us the blue points. For dimensionality reduction, the projections of the sample points should be as separate as possible.

The principle of PCA can be considered from two perspectives:

1. **Maximizing variance.** In one-dimensional space, variance can be used to express the degree of dispersion of data. The greater the variance, the more separate the sample points. The variance of the projection of each sample point on a dimension basis can be represented as follow:

$$Var(a) = \frac{1}{m}\sum_{i=1}^{m}(a_i - \mu)^2. \tag{3.1}$$

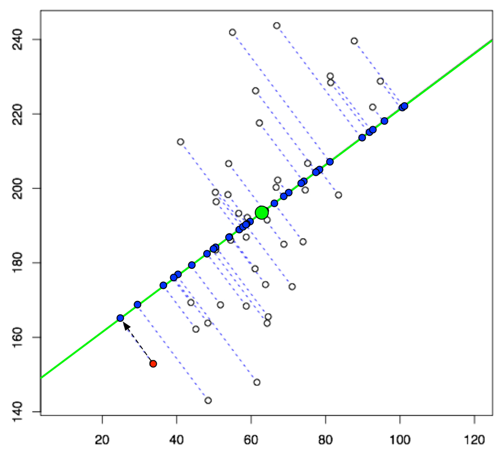


Figure 3.1: Sample points projection.[25]

In order to facilitate processing, the mean of each variable is reduced to 0, so the variance can be directly expressed by the sum of squares of each element divided by the number of elements:

$$Var(a) = \frac{1}{m}\sum_{i=1}^{m} a_i^2. \tag{3.2}$$

2. **Minimizing covariance.** For high-dimensional data, covariance can be used to express the correlation between two variables. In order to make the two variables represent as much original information as possible, we want to have no linear correlation between them. Therefore, the covariance between them should be minimized to obtain more original information. Same as before, reduce the mean of each variable to 0 and the covariance can be expressed as follow:

$$Cov(a, b) = \frac{1}{m}\sum_{i=1}^{m} a_i b_i. \tag{3.3}$$

Hence, when the mean value is 0, the covariance of the two variables is expressed as its inner product divided by the number of elements m.

3.2 Covariance matrix

In order to facilitate the calculation, we need to construct a covariance matrix. By using the covariance function, we can calculate the covariance matrix of the samples as follow:

$$\begin{aligned} Cov(X_i) &= \frac{1}{m}\sum_{i=1}^{m}(X_i - \bar{X_i})^2 \\ &= \frac{1}{m}\sum_{i=1}^{m}(X_i)^2 \\ &= \frac{1}{m}\sum_{i=1}^{m} X_i \cdot X_i^T \end{aligned} \tag{3.4}$$

Here is an example of how to reduce the dimensionality of a matrix. There is a 2×5 matrix, and we try to reduce it to 1×5. Suppose we have

$$X = \begin{pmatrix} -1 & -1 & 0 & 2 & 0 \\ -2 & 0 & 0 & 1 & 1 \end{pmatrix}, \tag{3.5}$$

since the matrix already has 0 as the mean for each row, we can now calculate the covariance matrix as follow:

$$C = \frac{1}{5}\begin{pmatrix} -1 & -1 & 0 & 2 & 0 \\ -2 & 0 & 0 & 1 & 1 \end{pmatrix}\begin{pmatrix} -1 & -2 \\ -1 & 0 \\ 0 & 0 \\ 2 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{6}{5} & \frac{4}{5} \\ \frac{4}{5} & \frac{6}{5} \end{pmatrix}. \tag{3.6}$$

Then, with this covariance matrix, we can find the eigenvalues $\lambda_1 = 2$, $\lambda_2 = \frac{2}{5}$ and corresponding eigenvectors $c_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $c_2 \begin{pmatrix} -1 \\ 1 \end{pmatrix}$ where c_1, c_2 are constants. Standardize the eigenvectors, and we get $v_1 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$ and $v_2 = \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$, which form a matrix $P = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$. Since we aim to reduce it to 1-dimensional, we take only the first row of the matrix P to calculate the dimensionality reduced matrix M:

$$\begin{aligned} M &= \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}\begin{pmatrix} -1 & -1 & 0 & 2 & 0 \\ -2 & 0 & 0 & 1 & 1 \end{pmatrix} \\ &= \begin{pmatrix} -\frac{3}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & \frac{3}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \end{aligned} \tag{3.7}$$

3.3 Singular Value Decomposition

PCA is the eigenvalue decomposition of covariance matrix, while SVD solves the PCA problem by singular values decomposition of covariance matrix. Suppose we have a covariance matrix A.

$$A = U\Sigma V^T, \tag{3.8}$$

where U and V are orthogonal matrices, Σ is diagonal matrix.

Singular value decomposition is to decompose the matrix A into the diagonal matrix Σ, and the diagonal element σ_i in Σ is called the singular value of matrix A. The key point of singular value decomposition is to perform eigenvalue decomposition on $A^T A$.

$$\begin{aligned} A^T A &= (U\Sigma V^T)^T U\Sigma V^T \\ &= V\Sigma^T U^T U\Sigma V^T \\ &= V\Sigma^T \Sigma V^T \\ &= V\Sigma^2 V^T \end{aligned} \tag{3.9}$$

The singular value σ_i of A is the square root of the eigenvalue λ_i of $A^T A$. If the eigenvector of $A^T A$ is v_i, then we can get the corresponding u_i of U:

$$u_i = \frac{Av_i}{\sigma_i}. \tag{3.10}$$

If we take $A = \frac{X^T}{\sqrt{m}}$, then we have

$$A^T A = (\frac{X^T}{\sqrt{m}})^T \frac{X^T}{\sqrt{m}} = \frac{1}{m} X X^T, \tag{3.11}$$

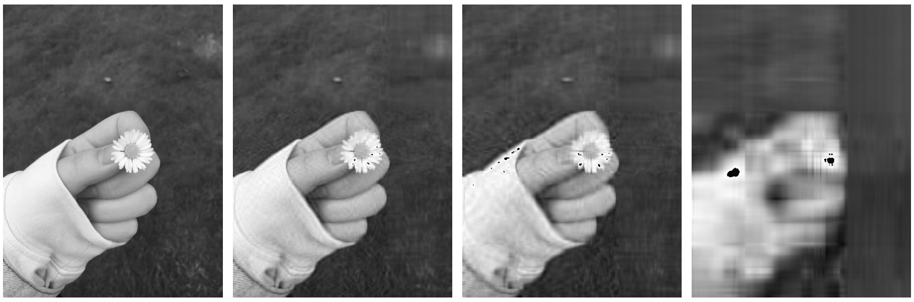
which equals to the equation of covariance matrix, so the PCA problem can be transformed into an SVD problem in this way.

3.4 Dimensionality reduction

Dimensionality reduction is necessary for exploring high-dimensional data [19]. Although part of the data is discarded while we reduced the dimensionality, the sampling density of samples can be increased after discarding this part of information, which is an important motivation for dimensionality reduction. Moreover, the eigenvectors corresponding to the smallest eigenvalues are often related to the noise which might affect the data. Discarding these smallest eigenvalues can have a denoising effect to some extent.

PCA is used to reduce the image dimensionality so that it stores only important pixels to retain the original image characteristics, and more effective in storage.

