Manhole Cover Detection from Natural Images

Sen Yan, B.Eng

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Intelligent Systems)

Supervisor: Kenneth Dawson-Howe

September 2020

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Sen Yan

September 5, 2020

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Sen Yan

September 5, 2020

Acknowledgments

Foremost, I would like to thank my supervisor, Dr. Kenneth Dawson-Howe for providing valuable suggestions and great encouragement throughout the duration of this dissertation.

I would also express my sincere gratitude to my second reader Dr. Gavin Doherty for his time and advice which also helped me immensely.

I must thank my family for supporting me to attend college and providing their emotional and moral support. They always stand by my side for everything, especially when I am faced with difficulties.

Finally, special thanks to my friends and my landlords for their consistent support and company in this strange period.

Sen Yan

University of Dublin, Trinity College
September 2020

Manhole Cover Detection from Natural Images

Sen Yan, Master of Science in Computer Science

University of Dublin, Trinity College, 2020

Supervisor: Kenneth Dawson-Howe

The accuracy of vehicle locating systems based on Global Positioning System is limited when used in small area. This study aims to improve the performance based on manhole covers, one of the road characteristics that is hardly considered before. In this dissertation, the work contains dataset establishment, image processing and evaluations. The manhole covers will be detected based on the image segmentation according to texture feature and ellipse fitting and filtering. Laplacian of Gaussian filter and Gabor filter has been used respectively to extract texture features. K-means algorithm, in which the number of clusters is determined by elbow method according to Sum of Square Error, is adopted to divide pixels in the image into different regions and Least-Squares method is used in ellipse fitting processing.

The results based on two kinds of filters have been evaluated and compared using evaluations including *Recall*, *Precision*, *Accuracy* and F_1 *score*. The performance of the method with Laplacian of Gaussian is better than the other one, and for images in which the ratio of the area of manhole cover to the image size is between 8% to 20%, the method proposed performs best. The study could be improved by selecting or combining more filters in the process of feature extraction and using better-performed clustering algorithms. The reliability could be improved by the expansion of the dataset.

Contents

Acknowledgments **iii**

Abstract **iv**

List of Tables **vii**

List of Figures **viii**

Chapter 1 Introduction **1**

1.1 Background 1
1.2 Motivation 1
1.3 Challenges 2
1.4 Overview 2

Chapter 2 Literature Review **6**

2.1 Edge Detection 6
2.2 Image Segmentation and Texture Feature 8
2.2.1 Gray Level Co-occurrence Matrix 9
2.2.2 Gabor Filter 10
2.2.3 Laplacian of Gaussian Filter 12
2.3 Summary 13

Chapter 3 Design and Implementation **14**

3.1 Architecture 14
3.2 Texture Feature Extraction 15
3.3 Segmentation 18

3.3.1 Sum of Square Error (SSE) . . . 18
3.3.2 Silhouette . . . 20
3.4 Ellipse Fitting . . . 24
3.4.1 Random Sample Consensus (RANSAC) . . . 24
3.4.2 Least-Squares (LS) Method . . . 26
3.5 Filtering with Conditions . . . 27
3.5.1 Geometric Properties of Ellipses . . . 27
3.5.2 Actual Situation of Manhole Covers . . . 30
3.6 Summary . . . 31

Chapter 4 Results and Discussions **32**
4.1 Dataset . . . 32
4.2 Evaluations . . . 34
4.3 Results . . . 39
4.4 Discussions . . . 39

Chapter 5 Conclusion and Future Work **42**
5.1 Conclusions . . . 42
5.2 Future Works . . . 43

Bibliography **44**

Appendices **47**

List of Tables

2.1 Parameters used in Canny edge detection for images shown in Figure 2.1 7

3.1 Sum of Square Error (SSE) results and its changes 20
3.2 Silhouette results 22

4.1 The attribute of each group 33
4.2 Results of the approach with Gabor Filters used in texture feature extraction 39
4.3 Results of the approach with LoG Filters used in texture feature extraction 39

List of Figures

1.1 Texture feature extraction and its results 3
1.2 Image segmentation and the result 4
1.3 Ellipse fitting and filtering and its output 5

2.1 Acceptable results of Canny edge detection 7
2.2 Unacceptable results of Canny edge detection 8
2.3 Results of some textual descriptors of GLCM 11
2.4 The results of 16 Gabor filters selected 12
2.5 The results of LoG edge detection using various parameters 13

3.1 The architecture of the system 15
3.2 The extraction and processing to $L*$, a and $b*$ channels 16
3.3 The results of LoG filters with pre-defined parameters 17
3.4 Histograms of certain pixels in one image 19
3.5 The relationship between k and SSE 21
3.6 The relationship between k and Silhouette 23
3.7 The input image and the result of segmentation 23
3.8 The final result of image segmentation 24
3.9 The result of best performed model from RANSAC algorithm 25
3.10 Several unacceptable results from RANSAC algorithm 26
3.11 The symmetry of an ellipse 28
3.12 Ellipses with different aspect ratios 29
3.13 Distances of points to an ellipse in a 2-dimensional space 30

4.1 Samples of pictures in Large group 34

4.2 Samples of pictures in Medium group 35
4.3 Samples of pictures in Small group 36
4.4 Samples of pictures in Tiny group 37
4.5 Final results of the system 41

Chapter 1

Introduction

1.1 Background

Autonomous driving (AD) technology has been a subject of much concern and controversy since it was first proposed. A survey conducted in several countries in 2014 of public opinion about self-driving vehicles (SDVs) shows that a large percentage of respondents still highly concerned about the safety consequences of equipment or system failure, although the majority of them felt positive about SDVs [1].

As of 2020, many companies have started testing their vehicles on roads [2]. Numerous media have even predicted when this technology would be officially put into use, but the safety issue is still one of the elements influencing people's attitude towards the technology [3]. It is widely accepted that the realization of AD technology requests the reliability and robustness of the position determination of the vehicle itself, as well as the prediction and observation of other traffic participants [4].

1.2 Motivation

Global Positioning System (GPS) has been widely accepted as an approach used in vehicle navigation and tracking systems [5]. However, the reliability and accuracy are still challenging issues [6] and the missing of road characteristics will also lead to a failure [7].

According to a previous study [8], the error in longitude and latitude coordinates is

10 to 15 meters, caused by different error sources, in 95% of readings. That means the accuracy of GPS is limited when it is used in a small area (e.g. to distinguish one side of the road to the other). That will lead to incorrect locating results of the vehicles in terms of positions and directions. These limitations are mainly caused by the method used in GPS, which is based on the measurement of the signal propagation delay from the satellite to the GPS receiver.

To overcome the shortcomings of the method used in GPS and improve the accuracy of vehicle locating results, a method based on other positioning principles shall be proposed and evaluated. Road characteristics including traffic lights, lamp posts and manhole covers are different in one direction of a certain road and the opposite direction, so methods based on the detection of road characteristics (e.g. manhole covers) could be helpful to improve the accuracy of vehicle locating system compared to the approach used in GPS.

The detection of some road characteristics, including traffic lights and traffic signs, have been introduced and conducted in previous studies [9][10], but there is still an absence of a the manhole cover detection from natural images. Accordingly, this dissertation can provide some possibilities for creating a vehicle locating system with higher accuracy than GPS based on manhole covers.

1.3 Challenges

The main objective of this dissertation is to detect and extract the manhole covers from the natural images, but it is a challenge to acquire the pictures and establish the dataset because images of manhole covers with appropriate size are uncommon and difficult to find.

1.4 Overview

This dissertation aims at improving the accuracy of vehicle locating system by using manhole cover detection based on natural images.

The method proposed includes four steps of work: feature detection from the natural pictures, image segmentation based on the feature detected, fitting image segments

to ellipses and filtering according to conditions predefined. The related works and knowledge, including edge detection methods, the definition of texture features and introduction to image segmentation are introduced in Chapter 2, while the implementation is detailed in Chapter 3.