The dimensionality of the original image on Figure 3.2 (a) is 1080. By applying the PCA algorithm in Python with this image, the dimensionality reduced to 43 when we choose to keep 99% information, which is shown in Figure 3.2 (b). The dimensionality reduced to 10, when we keep 98% information, as shown in Figure 3.2 (c). Finally, we try to keep 90% information, and the dimensionality has reduced to 5, which is shown in Figure 3.2 (d). Comparing these images,



(a) Original (b) PCA(99%) (c) PCA(98%) (d) PCA(90%)

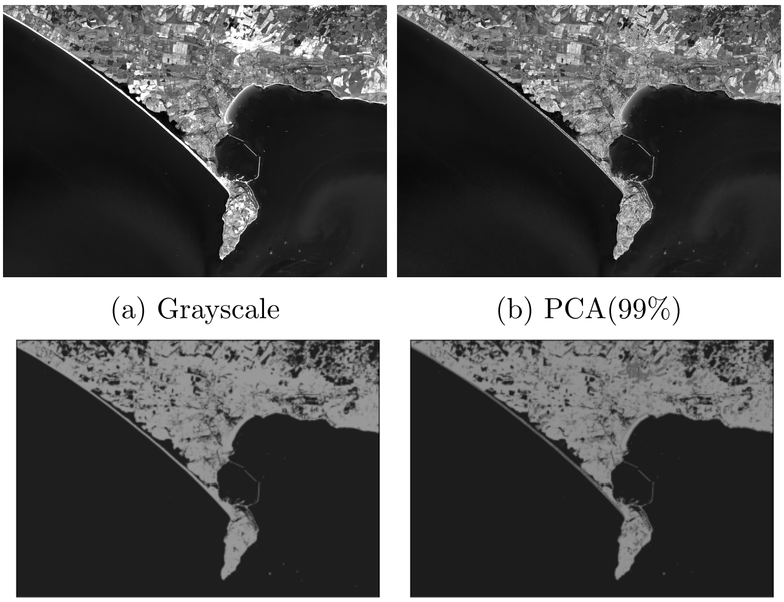
Figure 3.2: PCA dimensionality reduction

we can see that the less distortion of the reconstructed image, the higher the dimensionality we obtained after the PCA dimensionality reduction.

3.5 Dimensionality reduction before clustering

I captured a small part in a satellite image, and applied the PCA dimensionality reduction on the image. This process reduced the dimensionality from 3012 to 357, and 99% retained information.

By applying the K-means clustering algorithm to assign the pixels into 2 clusters on both the images before and after dimensionality reduction, we found that the time cost for the dimensionality reduced image is less than the original image and the images obtained are almost the same.



(a) Grayscale (b) PCA(99%)

(c) K=2 (d) K=2(PCA)

Figure 3.3: PCA dimensionality reduction on clustering

From the above images, we found that using PCA to reduce the dimensionality will make the algorithm run faster, since the complexity of the algorithm is reduced. Hence, PCA can be used as a preprocessing method for image processing, since lowering the dimensionality of an image makes the application of the next technique much simpler.

Chapter 4

Image Segmentation Techniques

Image segmentation techniques [26] segment images into regions based on features such as grayscale, color, texture, and shape, so that the differences between regions appear and the pixels which belong to the same region are similar. The commonly used segmentation techniques are:

- Threshold-based technique
- Edge detection technique
- Region-based technique
- Watershed transformation technique
- Active contour model

4.1 Threshold-based technique

Threshold-based segmentation is a traditional image segmentation method [30]. It is the most basic and most widely used segmentation technology in the field of image segmentation. Also, it is especially suitable for images where there are different gray scale ranges for target and background.

An image includes target, background and noise. A certain threshold T is set to divide the image into two parts: a group of pixels with values larger than T and a group of pixels with values smaller than T. The difficulty is how to choose an appropriate threshold T to achieve better segmentation.

There are several methods we can use to find the appropriate threshold T. I found the certain threshold T with Otsu's method, and produced the following segmented image with the threshold-based technique.

4.1.1 Otsu's method

Otsu's method is also called between-class variance method. It is derived by using the principle of least squares on the basis of the gray value histogram, and has the best segmentation threshold in the statistical sense.

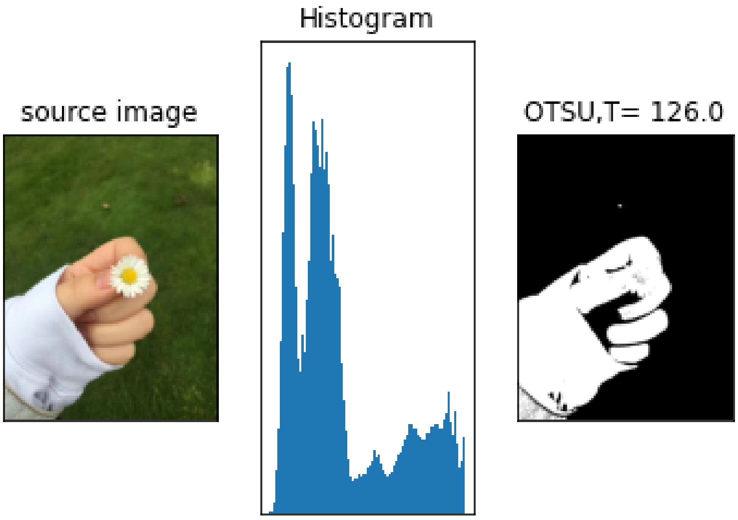


Figure 4.1: Threshold-based technique.

Suppose the image contains L graycale levels, the number of pixels with gray value i is N_i, and the total number of pixels is:

$$N = N_0 + N_1 + \cdots + N_{L-1}. \tag{4.1}$$

Then the probability of the point whose gray value is i is:

$$P_i = \frac{N_i}{N}. \tag{4.2}$$

According to the expectation equation, the mean of gray value is:

$$\mu_T = \sum_{i=0}^{L-1} i P_i. \tag{4.3}$$

According to the grayscale characteristics of the image, the image is divided into target c_0 and background c_1 with threshold value T. Then $\omega_0(T)$ and $\omega_1(T)$ respectively represent the probability of occurrence of c_0 and c_1 when the threshold is T [11]:

$$\begin{aligned} \omega_0(T) &= \sum_{i=0}^{T} P_i \\ \omega_1(T) &= 1 - \omega_0(T). \end{aligned} \tag{4.4}$$

The mean values of c_0 and c_1 are:

$$\begin{aligned} \mu_0(T) &= \frac{\sum_{i=0}^{T} i P_i}{\omega_0(T)} \\ \mu_1(T) &= \frac{\mu_T - \sum_{i=0}^{T} i P_i}{\omega_1(T)} \end{aligned} \tag{4.5}$$

The between-class variance with threshold value T in the histogram is defined as:

$$\sigma_B^2(T) = \omega_0(T)[\mu_0(T) - \mu_T]^2 + \omega_1(T)[\mu_1(T) - \mu_T]^2 \tag{4.6}$$

The optimal threshold is defined as the T value corresponding to the maximum variance between classes:

$$\sigma_B^2(T^*) = \max_{0 \leq T \leq L-1} \sigma_B^2(T). \tag{4.7}$$

According to the gray characteristics of the image, the image is segmented into two parts, the background and the target. The larger inter-class variance between them will give us the greater difference between these two parts of the image. When part of the background is incorrectly classified as the target or part of the target is incorrectly classified as the background, the difference between the target and background will become smaller. Therefore, the probability of misclassification will be minimized by the segmentation that maximizes the between-class variance.

In summary, threshold-based segmentation is a simple calculation method with high efficiency; but since it does not consider the spatial characteristics and only considers the characteristics of the gray value of the pixel itself, it is more sensitive to noise.