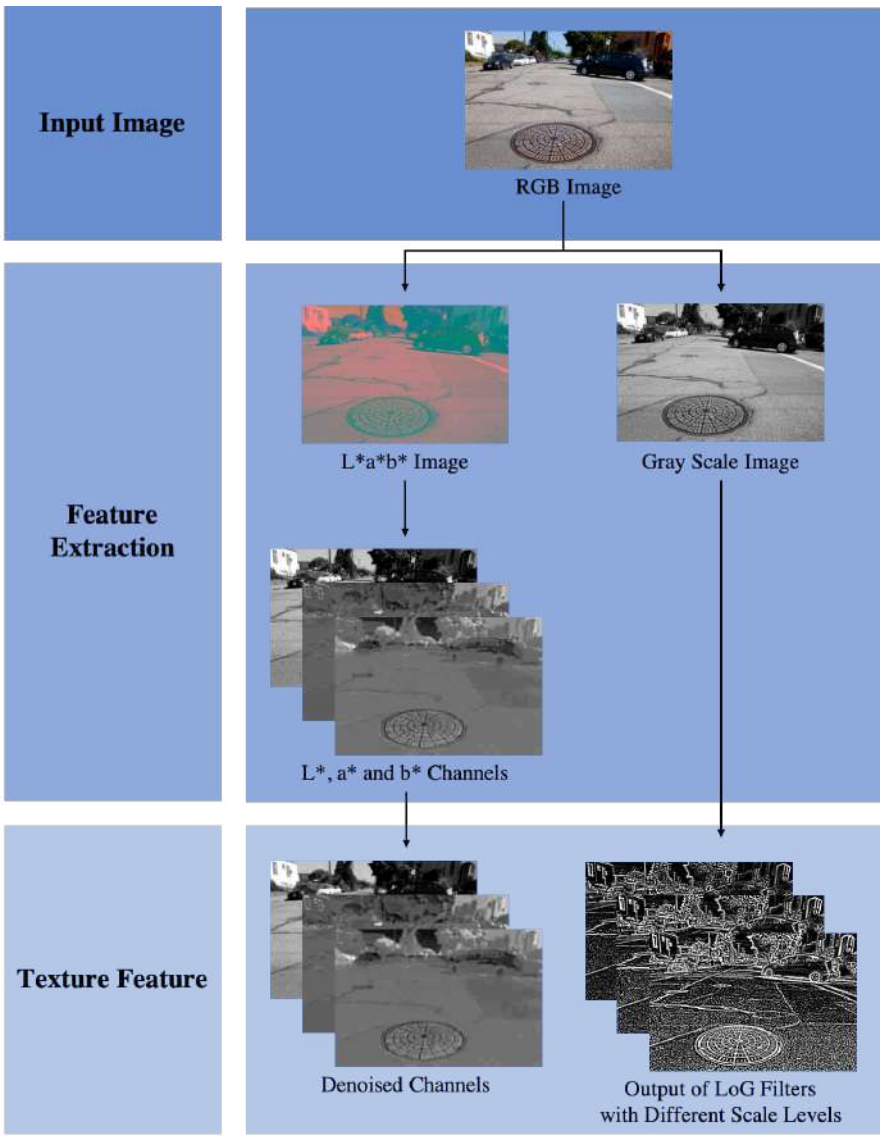


Figure 1.1: Texture feature extraction and its results

Several pictures are used to demonstrate the process of texture feature extraction in Figure 1.1 followed by the results. Image segmentation is then conducted based on

the features extracted. The pictures showing the process and the result could be seen in Figure 1.2.

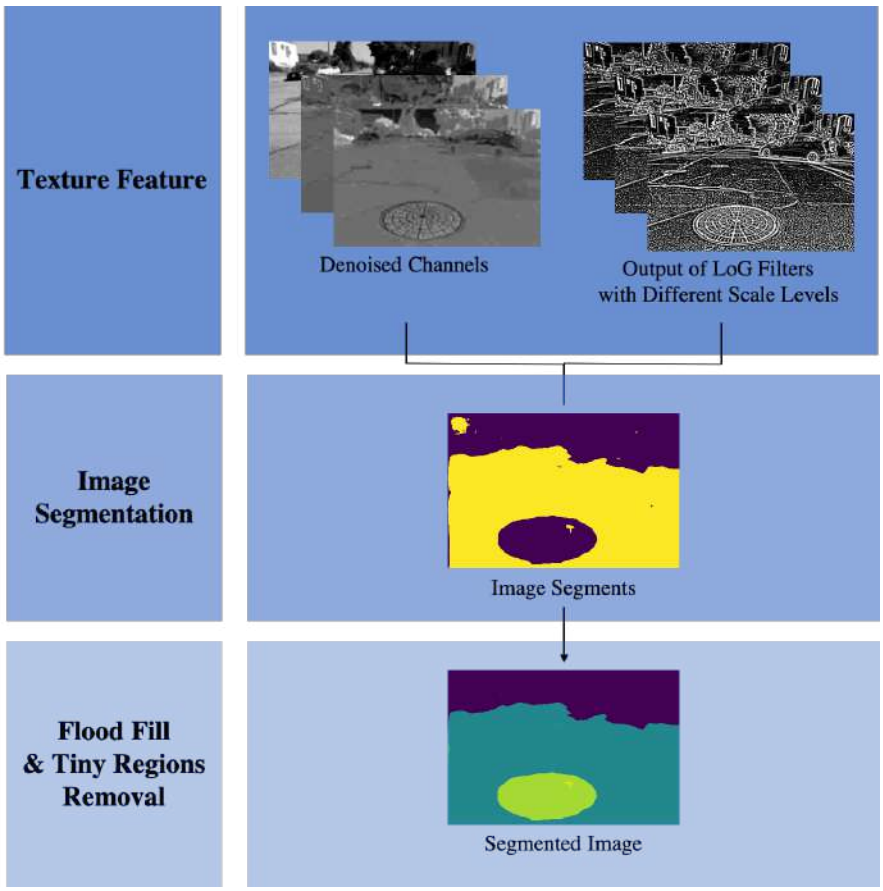


Figure 1.2: Image segmentation and the result

Every region in the segmented image will be fitted to ellipse and filtered with conditions predefined in terms of the geometric properties of ellipses and the actual situations of manhole covers. The conditions are used to examine the similarity between each region and its corresponding ellipse and promote each ellipse is a meaningful and appropriate candidate of a manhole cover.

The pictures used to demonstrate the processes mentioned above is shown in Figure 1.3. Evaluations of this system in terms of Recall, Precision, Accuracy and F1-score are used to represent the performance of the method proposed in this paper, and the description of the dataset and the discussion about the results are detailed in Chapter 4. Generally speaking, this approach performs best for the pictures in which the size of

a manhole cover accounts for 8% to 20% of the image size. The values of evaluations mentioned above are: $Recall = 62.5\%$, $Precision = 53.3\%$, $Accuracy = 40.0\%$ and $F_1\ score = 57.1\%$ respectively.

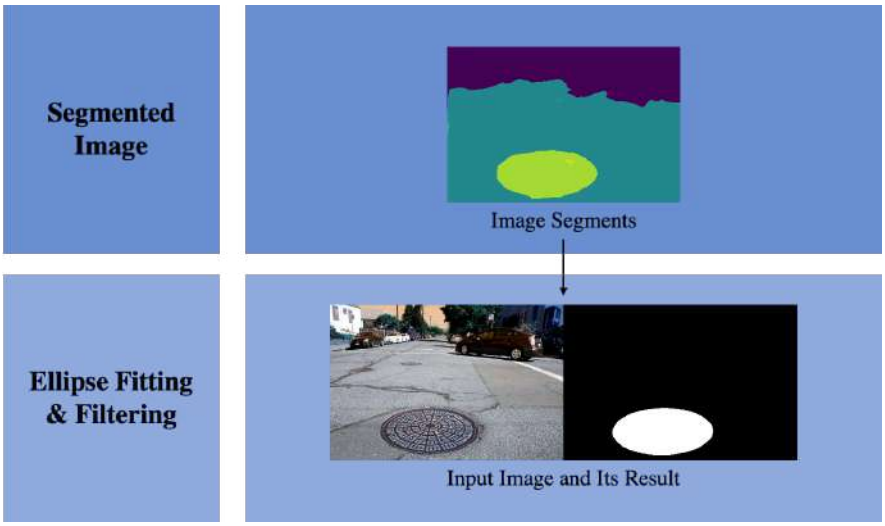


Figure 1.3: Ellipse fitting and filtering and its output

In the last part of this paper, conclusions are drawn based on the results and the plan about future work is also discussed.

Chapter 2

Literature Review

The object of this dissertation, as is mentioned and described in the last chapter, is to examine whether there exists any manhole cover in every single natural picture in the dataset established. Here I will introduce several different methods to detect ellipses from images, some definitions of features used in those methods, the reasons for using these features and how to use them.

Definitions and related introduction to the approaches about edge detection are introduced in Section 2.2, and those about image segmentation are included in the next section.

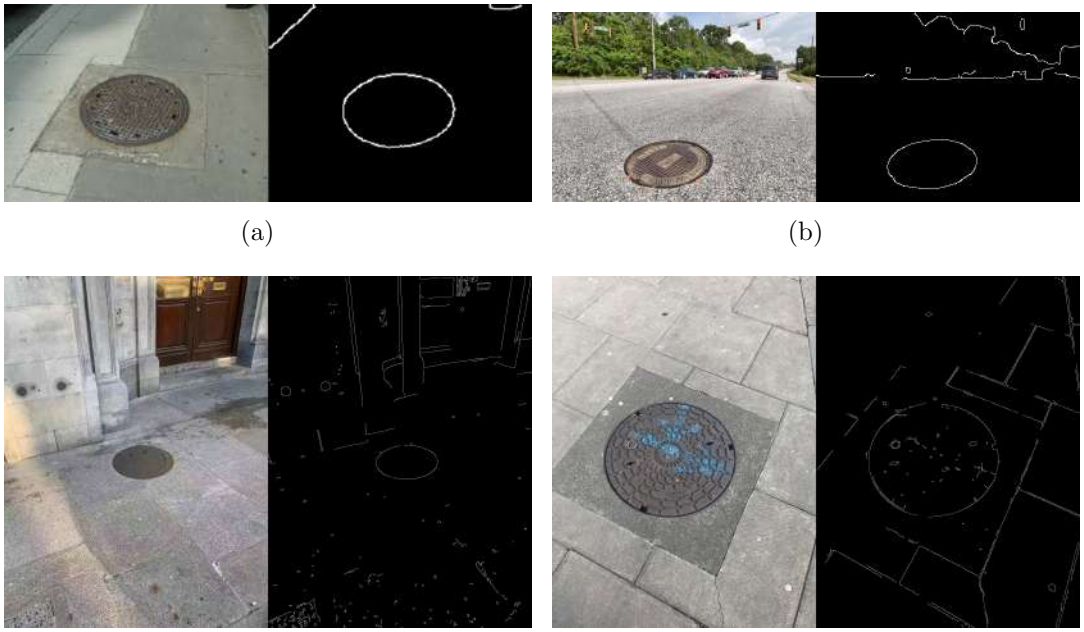
2.1 Edge Detection

Edge detection refers to the process of recognizing and locating the discontinuities (i.e. those points at which the pixel intensity changes sharply) in a digital image.