4.2 Edge detection techniques

Edge detection is an important way of image segmentation [16]. The edge of the image means the end of one region and the beginning of another region. The collection of pixels between adjacent regions constitutes the edge of the image. Therefore, the edge of the image, which has two elements: direction and amplitude, can be understood as a collection of pixels with spatial mutations in the grayscale of the image. According to the characteristics of the mutations, first-order and second-order derivatives can be used to detect the edges in the image. The edge detection is to first determine the pixels of the edges in the image, and then connect these pixels together to form the desired area boundary.

Edge detection technology can usually be divided into serial edge detection and parallel edge detection according to the processing technology. Serial edge detection is to determine whether the current pixel is a point on the detection edge, which depends on the verification result of the previous pixel. Parallel edge detection is whether a pixel belongs to the noble point of detecting edge, which depends on the pixel currently being detected and its neighboring pixels.

The simplest edge detection method is the differential operator method, which uses the discontinuous nature of the pixel values of adjacent areas and the first or second derivative to detect edge points.

4.2.1 Differential operators

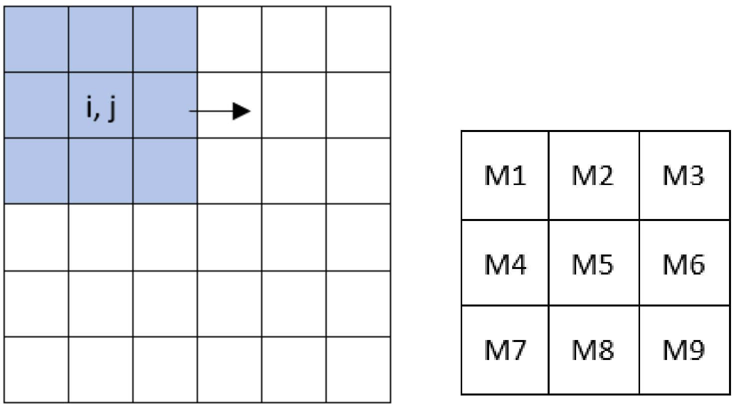
There are some commonly used first-order differential operators [12], Roberts operator, Prewitt operator and Sobel operator, and second-order differential operators, Laplace operator and Kirsch operator. Various differential operators are often represented by small-area templates, and differential operations are achieved by using templates and image convolution. These operators are sensitive to noise and only suitable for images with low noise and not complex images. Since the edges and noise are both grayscale discontinuous points, it is difficult to overcome the effect of noise by directly using differential operations. Therefore, the image must be smoothed and filtered before the edge is detected by the differential operator.

Since the pixels and noise in the direction of the vertical edges are grayscale discontinuous points, when transforming to the frequency domain, they are all high-frequency components in the frequency domain, and directly use differential operation will inevitably be affected by the noise. Therefore, the differential operators are only suitable for simple images with less noise. To solve this problem, the method adopted by the Laplace of Gaussian operator and the Canny detector is to smooth and filter the image first, and then use the differential operator to convolve the image, so that it will give a better edge detection result. Among them, the Laplacian of Gaussian operator uses the Laplace operator to calculate the second derivative of the Gaussian function, and the Canny detector is the first derivative of the Gaussian function. The two operators have achieved a relatively good balance between noise suppression and edge detection.

4.2.2 Filters and convolutions.

Filtering and convolution are commonly used operations in image processing. They are similar in principle, but there are some differences in details.

The filtering operation is the sum of the products of the corresponding pixels of the image and the mask.



(a) image (b) mask

Figure 4.2: Filtering

Then, the following equation can be used to calculate the filtered result of pixel (i, j):

$$\begin{aligned} G(i,j) = & I(i-1,j-1) \times M1 + I(i,j) \times M2 \\ & + I(i+1,j-1) \times M3 + I(i-1,j) \times M4 \\ & + I(i,j) \times M5 + I(i+1,j) \times M6 \\ & + I(i-1,j+1) \times M7 + I(i,j+1) \times M8 \\ & + I(i+1,j+1) \times M9. \end{aligned} \tag{4.8}$$

Slide the mask across the entire image and calculate the filtering result of each pixel in the image. The size of image remains unchanged after filtered. This is what we do for image filtering.

There are two main purposes for image filtering:

1. Extract image features through filtering, simplify the information in the image for subsequent image processing.

2. Eliminate the noise in the image to meet the needs of image processing.

The first point is used in edge detection, which is to simplify image information and use edges to represent the information in the image.

In the filed of image processing, convolution can be used for image blurring, sharpening and edge detecting.

In mathematics, convolution is an operator which can be represented by the following function:

$$(f \star g)(t) := \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau. \tag{4.9}$$

Convolution generates a new image by using the source image (f) and the convolution kernel (g), which is similar to filtering. The convolution kernel is used to slide on the image, multiply the gray value of the pixel on the image with the value on the corresponding convolution kernel. Then all the multiplied values are added together as the gray value of the pixel on the image corresponding to the centre pixel of the convolution kernel, and finally the process of sliding through all the images is completed. Different from image filtering, convolution operation will change the size of the image.

4.2.3 Comparison of different filters.

The edge detection algorithm is mainly based on the first and second derivatives of image intensity, but the derivative is usually very sensitive to noise, so it is necessary to use filters to remove noise, process the image with threshold algorithms, and finally do the edge detection. The following is a comparison of 5 common edge detection algorithms after denoised with Gauss filtering and thresholding.

From Figure 4.3, we can see that the Canny detector is the best operator for edge detection.

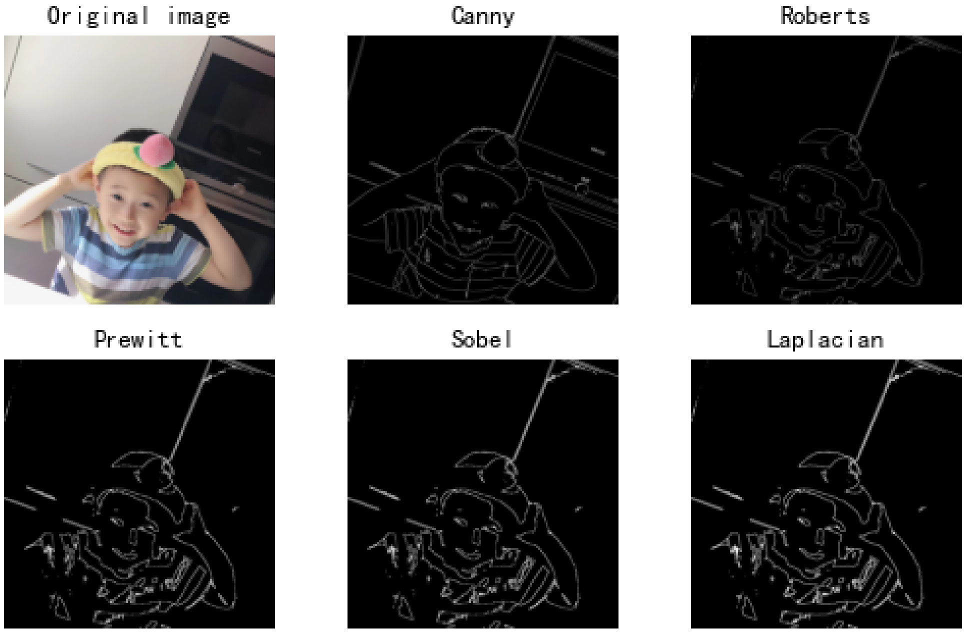


Figure 4.3: Comparison of different filters.

4.2.4 Canny detector

Canny detector is considered by many people to be the optimal algorithm for edge detection, because it can identify more actual edges while minimizing error caused by noise; the edge identified by the canny operator is closer to the actual edge in the image than the other detectors; the edges in the image are only identified once in Canny detector.

Canny detector is an edge operator that has been widely used in digital image processing in recent years. Through Canny detector, we can calculate the edge strength and edge gradient direction of the digital image to provide a basis for the subsequent edge point judgment. Canny detector uses functional derivation method to derive the first derivative of Gaussian function, which is the best approximation of the optimal edge detection operator. Since the convolution operations can be exchanged and combined, the Canny algorithm first uses a two-dimensional Gaussian function to smooth the image.

Procedures for Canny detector [3]:

1. **Noise removal.**
   Edge detection of images is sensitive to noise. Therefore, before edge detection, noise removal is required. Gaussian filtering is usually used to remove noise, which can filter out the noise part of the image, avoid the wrong noise information being mistakenly recognized as an edge when performing edge detection later. The Gauss function[20] is:

$$G(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{4.10}$$

2. **Find the gradient of the image.**

Algorithm 3 Canny Edge Detection [3]

Input: Image I, threshold T_l, T_h.
Output: Canny edge image E, gradient strength G, gradient direction θ.

Remove the noise of the image with Gauss filter
Compute gradients
for each pixel (x, y) **do**
  $G_x(x, y) = \frac{1}{2}I(x-1, y) - I(x, y) + \frac{1}{2}I(x+1, y)$
  $G_y(x, y) = \frac{1}{2}I(x, y-1) - I(x, y) + \frac{1}{2}I(x, y+1)$
  $G(x, y) = \sqrt{G_x^2 + G_y^2}$
  $\theta = tan^{-1}(G_y/G_x)$
end for
Non-maximum suppression
for each pixel (x, y) **do**
  Find 2 neighbours in direction of gradient θ
  if $G(x, y) \leq G(\text{any neighbour})$ **then**
    $G(x, y) \leftarrow 0$
  end if
end for
Double threshold
for each pixel (x, y) **do**
  if $G(x, y) \geq T_h$ **then**
    $E(x, y) \leftarrow 1$
  else
    $E(x, y) \leftarrow 0$
  end if
end for
Recursively find non-edge pixels with $G \geq T_l$ and with an edge pixel as a neighbour. Mark them as edge pixels as well.

Canny detector generally uses the kernel of other operators to calculate the gradient strength and direction, such as Roberts or Sobel. Suppose we have the x and y direction kernels as follow[28]:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}. \tag{4.11}$$

Then the gradient strength G and the gradient direction θ can be calculated as follow:

$$G = \sqrt{G_x^2 + G_y^2}, \tag{4.12}$$

$$\theta = \arctan(\frac{G_y}{G_x}). \tag{4.13}$$

3. **Non-maximum suppression.**
   Non-maximum suppression is done by finding the pixel with the maximum value in the edge, and suppress all other pixels.

4. **Double threshold.**
   Take a high threshold and a low threshold. If the gray value of the pixel is greater than the high threshold, we mark it as a strong edge pixel. If the gray value of the pixel is greater than the low threshold and less than the high threshold, we mark it as weak edge pixel. If the gray value of the pixel is less than the low threshold, we do not mark it as edge pixel.

5. **Edge tracking by hysteresis.**
   From the previous step, the strong edge pixels should all be contained in the resulting edge image. However, we can not make sure that all the weak edge pixels are contained in the resulting edge image. To decide which weak edge pixel are contained in the edge image, the 8 adjacent pixels should be considered. If any of the 8 pixels is a strong edge pixel, then the weak edge pixel should also be contained in the resulting edge image.

4.2.5 Edge-based segmentation

The Edge-based segmentation has accurate edge positioning. However, it cannot guarantee the continuity and closure of the edges. There are a lot of discontinued edges in the high-detail area, it is difficult to form a large area, and it is not suitable to divide the high-detail area into small fragments. Therefore, rather than generate a complete image segmentation process, edge detection can only generate edge points. So after the edge point information is obtained, subsequent processing or a combination of other related algorithms are needed to achieve the segmentation purpose.

Procedure of edge-based segmentation:

1. Use differential operator to determine the edges of objects.

2. Use mathematical morphology to fill the contours.

3. Remove small spurious objects.

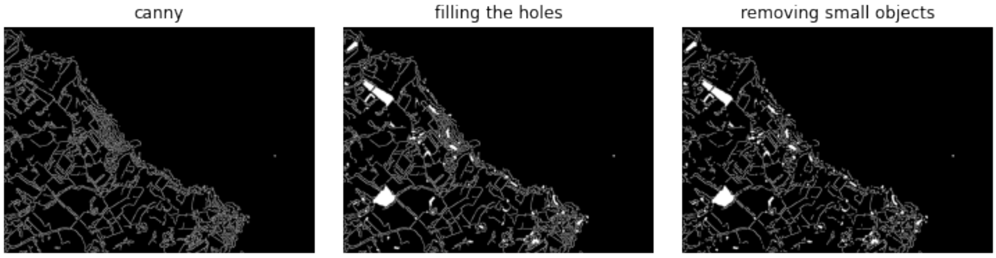


Figure 4.4: Edge-based segmentation

In Figure 4.4, I captured part of the satellite image and used the canny detector to determine the edges. In the second step, the image is obviously not filled correctly, and that is because the contours which are not completely closed can not be filled correctly.

4.3 Watershed transformation technique

The watershed transform can be classified as a region-based segmentation approach [33]. It is a segmentation method based on the mathematical morphology of topological theory, which is to regard the image as a topological topography. The gray value of each pixel in the image represents the altitude of the point. High gray values correspond to mountain peaks, and low gray values correspond to valleys.

If the water flows down, it will flow towards the valleys, and will not stop until a local lowest place. This local lowest area is called the catchment basin. Eventually all the water will be collected in different catchment basins. The ridge between the catch basins is called the watershed, which is the edge of the image corresponding to image segmentation(Figure 4.6). In image segmentation, we aim to find different catchment basins and watersheds in the grayscale image. The area composed of different catchment basins and watersheds is the target we want to segment.

The procedure of the watershed algorithm:

1. Classify all the pixels in the gradient image according to the gray value, and set a geodesic distance threshold.

2. Find the pixel with the smallest gray value. Make these points as the starting points, and the threshold increased from the minimum.

3. As the water level grows, it will reach surrounding neighboring pixels. Measure the geodetic distance from these pixels to the starting point. If it is less than the set threshold, these pixels will be submerged. Otherwise, a dam on these pixels should be set to classify these neighbouring pixels.

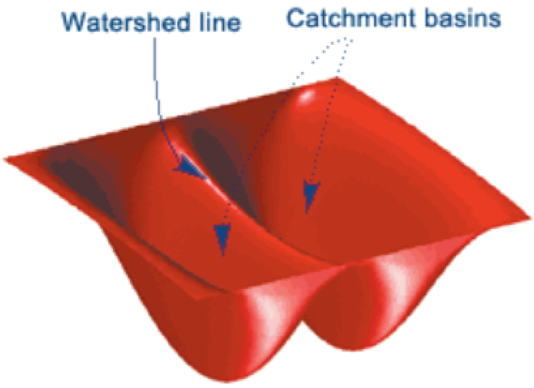


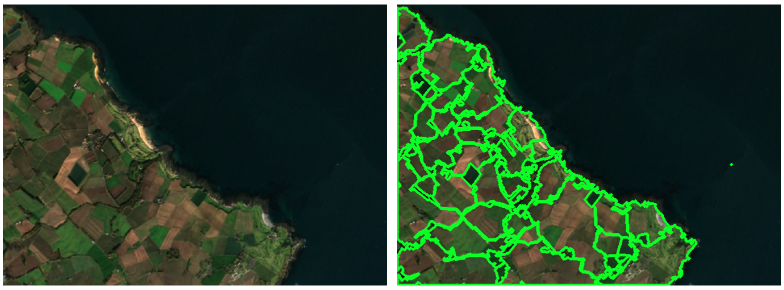
Figure 4.5: Watershed transformation[6]

Generally, gradient images have higher pixel values at the edges and lower pixel values in other places. Ideally, the mountain ranges are just at the edges. Therefore, we can find the mountain range based on the gradient.