A previous paper [11] discussed and compared several edge detection techniques in two groups (i.e. methods based on Gradient-based and Laplacian-based). The paper states that the sensitivity to noise is the major drawback with negative effects of Gradient-based approaches such as Prewitt filter. At the same time, Canny edge detection, as a method based on Gaussian filter, performs better than other approaches discussed in the paper (e.g. Sobel, Robert, Laplacian of Gaussian). But, as the author mentioned in the paper, it is extremely dependent on the adjustable parameters.

Figure 2.1 shows several acceptable results with parameters (detailed in 2.1, in

which σ represents the standard deviation of the Gaussian filter for smoothing and T_1 and T_2 are intensity thresholds) obtained from plenty of trials and comparisons.



(a) (b)

(c) (d)

Figure 2.1: Acceptable results of Canny edge detection

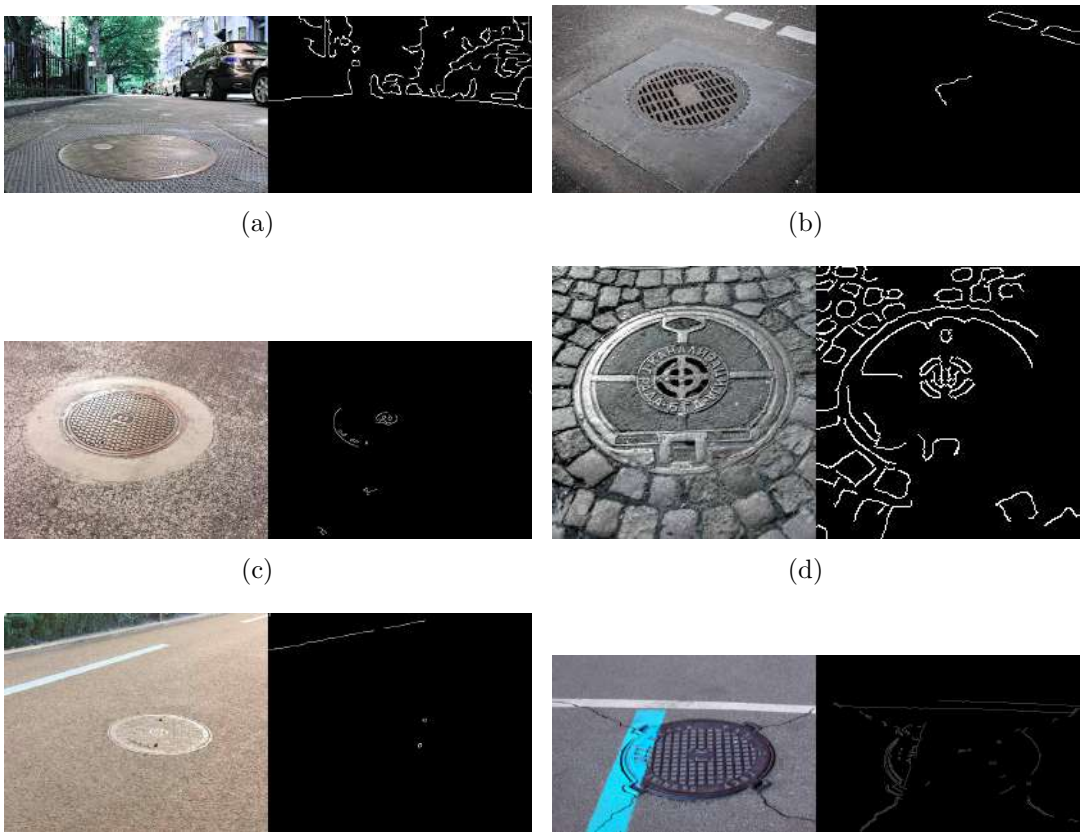
Table 2.1: Parameters used in Canny edge detection for images shown in Figure 2.1

Image ID	σ	T_1	T_2
(a)	13	50	150
(b)	13	80	200
(c)	9	70	150
(d)	9	90	190

It could be seen that regarding the differences (e.g. size of images, the color of targets, noise) among the images, the parameters are not all the same, even completely different. That means we need to select or adjust the parameters manually for every single image when using Canny edge detection, and this is totally unsuitable for a large dataset and unacceptable for the motivation of this dissertation.

However, the adoption of same parameters when processing different images in the dataset makes the results vary a lot, most of which are not acceptable for the following

steps. Figure 2.2 shows some terrible results, in which the parameters used are: $\sigma = 13$, $T_1 = 50$ and $T_2 = 150$.



(a) (b) (c) (d) (e) (f)

Figure 2.2: Unacceptable results of Canny edge detection

2.2 Image Segmentation and Texture Feature

Image segmentation means the process of dividing a digital image into several segments according to the similarity and difference among pixels. It mainly aims at providing a more meaningful version of image that is easier to analyze [12]. Except for color feature, image texture is another feature that is usually used to segment images into regions of interest. A previous paper discussed and divided texture analyzing approaches into four categories: statistical, structural, model-based and transform-based methods [13].

2.2.1 Gray Level Co-occurrence Matrix

Gray level co-occurrence matrix (GLCM), as a traditional statistical approach, use the distribution of gray level values among the pixels in gray-level image to distinguish different textures. It is usually not directly used as a feature to distinguish the textures due to the large dimension, but the statistical textual descriptors calculated based on GLCM are widely accepted to present the texture features of the image. For a point (x_1, y_1) in a certain image, the gray-scale value is represented as g_1. Assuming the position of the point corresponding to it is (x_2, y_2), the gray-scale value is g_2. Combining these two gray-scale values, we could get a gray-scale pair (g_1, g_2) and GLCM could be acquired by counting and calculating the probability matrix P of each gray-scale pair in the whole image. Some of the most commonly used and widely accepted textual descriptors are the angular second moment (ASM), the contrast, the inverse differential moment (IDM), the entropy and the dissimilarity, and they could be calculated by the equations below (see Equation 2.1 to Equation 2.5) [14], in which i and j represents the spatial coordinates of the function $P(i, j)$ and n is the gray tone.

$$ASM = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(P\left(i,j\right)\right)^2 \tag{2.1}$$

ASM reflects the uniformity of image gray-scale distribution and texture thickness. When the image texture is uniform and regular, the ASM value is relatively large. On the contrary, the values of the GLCM will be similar and the ASM value will be small.

$$Contrast = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(i-j\right)^2 \cdot P\left(i,j\right) \tag{2.2}$$

Contrast shows the sharpness of the image. The sharp texture results in a high contrast of the image.

$$IDM = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{P\left(i,j\right)}{1+\left(i-j\right)^2} \tag{2.3}$$

IDM reflects the clarity and regularity of the texture. The IDM value will be large if the texture of the image is clear and easy to describe.

$$Entropy = -\sum_{i=1}^{n}\sum_{j=1}^{n} P(i,j) \cdot \log P(i,j) \tag{2.4}$$

Entropy measures the randomness of the amount of information contained in an image and expresses the complexity of the image. It will reach the maximum when all the values in GLCM are equal or the randomness of the pixel values reaches the peak.

$$Dissimilarity = \sum_{i=1}^{n}\sum_{j=1}^{n} |i-j| \cdot P(i,j) \tag{2.5}$$

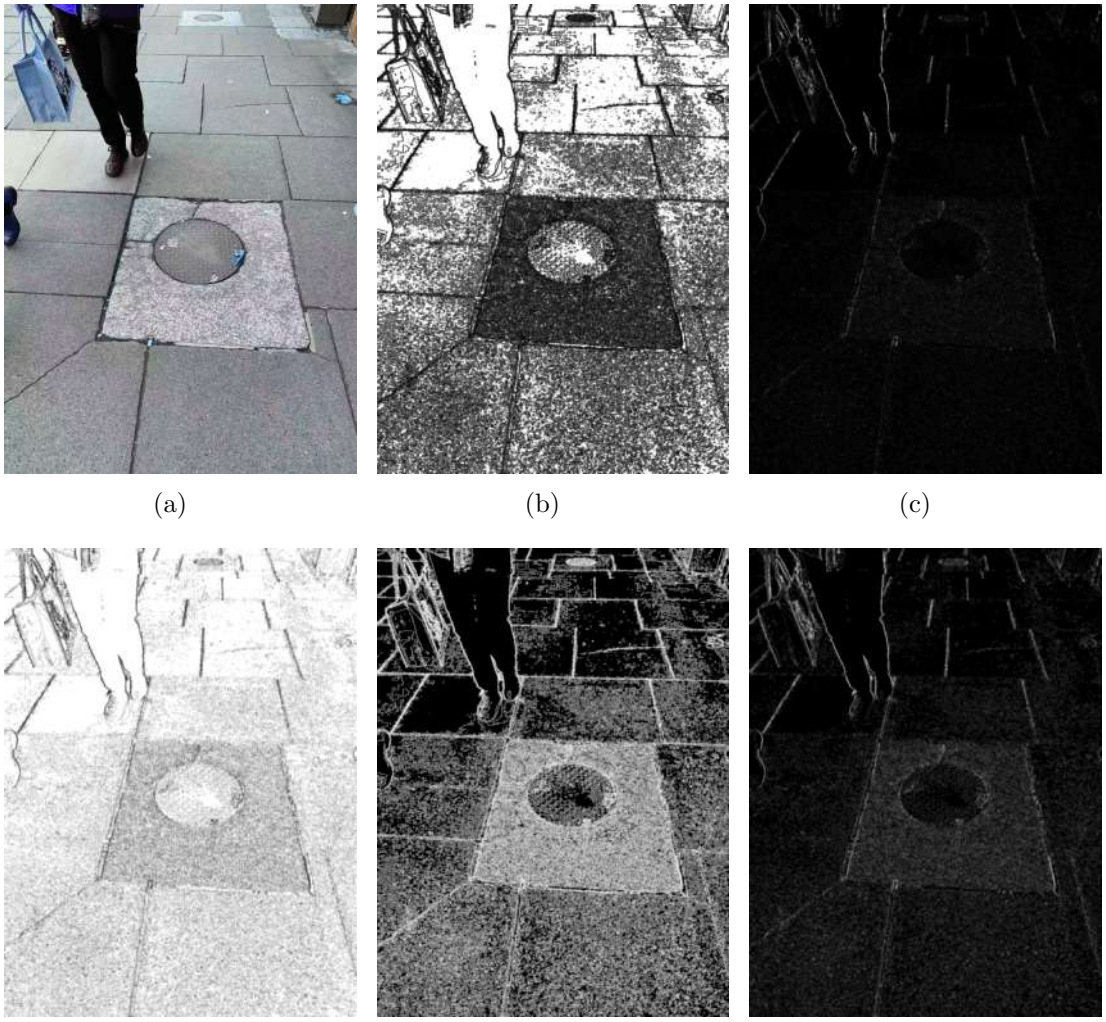
A sample image and the related results calculated with the equations could be seen in the Figure 2.3, in which 2.3(a) shows the original image, 2.3(b) is the result of ASM, 2.3(c) represents the result of the contrast, 2.3(d) comes from IDM equation (i.e. Equation 2.4), 2.3(e) is the result of entropy and 2.3(f) shows the dissimilarity and the size of the original image is 1080×1440.