4.3.1 Watershed transformation segmentation

Watershed transformation is used in many areas of image processing due to its following advantages: intuitive, fast and can be calculated in parallel, and always generates a complete boundary to avoid later process of boundary connection.

Watershed transformation technique is applied on a small part of a satellite image to segment the image into different regions.



(a) Original (b) Segmented

Figure 4.6: Watershed segmentation.

From the result shown above, we can observe that the watershed algorithm still has some disadvantages:

1. Over-segmentation. Due to the interference of noise points or other factors, dense small areas may be obtained, that is, over-segmentation. This is because there are many local minimum points in the image, and each point will form a small area by itself.

2. Sensitivity to noise. Some local changes will cause obvious changes in the segmentation results. Strong noise sometimes makes the watershed transformation unable to find the true boundary.

3. Difficult to accurately detect low-contrast boundaries. The low contrast makes the noise relatively high. So due to the previous reason, the watershed transformation still cannot work quite well for this kind of picture.

There are two methods can be adopted to eliminate the over-segmentation problem. First one is that we can use the prior knowledge of the image to remove irrelevant edge information. The second one is that the gradient function can be modified so that the collection basin only responds to the desired target. To reduce the over-segmentation, the gradient function is usually modified. A simple method that normally used is to threshold the gradient image, which eliminates the over-segmentation problem caused by some tiny changes in grayscale.

4.4 Region-based technique

The region-based segmentation method is a segmentation technology based on directly finding the region [17], which has two basic forms: one is region growth, which starts from a single pixel and gradually merges to form the required segmentation region; the other starts from the overall situation and gradually cuts to the required segmentation area.

4.4.1 Region growing

Region growing is to group similar pixels to form regions. First, initialize a seed pixel for each desired region as the starting point and the growing process start from these seed pixels. Then, merge the neighbouring pixels around the seed pixel that have high similarity as the seed into the region where the seed pixel is located. New pixels continue to grow around as seeds, until no more pixels that meet the conditions can be included, and a region is grown.

The procedure of region growing is:

1. Randomly select a pixel in the image as the initial seed pixel;

2. Search for unmarked pixels near the seed. Then merge the pixels into the segmentation area, if its difference is within the specified threshold. The threshold needs to be set according to different situations;

3. Repeat step 2 until the area stops expanding, and at this time randomly select a pixel in the non-selected area again as the seed pixel;

4. Repeat the above steps until each pixel in the image is assigned to a region.

There are three problems that the region growing algorithm needs to solve:

1. Choose a group of seed pixels that can correctly represent the desired area;

2. Determine the criteria for including neighboring pixels in the growth process;

3. Specify the conditions or rules to stop the growing process.

(a) Original (b) Segmented

Figure 4.7: Region growing

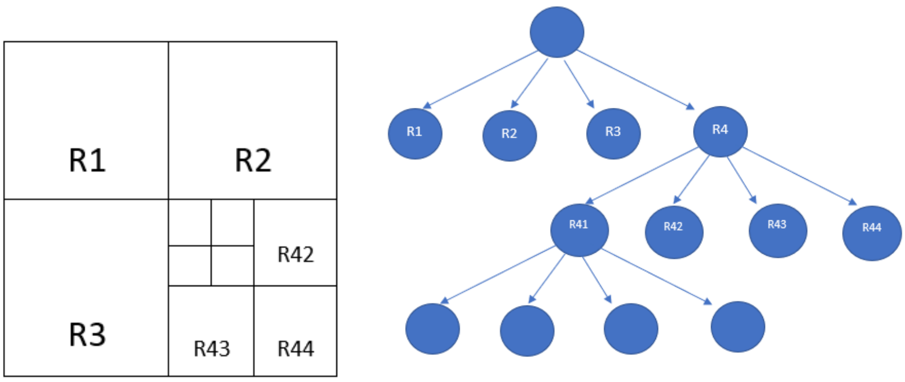
The key point of this method is to choose suitable the initial seed pixels and reasonable growth criteria. The region growing is simple and especially suitable for segmenting small structures, such as tumors and scars.

Region growing method is applied on the image shown above, the result we observed is that it is sensitive to noise, therefore resulting in holes (Figure 4.7) in the extracted area.

4.4.2 Region splitting and merging

Another method region splitting and merging, which is to divide the image into a series of arbitrary disjoint regions at the beginning, and then merge or split them to meet the constraints. By splitting, regions with different characteristics can be divided into four quadrants; and by merging, regions with the same characteristics can be merged.

This splitting process can be expressed in the form of a quadtree:



(a) Region splitting (b) Quadtree

Figure 4.8: Quadtree

The procedure of region splitting and merging is:

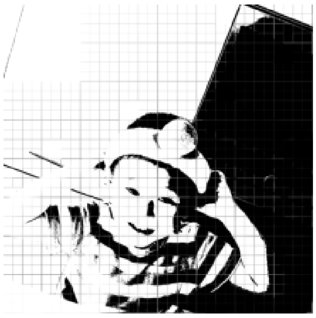
1. Label all pixels in the image separately.

2. For the area identified by the same label, if the pixel grayscale value is not in the same range, divide the area into four quadrants.

3. Repeat the above steps until all sub-regions have similar grayscale value.

4. Compare adjacent sub-regions and merge similar regions until all sub-regions have no adjacent similar regions.

(a) Original



(b) Segmented

Figure 4.9: Region splitting and merging

The region splitting and merging (Figure 4.9) method is suitable for complex images. However, the algorithm is complicated and has a large amount of calculations, and splitting may break the boundary of the regions.

The difficulty of this algorithm is that the splitting and merging criteria are not easy to decide. The splitting criterion is also called the uniformity test criterion, which is used to judge whether the block image needs to be split.

At the beginning, we used whether the difference between the maximum and minimum gray values in each image area was within the allowable deviation range as the uniformity test criterion. Then, by the continuous development of the uniformity test criteria, the statistical tests, such as minimum mean square error and F detection, are now the most commonly used uniformity test criteria.

4.4.3 Region-based method using the watershed transformation

The watershed transformation can also be used on region-based segmentation. The procedure of region-based method using the watershed transformation is:

1. Produce a histogram of grey values.

2. Use an appropriate filter to determine the elevation map of the image.

3. Find two extreme parts of the histogram of gray values, use these two extreme parts to determine the markers of foreground and background.

4. Fill the region of the elevation map starting from the markers determined above with Watershed transformation.

This technique is applied on a satellite image(Figure 4.10). First, a histogram of gray values is produced for the image. Then, Sobel filter is used to determine the elevation map of the image. After that, two extreme parts of the histogram of gray values 0.1, 0.3 are found, and these two values are used to determine the markers of foreground and background. Finally, filled the region of the elevation map starting from those markers with watershed transform.