2.2.2 Gabor Filter

Gabor filter is widely used to extract features by analyzing the frequency domain of the image [15][16]. For a digital image, different textures usually have different center frequencies and bandwidths in frequency domain and accordingly a set of Gabor filters could be defined. Each Gabor filter will only detect one kind of features corresponding to the frequency of the filter. Combining all the output results, the texture features would be successfully extracted.

Gabor filter allows users to combine the filters and define the filter set flexibly, and it is regarded as one of the algorithms closest to modern deep learning methods for image classification.

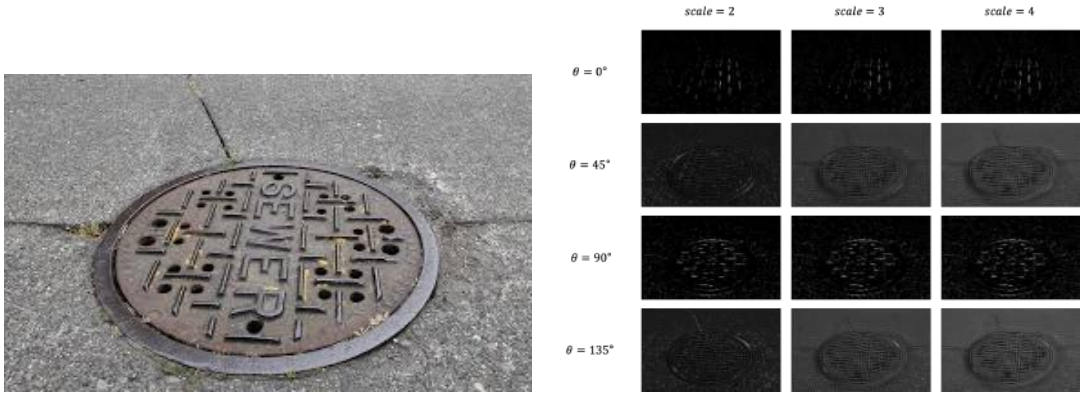
For a sample image input (see Figure 2.4 – 2.4(a)), of which the size is 290×174, sixteen Gabor filters (with three scales in four orientations) have been applied to extract the texture feature. The results could be viewed in Figure 2.4. The scales used are 2, 3 and 4 respectively and the orientations are $0°$, $45°$, $90°$ and $135°$.



(a) (b) (c)

(d) (e) (f)

Figure 2.3: Results of some textual descriptors of GLCM



(a) (b)

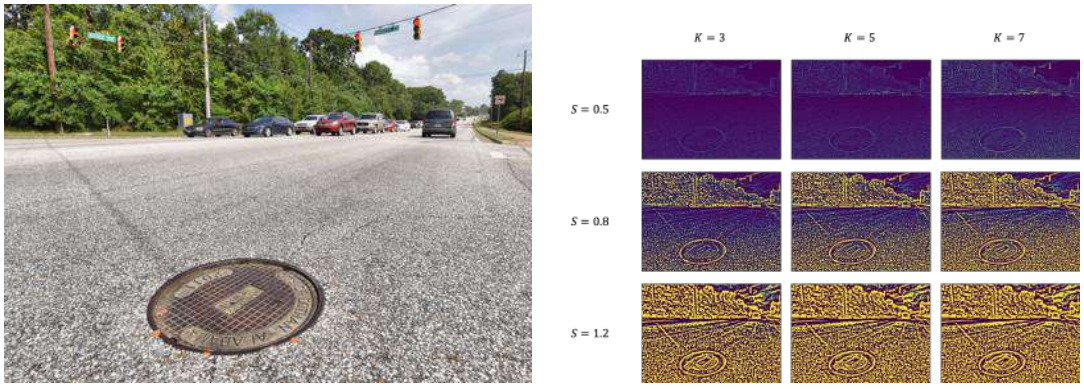
Figure 2.4: The results of 16 Gabor filters selected

2.2.3 Laplacian of Gaussian Filter

As is discussed in a previous paper [17], texture features could also be extracted by calculating the convolution of a natural image with a set of filters. In addition to Gabor filter introduced above, it is worth mentioning another one – Laplacian of Gaussian filter.

Laplacian of Gaussian (LoG) filter could also be used as an edge detecting method, as is discussed in [11], with strong robustness, high boundary-positioning accuracy, good edge continuity and less adjustable parameters (i.e. the kernel size K and the scale value S) than Canny's approach. It combines Gaussian Smoothing with Laplacian Filtering and produces a kernel that could be calculated in advance. The output of the LoG filter is a grey-level image, rather than a binary one as Canny detector does.

When using only one LoG filter with well-selected parameters, the edge of a certain image could be detected, while combining the results of a set of LoG filters with different parameters, the texture information could be described. A sample image and the influence caused by different parameters selected are demonstrated in Figure 2.5, in which the size of the image is 400×287.



(a) (b)

Figure 2.5: The results of LoG edge detection using various parameters

2.3 Summary

As is discussed in the last section, the motivation of this dissertation is to apply image processing methods to detect and extract the ellipse in natural images.

Edge detection techniques allow us to extract the edge of a digital image and if the parameters are well-defined, the results would be great. Once the edges have been extracted, we could select the edge points and fit them into ellipses, then using several conditions to examine the possibility for this ellipse to be a manhole cover.

Image segmentation provides us with another method to recognize an ellipse. Using the texture features detected by the approaches introduced above, pixels in a certain image could be segmented into several groups. We could calculate the similarity between each group and a potential ellipse, then examine if this part could be a manhole cover with conditions predefined.

Chapter 3

Design and Implementation

This chapter mainly focuses on the processes of operation. It includes the theories and methods considered and tried when analyzing the task and more importantly, the tools and technologies finally adopted to complete the task.

Generally speaking, the method used in this study could be divided into four steps – texture feature extraction, image segmentation, ellipse fitting and filtering by conditions. A more detailed explanation could be seen in the following sections based on the architecture of the dissertation.

3.1 Architecture

Figure 3.1 shows the architecture of the approach used in this dissertation. As is shown and discussed above, it contains feature extraction, image segmentation, ellipse fitting and filtering by conditions pre-defined.

Firstly, a natural image will be convolved with a set of filters to extract texture features [17], which would be then used to assign the pixels in the image into several groups according to clustering algorithms, of which the number of clusters will be determined in advance. Every group of pixels could be regarded as one segment of the original image. After that, the possibility of this segment becoming a candidate ellipse will be examined according to the basic graphical information. The last step is to filter the candidates with pre-defined conditions to determine if the ellipse could be a manhole cover.

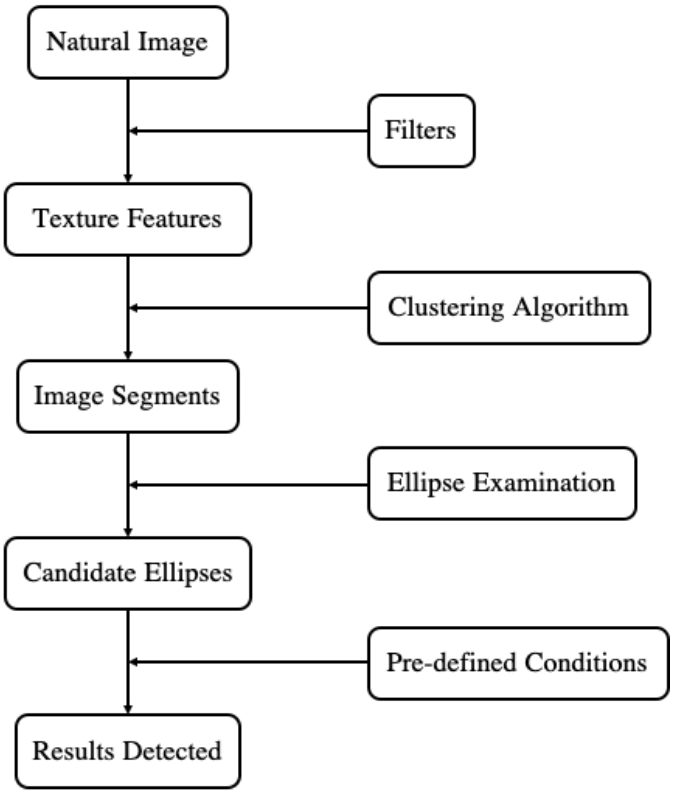


Figure 3.1: The architecture of the system

3.2 Texture Feature Extraction

As is mentioned in the architecture, the texture feature will be extracted by calculating the convolution of the image with a set of filters. That means the selection of the filter set is extremely influential and important to the feature results, which will directly affect the segmentation step.

$L*a*b*$ color space provides a conversion of the image information to one lightness-component $L*$ and two color-components (i.e. $a*$ and $b*$), in which $L*$ channel only expresses the lightness information and is designed to approximate human vision. Additionally, $L*a*b*$ color space is more perceptually uniform according to the related study [18], so the first step of texture feature extraction is to convert the image into L*a*b* color space. In order to reduce the influence of noise, median filter is applied to $L*$, $a*$ and $b*$ channel respectively to acquire denoised results. Figure 3.2 demonstrates

the process described above.

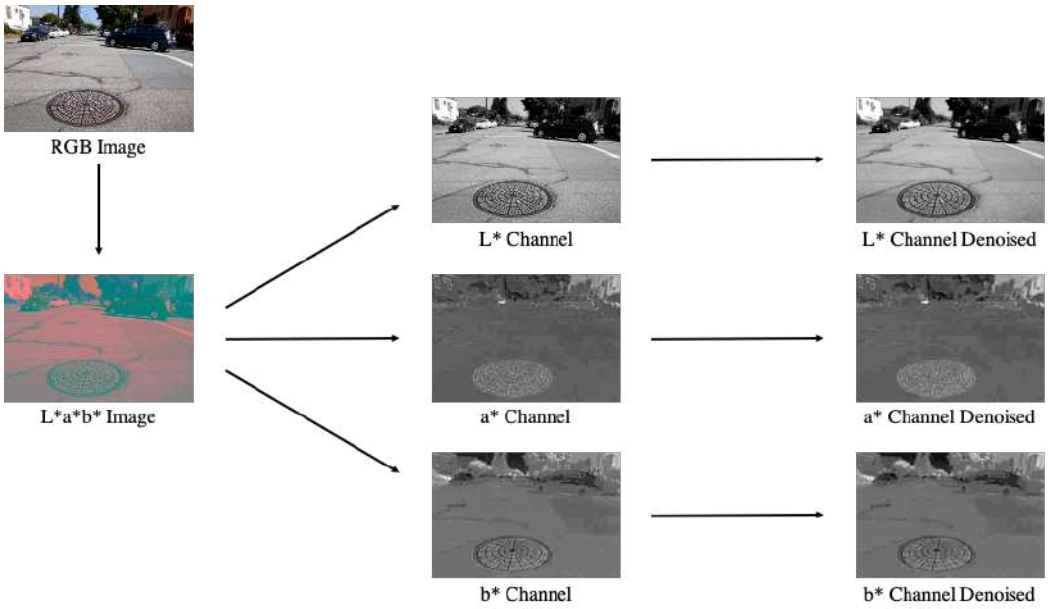


Figure 3.2: The extraction and processing to $L*$, a and $b*$ channels

As is introduced in Section 2.2.3 in this paper, we could combine the results of a set of LoG filters to extract the texture feature from an input image. When focusing on the texture features, the color information can be transferred into intensity information, so the LoG filters with different parameters are applied to the gray scale image converted from the original image. Here show the parameters used: kernel size $k = 5$, scale value $s = 0.5$, 0.8 and 1.2 respectively and the image size of the sample is 600×400. The process and the outputs could be seen in Figure 3.3.

It has been discussed in [19] that if the filters are well-selected, the spectral histogram could be used to represent an arbitrary texture. With the filters introduced and shown above, it is feasible and possible to calculate the spectral histogram of a window W, of which the size has been predefined, to represent the texture feature of the center point P of the window. Integral histogram has been introduced and discussed in [20] to speed up the histogram generation process.

Gathering the denoised results in three channels in L*a*b* color space and three outputs of LoG filters, we could establish an $M \times N$ feature matrix, of which M is the number of results used (6 in the sample discussed) and N represents the number

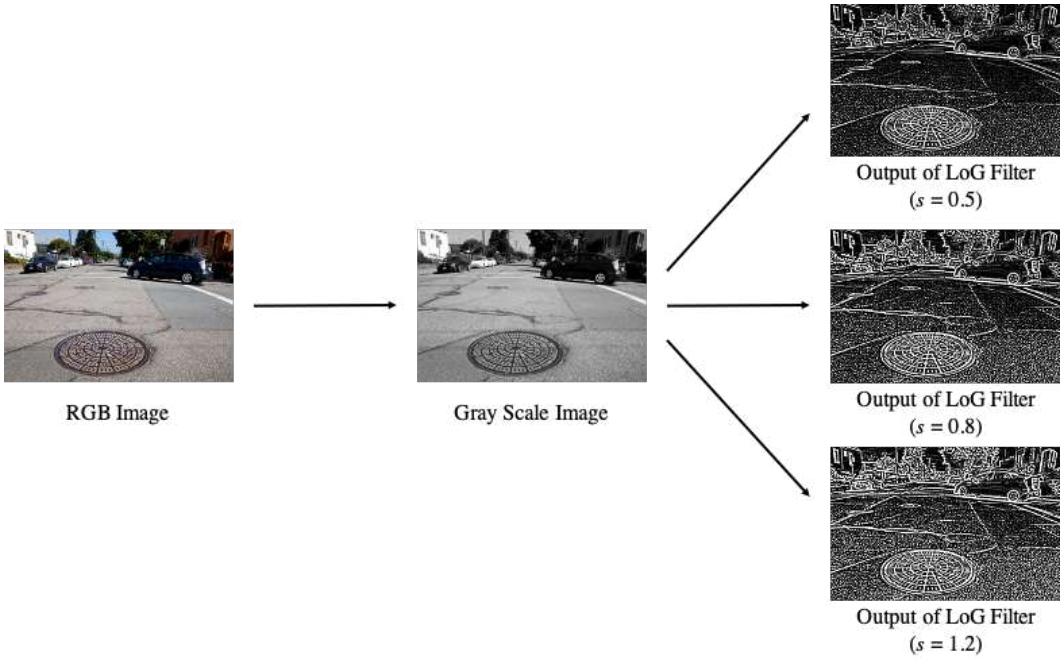


Figure 3.3: The results of LoG filters with pre-defined parameters

of pixels in the image ($600 \times 400 = 240,\ 000$ for the sample image) according to the theory proposed in a related paper [18]. Based on the theory about integral histogram and the feature matrix established, a histogram could be used to represent the texture information of every single window, in which the features of all the pixels are divided into different bins to represent the texture feature of its center point. The number of bins used in histograms is 11 for every single result, so the total of bins for all the six outputs is 6×11, and the window size is determined by the size of the input image.

Figure 3.4 shows the histograms of texture feature for several pixels in an the original RGB image shown in Figure 3.2. In the figure shown below, picture 3.4(a) shows the original image with several pixels selected as the target points, and picture (b) demonstrates the histograms (i.e. texture feature) of these target points. The position of each target point is: A $(100, 410)$, B $(110, 450)$, C $(115, 370)$, D $(140, 320)$, E $(160, 300)$ and F $(250, 310)$ respectively.

From the histograms shown in picture 3.4(b) in Figure 3.4, it could be easily known that for pixels from different regions visually distinguishable according to texture features, the histograms are also various (e.g. point A and D), and similar histograms are

shown when the pixels are lying in the same regions (i.e. point A, B and C and point D, E and F). That means if we could divide pixels in the image into several different parts according to the texture feature detected, the segmentation would be completed and regions with different shapes could be distinguished.

3.3 Segmentation

Based on the introduction and processing above, we get a histogram for every single pixel to represent the texture feature. In this section, we use clustering algorithm to divide the pixels into several differentiated regions based on the features extracted. K-means algorithm is one of the most popular clustering algorithms [21], so it is adopted in this dissertation to complete the segmentation processing. The development of K-means algorithm has been discussed in a previous paper [22], so here only introduce the basic idea about that.

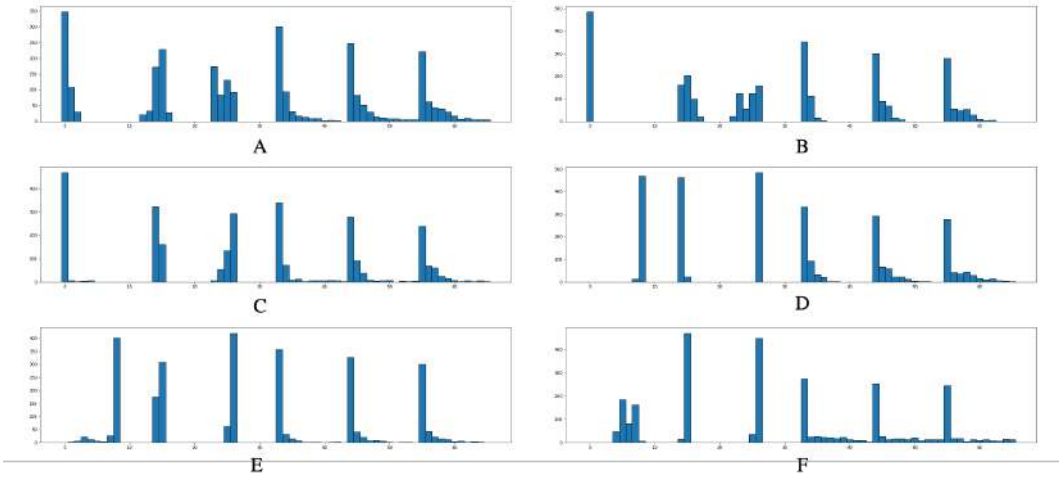
For a certain dataset, the goal of K-means algorithm is to relocate each piece of data to its nearest center based on an existing clustering and repeat the process with a set of updated centers until the convergence criteria (e.g. predefined number of iterations, difference on the value of the distortion function) is satisfied [21].