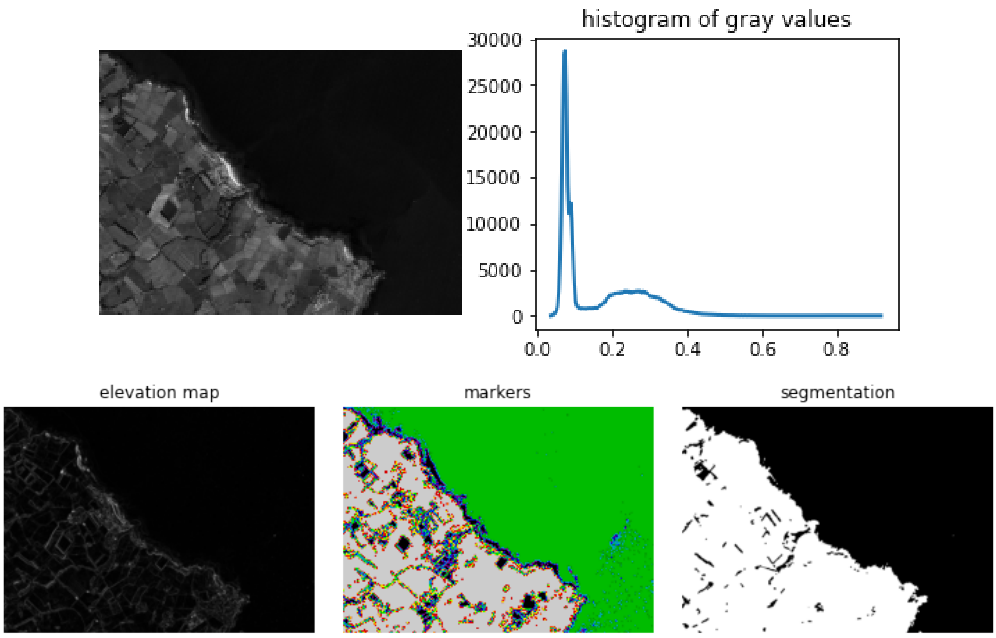


Figure 4.10: Region-based method using watershed transformation

4.5 Active contour model

Active contour model is a segmentation method based on energy functional [18]. It is to use a continuous curve to express the contour of the target, and define an energy functional so that its independent variables include the edge curve. Therefore, the segmentation process is transformed into a process of solving the minimum value of the energy functional. Generally, it can be achieved by solving the Euler, Lagrange functions. When the energy reaches the minimum, the curve position is the contour of the target.

According to the different expressions of curves in the model, active contour models can be divided into two types: **parametric active contour model** and **geometric active contour model**.

4.5.1 Parametric active contour model

The parametric active contour model is based on the Lagrange framework and directly expresses the curve in a parametric form of the curve. The most representative one is the **Snakes model**.

Snakes model [34] is to first provide the position of an initial contour of the image to be segmented, and define an energy function for it, so that the contour approaches the target in the direction of energy reduction. When the energy function reaches the minimum, the provided initial contour converges to the true contour of the target in the image.

The Snakes energy function is composed of an internal energy function and an external energy function. The internal energy controls the smoothness and continuity of the contour. The external energy consists of image energy and constrained energy, which controls the contour converges to the actual contour. The constrained energy can be based on the specific object, which makes the Snakes model have great flexibility.

The Snakes model is widely used in the fields of digital image analysis and computer vision. Since the Snakes model has a good ability to extract and track the contour of a target in a specific area, it is very suitable for the processing of medical images such as CT and MR images to obtain the contours of specific organs and tissues.

Define the Snakes model[27] as an deformable curve:

$$V(s) = \{X(s), Y(s)\}, \tag{4.14}$$

where s is the normalized curve length, ranging from 0 to 1. The energy function of the Snakes model can be represented as follow:

$$E_{Snakes} = E_{internal} - E_{external}, \tag{4.15}$$

where $E_{internal}$ is the internal energy of the contour which produced by controlling the stretching and preventing contour discontinuity, $E_{external}$ moves the Snakes model towards a image feature, attracting the Snakes model to move to the feature of interest. These energy functions can be represented as follow:

$$E_{int} = \frac{1}{2}\int_0^1 [\alpha(|\frac{dX}{ds}|^2 + |\frac{dY}{ds}|^2) + \beta(|\frac{d^2X}{ds^2}|^2 + |\frac{d^2Y}{ds^2}|^2)]ds \tag{4.16}$$

$$E_{ext} = \int_0^1 (f[X(s), Y(s)])ds, \tag{4.17}$$

$$f(x, y) = -|\nabla G_\sigma(x, y) \times I(x, y)|, \tag{4.18}$$

where $I(x, y)$ is a gray-level image and $G_\sigma(x, y)$ is a Gaussian function.

The Snakes model has some advantages that classic methods do not have: image data, initial estimation, target contour, knowledge-based constraints are unified in one process. After proper initialization, it can autonomously converge to the minimum energy state; scale minimizing energy from the first to the finest in space can greatly expand the capture area and reduce complexity.

At the same time, the Snakes model also has its disadvantages: it is sensitive to the initial position and needs to rely on other mechanisms to place Snakes near the image features of interest. Also, due to the non-convexity of the Snakes model, it may converge to a local extreme point or even diverge .

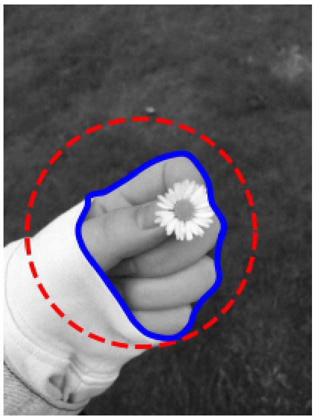
4.5.2 Geometric active contour model

The geometric active contour model [38] is based on the curve evolution theory and the level set method. It expresses the low-dimensional evolution curve or surface through a high-dimensional function surface, and expresses the evolved curve or surface as an indirect expression of the zero level set of the high-dimensional function surface form. The evolution equation of the curve or surface is transformed into the evolution partial differential equation of the high-dimensional level set function to avoid the parameterization process of the deformed curve or surface.

4.5.3 Active contour segmentation

By using active contour model, we can get a contour around the target to be segmented. Here is an example to segment my hand from the rest of the image.

(a) original image



(b) active contour model

Figure 4.11: Active Contour(Snakes) Model

After proper initialization, 'Snakes' model can autonomously converge to the minimum energy state.

In the image above, the 'Snakes' model is used in Python to segment my hand from the rest of the image, the image is segmented by fitting the closed curve to my hand. First, the red dotted circle is initialized to provide the position of an initial contour of my hand, then an energy function is defined for it. The contour approaches my hand in the direction of energy reduction, and it autonomously converges to the true contour of my hand when the energy function reaches the minimum. As can be seen in Figure 4.11(b), my hand is perfectly segmented from the rest of the image by the blue contour.