Regarding to the segmentation problem, the dataset is a single input image and each pixel is a target data. The feature extracted in the last step defines a 66-dimensional space where all the pixels are lying in. K-means algorithm is used to relocate each pixel to its nearest center point based on a clustering (i.e. k pixels randomly selected in the image) and provides a label for each pixel regarding to the related center. Then pixels with same labels will be regarded as one region due to the similarity of their texture features. Accordingly, the number of clusters shall be selected well in order to solve the segmentation problem.

3.3.1 Sum of Square Error (SSE)

There are several methods to select the number of clusters (i.e. the value of k in this problem) discussed in previous papers. The introduction and application of elbow method has been detailed in [23] by looking at the ideal k value graph with the position on the elbow along with the Sum of Square Error (SSE).

(a)



(b)

Figure 3.4: Histograms of certain pixels in one image

As the number of clusters k increases, the data will be clustered in more detail and the aggregation degree of each cluster will gradually increase, so the value of SSE will become smaller accordingly. When k is smaller than the best value k_0, the increase of k will lead to a drastic reduction in the aggregation degree, which sharply decreases the drop of SSE value. On the contrary, when k is larger than the best value k_0, the drop of SSE value will become smoother. That is to say, the best value k_0 lies in the elbow of the curve showing the relationship between k and the SSE value. Figure 3.5 is the line chart reflecting the relationship in the image shown in Figure 3.4 – 3.4(a). The SSE values and its changes are shown in Table 3.3.1.

Table 3.1: Sum of Square Error (SSE) results and its changes

Number of Clusters	SSE Results (1e10)	Differences of SSE (1e10)
1	4.9269	-
2	3.5995	1.3275
3	3.0064	0.5931
4	2.6522	0.3542
5	2.3318	0.3115
6	2.0680	0.2727
7	1.8394	0.2286
8	1.7043	0.1422

3.3.2 Silhouette

Another score used to evaluate the performance of the k value is called Silhouette, of which the definition and application have been detailed in a previous paper [24]. For a single data point P_i, the Silhouette value $s(i)$ could be calculated with Equation 3.1.

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{3.1}$$

$a(i)$ represents the distance of P_i to its own cluster C_0 and it could be acquired by calculating the average distance of P_i to all the other samples in C_0. In Equation 3.2, n_0 is the number of points in cluster C_0, $d(P_1, P_2)$ represents the Euclidean distance between P_1 and P_2.

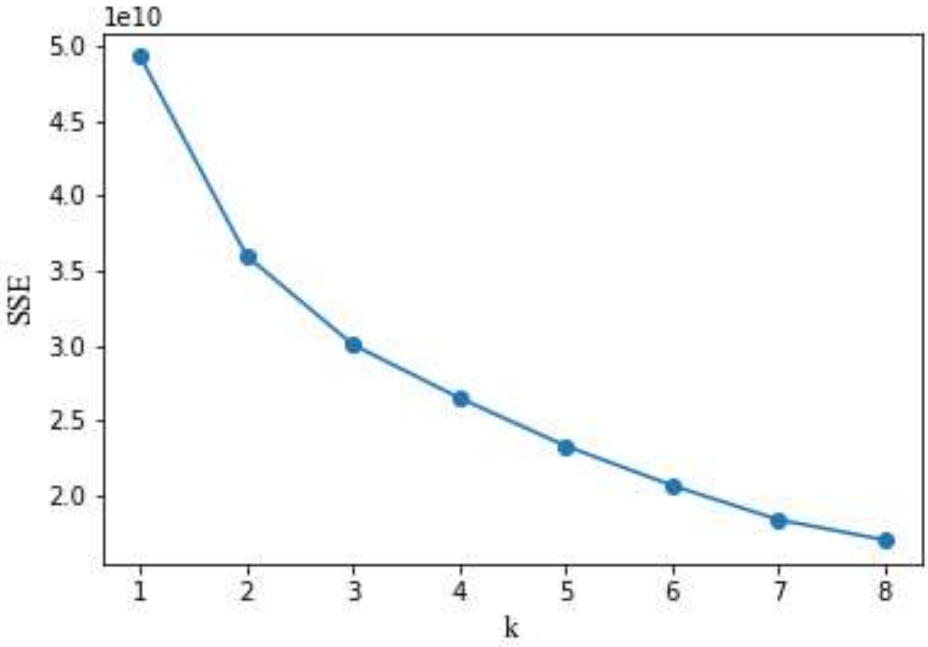


Figure 3.5: The relationship between k and SSE

$$a(i) = \frac{1}{n_0 - 1} \sum_{j=1, j \neq i}^{n_0} d(P_i, P_j) \tag{3.2}$$

$b(i)$ is the distance of P_i to its nearest cluster C' and it could be obtained by calculating the average distance of P_i to all the samples in C'. In Equation 3.3, n' is the number of points in cluster C', $d(P_1, P_2)$ represents the Euclidean distance between P_1 and P_2.

$$b(i) = \frac{1}{n'} \sum_{k=1}^{n'} d(P_i, P_k) \tag{3.3}$$

The Silhouette value for the whole cluster is defined by calculating the average of the Silhouette values for all points in the cluster, as Equation 3.4 shows. A good division is determined by a small value of $a(i)$ and a large value of $b(i)$, so the larger the Sil value, the better the clustering.

$$Sil = \frac{1}{n}\sum_{i=1}^{n} s\left(i\right) \tag{3.4}$$

Figure 3.6 reflects the relationship between k and Silhouette in the image shown in Figure 3.4 – (a), and the Sil values are shown in Table 3.3.2.

Table 3.2: Silhouette results

Number of Clusters	Silhouette Results
2	0.2776
3	0.2679
4	0.2730
5	0.2642
6	0.2855
7	0.3124
8	0.3235

As is shown in Figure 3.6, when the value of Silhouette reaches the peak, the value of k is 8. However, it is worth noting that from the elbow diagram of k and SSE (i.e. Figure 3.5), it could be seen that if k is larger than 5, the values of SSE are too small. Regarding to both SSE value and Silhouette value, k is finally set to 2.

According to the definition and calculation of the Silhouette values, there are two elements influencing the Sil value – $a\left(i\right)$ and $b\left(i\right)$. As the value of k is becoming too large, points in the image are over-divided, which will increase both the value of $a\left(i\right)$ and $b\left(i\right)$.

In summary, when using Silhouette value to determine the number of clusters, the value of SSE should be considered at the same time. Accordingly, SSE value plays a decisive role in selection of k. Based on the number of clusters determined, the division of input image could be conducted, and the results of segmentation is shown in Figure 3.7.

It could be seen that there are still some tiny regions segmented incorrectly, and some unconnected area with similar texture features have the same label. To reduce the negative influences of them in the following steps, the all the regions in the result will be relabeled based on Flood Fill algorithm [25], while tiny regions (e.g. the regions with an area smaller than 1% of the image size) will be removed. The final result could

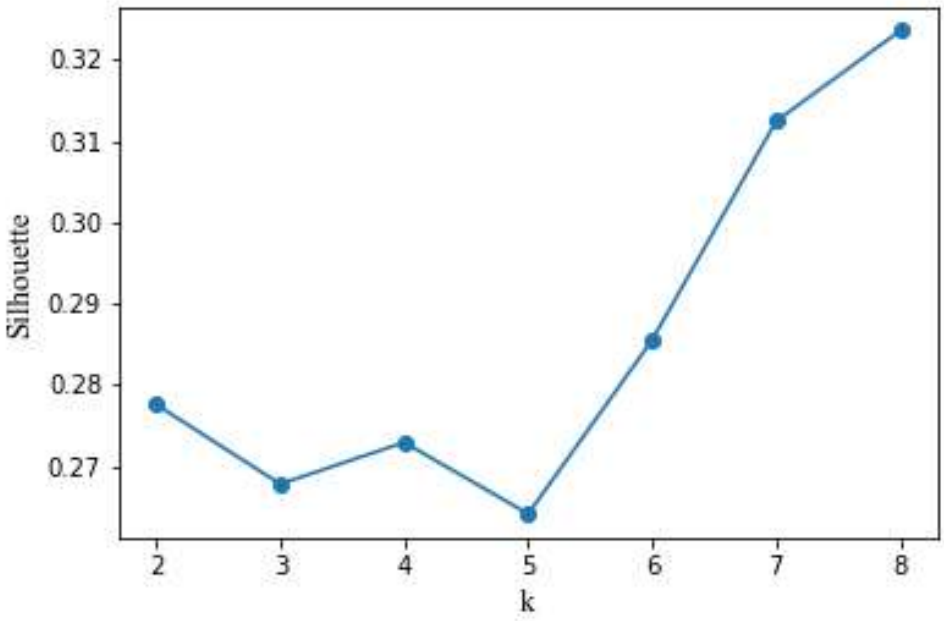


Figure 3.6: The relationship between k and Silhouette

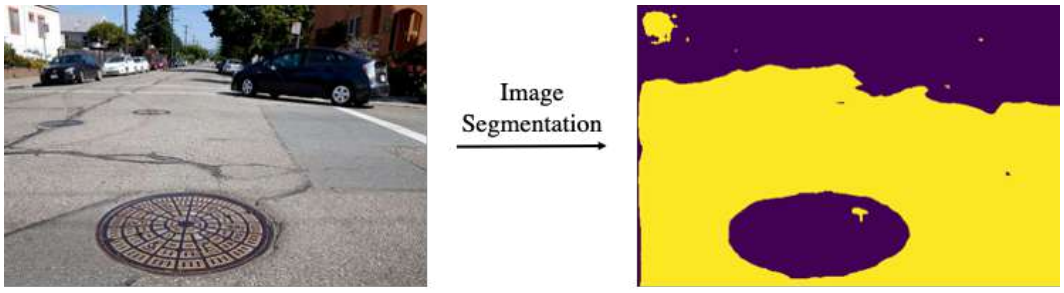


Figure 3.7: The input image and the result of segmentation

Figure 3.8: The final result of image segmentation

be viewed in Figure 3.8.

3.4 Ellipse Fitting

Processes above have divided the input natural image into multiple regions based on different textural features. As is mentioned in the architecture, ellipses will be extracted from the segmented image after the process of ellipse fitting. There are two methods considered when fitting the segments into ellipses – Random Sample Consensus and Least-Squares method.

3.4.1 Random Sample Consensus (RANSAC)

The Random Sample Consensus algorithm, which is proposed by Fischler and Bolles in 1981 and introduced [26] and applied in [27], is a re-sampling technique that generated candidate solutions by using the fewest data points required to estimate the underlying model parameters.

The implementation of this algorithm could be conducted by the following steps: 1) Selecting several data points randomly and building a model to fit with them; 2) Verifying with other data points and determining if this model is appropriate; 3) Comparing all the appropriate and selecting the best one as the final result.

The minimum number of data points that could be used to define an ellipse in a two-dimensional space is 5 (introduced in Section 3.5.1). For a segmented image, the set of data points consists of all the points lying in the edges. Five points in the set

Figure 3.9: The result of best performed model from RANSAC algorithm

will be selected randomly to define an ellipse. Then, all the other points will vote for the ellipse defined according to the distance of each point to the ellipse. The number of all the positive voters determines the performance of the model. When the number of program executions reaches the specified number of iterations, the model with best performance will be adopted as the final solution. A sample used to show the result of RANSAC algorithm could be viewed in Figure 3.9, in which the size of the input image is 259×194, the curve of ellipse represents the best performed one, points around the curve represent the positive voter.

Although RANSAC algorithm has a strong robustness (i.e. the model could be estimated even with a large number of outliers), the disadvantages are more significant. In theory, the optimal solution can be found it the number of iterations is not limited. The size of the sample image is too small, which provides a small number of data points, but normally, the edge of the whole image contains more than $1,500$. To find the best-performed model, the maximum number of program execution is more than 6.2×10^{13}. On the contrary, if the number of iterations is limited (e.g. 500), it is almost impossible to find the best-performed one regarding to the maximum number of program execution. Figure 3.10 shows several badly performed results using the same segmented image input when the number of iterations is limited, in which the curves of ellipses are unacceptable results from RANSAC algorithm, and the points represent the positive voters for the curve.

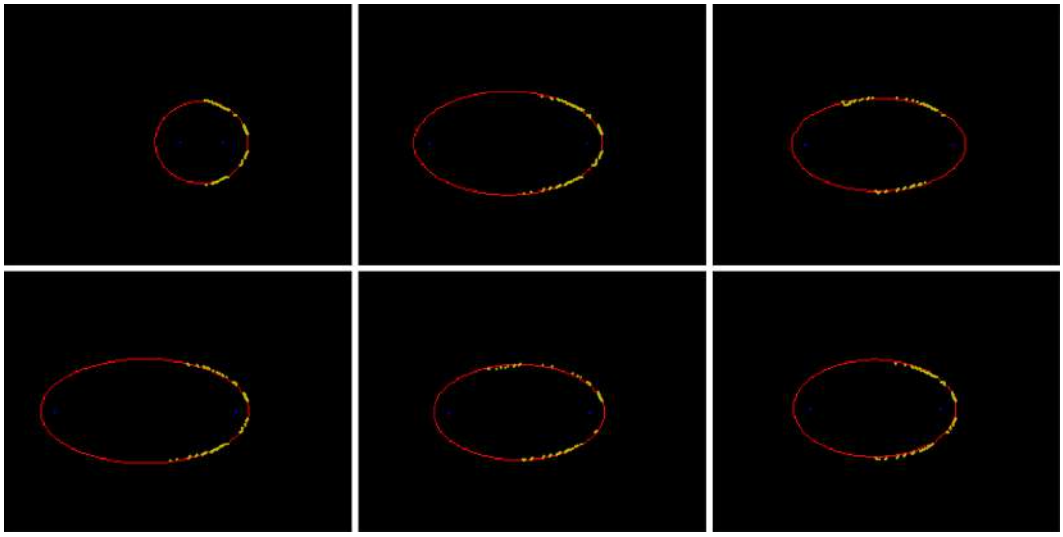


Figure 3.10: Several unacceptable results from RANSAC algorithm

3.4.2 Least-Squares (LS) Method

Open Source Computer Vision Library (OpenCV) provides a method for fitting an ellipse to a set of points based on a least-squares (LS) fitness function [28], according to the algorithm published by Andrew and Robert [29].

The basic idea of LS method is to solve the characteristic equation defined by all the points in the edge of segmented image input with the ellipse equation, according to Lagrange multiplier.

Compared with RANSAC algorithm, ellipse fitting provided by OpenCV is less computationally expensive, so it runs much faster. On the other hand, this approach uses all the data points, which ensures no key point with essential and influential information to the ellipse is missed, although the noise will slightly reduce the accuracy of the solution. Besides, the segmented images have less noise when detecting the edge information, which decrease the influence caused by the reduction of accuracy in this approach. Therefore, LS method provided by OpenCV is adopted to fit the segments to ellipses.

3.5 Filtering with Conditions

The conditions used could be divided into two categories: conditions based on the geometric properties of ellipses and conditions based on actual situations of manhole covers. The former restricts the figure detected to be an ellipse, while the latter avoids recognition of other elliptical or circular objects that are not manhole covers (e.g. vehicle tires).

3.5.1 Geometric Properties of Ellipses

Several geometric properties have been discussed and used to detect ellipses in several previous papers [30][31]. The main idea contains three aspects: 1) the symmetry of ellipse, 2) the aspect ratio of the ellipse, and 3) the distance of edge points to two focal points of the ellipse.

As is known and introduced in related papers and studies mentioned above, the equation of an ellipse in a two-dimensional space could be expressed as Equation 3.5 shows, in which A, B, C, D and E are coefficients and x and y are variables.

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + 1 = 0 \tag{3.5}$$

Accordingly, the position of the ellipse center $O(x_0,\ y_0)$, the length of major and minor axes ($2a$ and $2b$) and the rotation angle θ could be calculated by Equation 3.6, 3.7 and 3.8 respectively.

$$\begin{cases} x_0 = \frac{BE-2CD}{4AC-B^2} \\ y_0 = \frac{BD-2AE}{4AC-B^2} \end{cases} \tag{3.6}$$

$$\begin{cases} a^2 = \frac{2\left(Ax_0^2+Cy_0^2+Bx_0y_0-1\right)}{A+C-\sqrt{(A-C)^2+B^2}} \\ b^2 = \frac{2\left(Ax_0^2+Cy_0^2+Bx_0y_0-1\right)}{A+C+\sqrt{(A-C)^2+B^2}} \end{cases} \tag{3.7}$$

$$\theta = \frac{1}{2}\arctan\frac{B}{A-C} \tag{3.8}$$

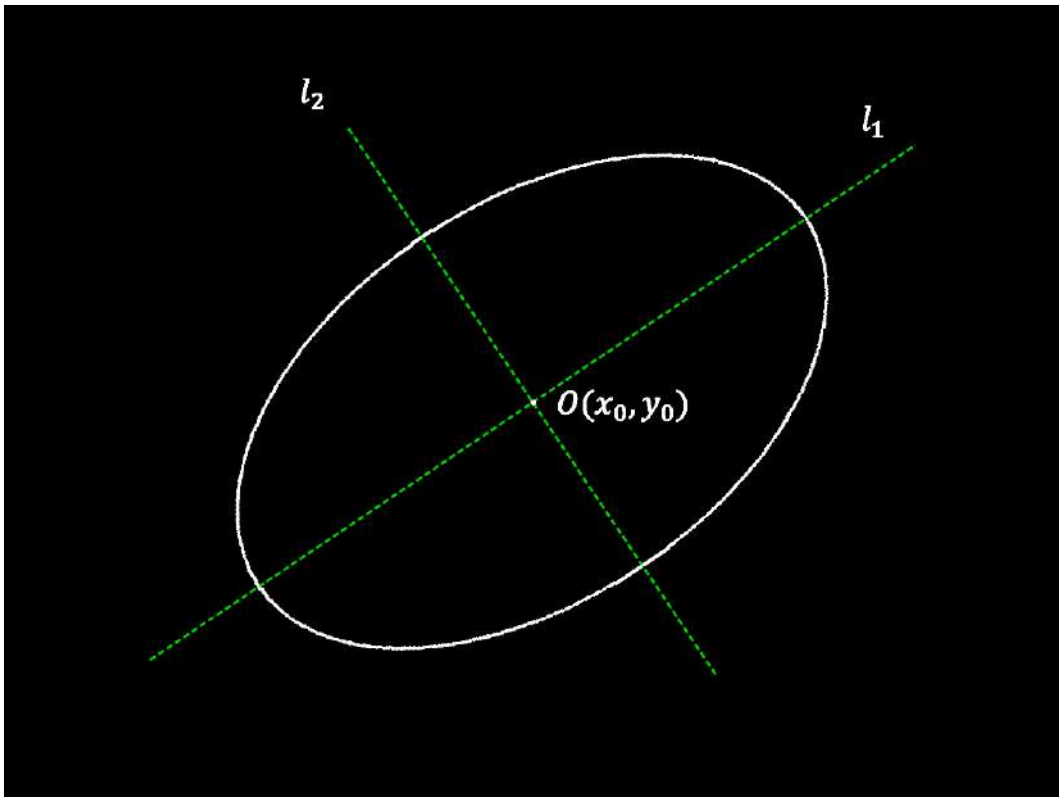


Figure 3.11: The symmetry of an ellipse

Symmetry

An ellipse is an axisymmetric figure with two symmetry axes, where the major and minor axes are lying. At the same time, the ellipse is a centrally symmetric figure [32].