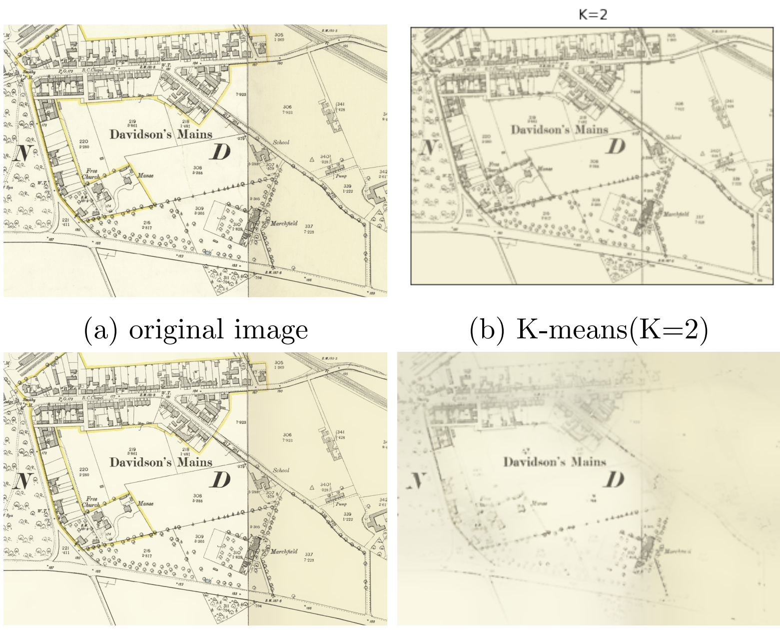
However, active contours model is sensitive to the initial contour. It relies on other mechanisms to place the initial contour around the desired target area. And it is unable to converge to the concave point on the boundary.

Chapter 5

Conclusion

In this project, clustering methods, principal component analysis and image segmentation techniques have been studied.

Principal Component Analysis (PCA) is a very effective method to simplify and accelerate the further processing of images. In this project, the PCA algorithm is used to reduce the dimensionality of images before applying some of the techniques, saving time for processing the images while preserving a large amount of the original information of images.



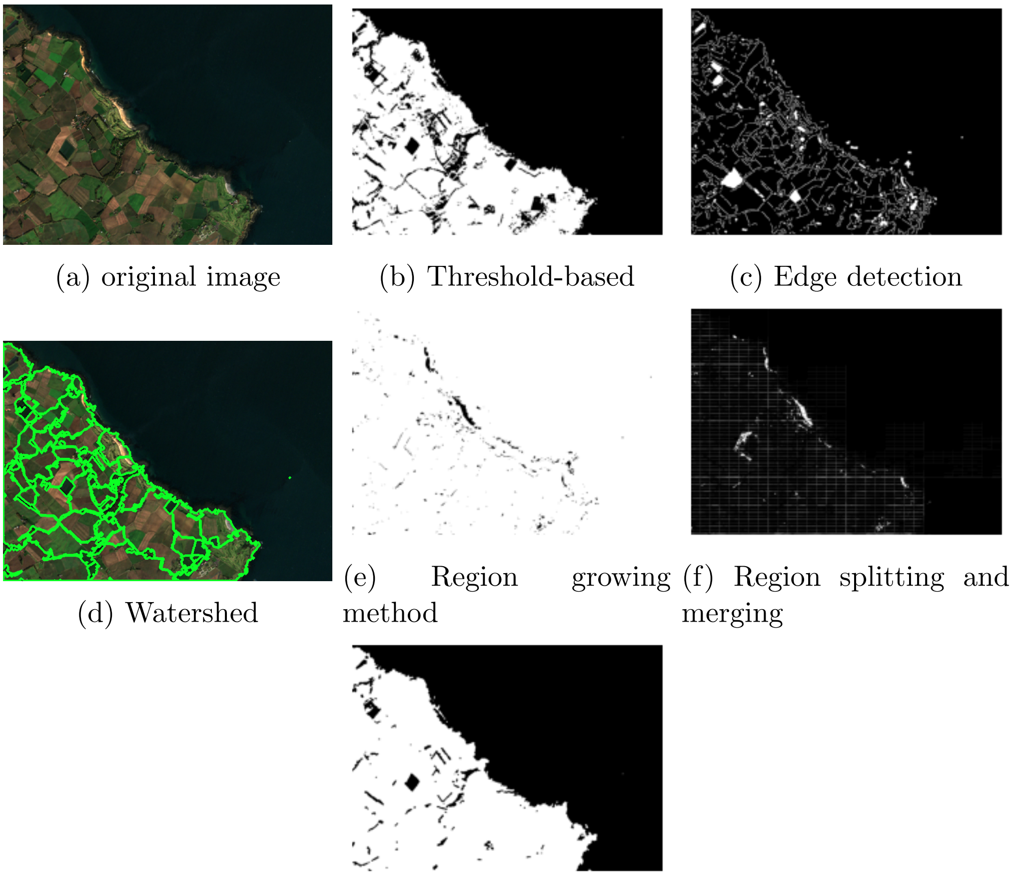
(a) original image (b) K-means(K=2)

(c) Mean shift(small) (d) Mean shift(large)

Figure 5.1: Comparison of K-means & Mean shift

In Figure 5.1, both clustering methods are applied on the map image where the background has obvious colour differences within the map. By comparing these images, K-means clustering works well to correct the colours in the image when $K = 2$ is set. However, while applying Mean shift clustering, it is difficult to determine a suitable bandwidth to correct the colours in the map image. When a small bandwidth is chosen, the clustering process cannot correct the colours.

However, when a large bandwidth is chosen, the result image is over smooth, which makes the image looks blurry.



(a) original image

(b) Threshold-based

(c) Edge detection

(d) Watershed

(e) Region growing method

(f) Region splitting and merging

(g) Region-based watershed

Figure 5.2: Satellite image with different techniques

In Figure 5.2, different techniques apart from active contours have been applied on the satellite image. Active contour model is not suitable for satellite image segmentation, since it relies on other mechanisms to place the initial contour around the desired target area. For a satellite image, it is difficult to place an initial contour around the whole land area.

By comparing these images,the watershed transformation technique performs more effective than others. The edge detection method, region growing and region splitting and merging shown above are obviously not well segmented. The threshold-based method and region-based with watershed transformation method have some holes at the land area which are not segmented properly. Although the result image of watershed transformation technique appears over-segmented, it still separate the sea and land very well. The only problem is that the land has been segmented into many small regions.

Apart from map colour correction, clustering can also be used to perform many surface analyses, allowing you to quickly win in various fields. For example, marketers can perform clustering analysis to quickly segment customer demographics. Insurance companies can quickly and deeply understand risk factors and locations, and generate an initial risk profile for policyholders. And apart from satellite image segmentation, the traditional segmentation techniques can also be used for medical imaging, such as locate tumors[15] and other pathologies.

Overall I believe my project has gone well to a certain extent and I have achieved the purposes of map colour correction and satellite image segmentation, though there still exist some improvements which I think are worth researching if there were more time. With more time to spend, I would explore further processing to eliminate the over-segmentation problem of the watershed transformation technique to achieve a better sea and land segmentation effect.

Bibliography

[1] The European Space Agency. *Sentinel-2, Colour vision for Copernicus.* URL: http://www.esa.int/Applications/Observing_the_Earth/Copernicus/Sentinel-2. (accessed: 01.02.2021).

[2] S. Anand et al. "Semi-Supervised Kernel Mean Shift Clustering". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.6 (2014), pp. 1201–1215. DOI: 10.1109/TPAMI.2013.190.

[3] Waheed Iqbal Bilal Iqbal and Abdelkarim Erradi. "Canny edge detection and Hough transform for high resolution video streams using Hadoop and Spark". In: *Cluster Computing* (2019). URL: https://doi.org/10.1007/s10586-019-02929-x.

[4] Fearghal Corbett. *The difference between vector and bitmap/raster images.* URL: https://thegoodfolk.co.uk/the-difference-between-vector-and-bitmap-raster-images/. (accessed: 11.18.2020).

[5] SANTOSH DAHAL. *What is Image.* URL: https://www.suntos.com.np/computer-vision-for-robotics/what-is-image.html#color-spaces. (accessed: 11.18.2020).

[6] Steve Eddins. "The Watershed Transform: Strategies for Image Segmentation". In: *MathWorks* (2001). URL: https://uk.mathworks.com/company/newsletters/articles/the-watershed-transform-strategies-for-image-segmentation.html.

[7] ESA. *Copernicus Open Access Hub, Sentinel-2 images.* URL: https://scihub.copernicus.eu/dhus/#/home. (accessed: 01.02.2021).

[8] ESA. *MultiSpectral Instrument (MSI) Overview.* URL: https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-2-msi/msi-instrument. (accessed: 18.11.2020).

[9] ESA. *Sentinel-2.* URL: https://www.esa.int/ESA_Multimedia/Images/2008/04/Sentinel-2. (accessed: 18.11.2020).

[10] Ravindran G. *Integrated Feature Extraction for Image Retrieval.* URL: https://www.researchgate.net/figure/llustration-of-the-HSV-Color-Space-B-Color-Feature-Extraction-Color-feature-is-extracted_fig1_321126312. (accessed: 31.01.2021).

[11] Z. Gan et al. "Multilevel Segmentation Optimized by Physical Information for Gridding of Microarray Images". In: *IEEE Access* 7 (2019), pp. 32146–32153. DOI: 10.1109/ACCESS.2019.2900249.

[12] Z. Gan et al. “Multilevel Segmentation Optimized by Physical Information for Gridding of Microarray Images”. In: *IEEE Access* 7 (2019), pp. 32146–32153. DOI: 10.1109/ACCESS.2019.2900249.

[13] GISGeography. *What is NDVI (Normalized Difference Vegetation Index)?* URL: https://gisgeography.com/ndvi-normalized-difference-vegetation-index/. (accessed: 01.02.2021).

[14] Jorge Cadima Ian T. Jolliffe. “Principal component analysis: a review and recent developments”. In: *The Royal Society* (2016). URL: https://doi.org/10.1098/rsta.2015.0202.

[15] Wei Wu, Albert Y C Chen, Liang Zhao Jason J Corso. “Brain tumor detection and segmentation in a CRF (conditional random fields) framework with pixel-pairwise affinity and superpixel-level features”. In: *Int J Comput Assist Radiol Surg.* (2013). DOI: 10.1007/s11548-013-0922-7.

[16] Jianping Fan et al. “Automatic image segmentation by integrating color-edge extraction and seeded region growing”. In: *IEEE Transactions on Image Processing* 10.10 (2001), pp. 1454–1466. DOI: 10.1109/83.951532.

[17] H. G. Kaganami and Z. Beiji. “Region-Based Segmentation versus Edge Detection”. In: *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. 2009, pp. 1217–1221. DOI: 10.1109/IIH-MSP.2009.13.

[18] C. Li et al. “Minimization of Region-Scalable Fitting Energy for Image Segmentation”. In: *IEEE Transactions on Image Processing* (Oct. 2008). ISSN: 1941-0042. DOI: 10.1109/TIP.2008.2002304.

[19] Laurens van der Maaten, Eric Postma, and Jaap van den Herik. *Dimensionality Reduction: A Comparative Review*. Tilburg centre for Creative Computing, Tilburg University, 2009. URL: https://members.loria.fr/moberger/Enseignement/AVR/Exposes/TR_Dimensiereductie.pdf.

[20] Ehsan Akbari Sekehravani. Eduard Babulak. Mehdi Masoodi. “Implementing canny edge detection algorithm for noisy image”. In: *Bulletin of Electrical Engineering and Informatics* (2020). DOI: 10.11591/eei.v9i4.1837.

[21] Dorin Comaniciu. Peter Meer. “Mean Shift: A Robust Approach toward Feature Space Analysis”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2002). DOI: 10.1109/34.1000236.

[22] G. Meeragandhi, bS.Parthiban, and cNagaraj A. “Ndvi: Vegetation change detection using remote sensing and gis – A case study of Vellore District”. In: Mar. 2015. URL: https://www.researchgate.net/publication/273520952_Ndvi_Vegetation_change_detection_using_remote_sensing_and_gis_-_A_case_study_of_Vellore_District.

[23] Mixam. *Colour guide for printing*. URL: https://mixam.co.uk/support/colour. (accessed: 31.01.2021).

[24] S. Na, L. Xumin, and G. Yong. "Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm". In: *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*. 2010, pp. 63–67. DOI: 10.1109/IITSI.2010.74.

[25] Lior Pachter. "What is principal component analysis?" In: *Bits of DNA* (2014). URL: https://liorpachter.wordpress.com/2014/05/26/what-is-principal-component-analysis/.

[26] SUNIT PRASAD. "What is Image Segmentation?" In: *Analytix Labs* (). URL: https://www.analytixlabs.co.in/blog/what-is-image-segmentation/.

[27] B.B. Zobel R. Setola L.M. Montoni. Elsevier, 2016. Chap. Chapter 27 Motion Artifacts Compensation in DCE-MRI Framework Using Active Contour Model.

[28] Chiluka Nagaraju. I Raja Sekhar Reddy. R. Pradeep Kumar Reddy. "Canny Scale Edge Detection". In: *International Journal of Engineering Trends and Technology* (2016). DOI: 10.14445/22315381/IJETT-ICGTETM-N3/ICGTETM-P121.

[29] Linda G.Shapiro Robert M.Haralick. "Survey: Image segmentation techniques". In: *Computer Vision, Graphics, and Image Processing* (1985). URL: https://www.sciencedirect.com/science/article/pii/S0734189X85901537.

[30] Khamitkar S.D Salem Saleh Al-amri N.V. Kalyankar. "Image Segmentation by Using Thershod Techniques". In: *JOURNAL OF COMPUTING* 2 (5 2010). ISSN: 2151-96172. URL: https://arxiv.org/ftp/arxiv/papers/1005/1005.4020.pdf.

[31] National Library of Scotland. *Map Images*. URL: https://maps.nls.uk/. (accessed: 01.02.2021).

[32] Christian Lantuéjoul Serge Beucher. "Use of Watersheds in Contour Detection". In: *International workshop on image processing, real-time edge and motion detection* (1979). URL: https://www.researchgate.net/publication/230837989_Use_of_Watersheds_in_Contour_Detection.

[33] Jean Serra. *Image Analysis and Mathematical Morphology*. USA: Academic Press, Inc., 1983. ISBN: 0126372403. DOI: 10.5555/1098652.

[34] Tao Zhou Shi Qiu Jingtao Sun et al. "Spiculation Sign Recognition in a Pulmonary Nodule Based on Spiking Neural P Systems". In: *BioMed Research International* (2020). URL: https://doi.org/10.1155/2020/6619076.

[35] Lukas Stratmann. *CMY and CMYK*. URL: http://color.lukas-stratmann.com/color-systems/cmy.html. (accessed: 31.01.2021).

[36] Tutut Herawan Wan Maseri Wan Mohd Abul Beg and Khandakar Fazley Rabbi. *Computational Intelligence and Intelligent Systems*. Springer Verlag, 2012. DOI: 10.1007/978-3-642-34289-9_22.

[37] R. Xu and D. C. Wunsch. "Clustering Algorithms in Biomedical Research: A Review". In: *IEEE Reviews in Biomedical Engineering* 3 (2010), pp. 120–154. DOI: 10.1109/RBME.2010.2083647.

[38] A. Yezzi et al. "A geometric snake model for segmentation of medical imagery". In: *IEEE Transactions on Medical Imaging* 16.2 (1997), pp. 199–209. DOI: 10.1109/42.563665.