Once a region is fitted to an ellipse, the directions of its major and minor axes could be calculated based on the rotation angle. That is to say, the symmetry of the region could be examined according to the center of the ellipse and the major and minor axes. Regions with weak symmetry will not be adopted as the final results.

Figure 3.11 shows the symmetry of an ellipse, in which $O(x_0,\ y_0)$ is the center of symmetry and l_1 and l_2 are the lines of symmetry.

Aspect Ratio

For an ellipse, the aspect ratio refers to the ratio of the major axis to the minor axis (i.e. a / b). Several ellipses with different aspect ratios are shown in Figure 3.12, the

value of aspect ratio is 1, 2, 3, 4 and 5 respectively for each ellipse from left to right.

It could be seen in the figure that, if we locate the major axis of an ellipse in horizontal direction, as the value of aspect ratio increases, the ellipse looks slimmer and slimmer in vertical direction and the area of the ellipse is becoming smaller and smaller. When aspect ratio is larger than 4, the geometric information in vertical direction has become indistinguishable. That means the regions in the input image, from which the ellipses are fitted, are not ideal for ellipse fitting. Therefore, the ellipses will be excluded from the candidates.

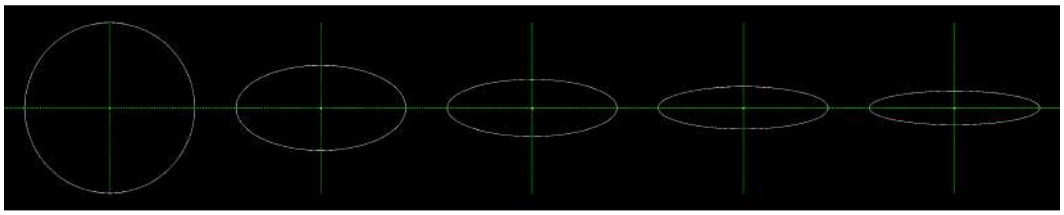


Figure 3.12: Ellipses with different aspect ratios

Distance to Focal Points

The sum of distances between any point P, which is lying on the edge of ellipse E, and two focal points of the ellipse F_1 and F_2 is a constant. Accordingly, a distance threshold Dis could be set to examine whether a point fits the curve of ellipse well, as is shown in Figure 3.13, in which P_0 is a point lying on the edge of ellipse while P_1 and P_2 are not, the sum of the Euclidean distances of P_0 to two focal points F_1 and F_2 is used as the basic distance D_0, then the distance Dis threshold could be defined with Equation 3.9, in which ξ represents the tolerance of the distance value.

$$Dis = D_0 \pm \xi \tag{3.9}$$

A certain edge point of the region will be marked as an opponent if the sum of distances of this point to focal points is larger than Dis, and if the majority of all the edge points of this region are marked as opponents, the ellipse defined by this region will be removed from the candidates.

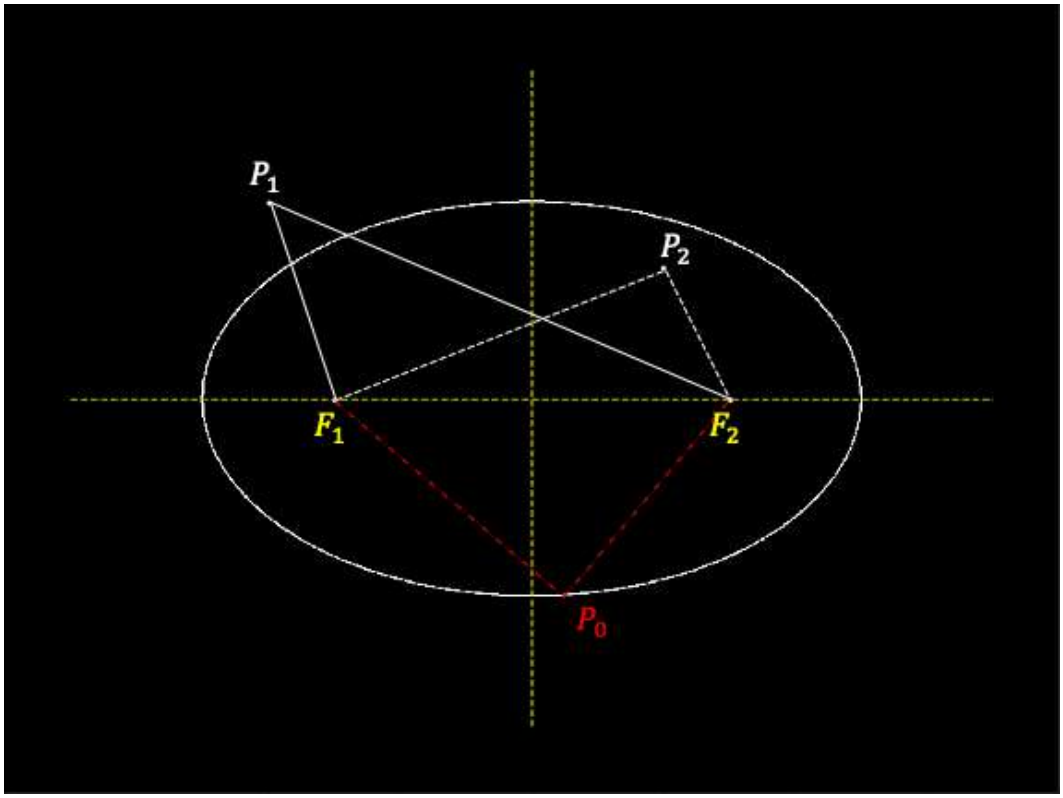


Figure 3.13: Distances of points to an ellipse in a 2-dimensional space

3.5.2 Actual Situation of Manhole Covers

The candidates meeting the conditions of geometric properties of ellipse has a great similarity to an ellipse. However, there are still some oval items in the natural images (e.g. wheels or tires of vehicles). These items will be incorrectly regarded as potential ellipses with the processes introduced above, so two conditions based on actual situation of manhole covers are used to remove them from the candidates.

Area Limitation

Oval items of which the sizes are too small in the input image are not reliable for locating or route planning. Besides even though those oval items are manhole covers, they will not influence the location or route immediately because tiny size means a long distance between the camera to the item according to principles of perspective.

Therefore, regions with size smaller than one percent of the image size will be

excluded in this dissertation.

Direction of Major Axis

According to the principles of perspective, a manhole cover on the road looks like an ellipse of which the major axis is lying in the horizontal direction in a natural image. In different situations, the direction of ellipses might not be totally horizontal, so a tolerance of angle is used to select ellipses closer to manhole covers regarding the direction. In this dissertation, the ellipses will be removed if the angle between its major axis and the horizontal line is larger than 45°.

3.6 Summary

The manhole covers in a natural image will be detected with the processes introduced above, including texture feature extraction, image segmentation, ellipse fitting and filtering.

In the process of texture feature extraction, the input image is transferred into $L * a * b*$ color space and medium filter is used for three channels. The gray-level image is convolved with a set of LoG filters with different parameters. The results are divided into several bins in histograms according to the intensity respectively. A window is used to reflects the texture feature of its center point by counting the size of every bin in the histogram for every result.

Then, the image is segmented with K-means algorithm, of which the number of clusters is mainly determined by elbow method according to SSE value. Flood Fill algorithm is used to the output to distinguish regions incorrectly divided, and the tiny regions are removed to reduce their effects to the final result.

Regions segmented are fitted to ellipse with Least-Squares method used and provided in OpenCV. At last, the candidate ellipses are examined with conditions predefined in terms of geometric properties of ellipse and the actual situation of manhole covers. The final results will be shown in the next chapter.

Chapter 4

Results and Discussions

This chapter mainly focuses on the results from the method introduced in Chapter 3. The description of the dataset and discussions about the results are also demonstrated here.

4.1 Dataset

At first, image recognition methods based on Machine Learning (ML) such as Convolutional Neural Network (CNN) have also been taken into consideration. However, the training of the model or neural network requires a dataset containing tens of thousands of pictures, but unfortunately, no such dataset could be found on the Internet. On the other hand, it is extremely difficult to establish such a dataset in a limited time. Therefore, methods based on ML were abandoned.

After the method based on feature extraction being adopted, the requirement of number of pictures in the dataset decreased a lot, but there are too few natural images containing manhole covers on the Internet, so an appropriate dataset needs to be established firstly.

After selecting limited number of natural images on the Internet, 21 of them were added into the dataset. At the same time, the dataset was expanded by several pictures taken with mobile phones. To acquire adequate and suitable pictures, the limitation of the distance between mobile phones and the manhole covers and the directions of lens shall be too strict or too loose. Finally, 26 out of over 40 pictures were selected to

expand the dataset. In a word, the dataset consists of 21 natural images selected from pictures on the Internet and 26 pictures taken with mobile phones.

The images in dataset were then divided into four groups according to the ratio of the area of ellipse to the image size. The range of ratio and the number of pictures in each group could be seen in Table 4.1.

Table 4.1: The attribute of each group

Group ID	Range of Ratio	Number of Pictures
Large	more than 20%	8
Medium	8% to 20%	12
Small	1% to 8%	14
Tiny	less than 1%	13

Some examples of pictures in Large group are shown in Figure 4.1. As is shown in Table 4.1, in these images, the ratio of area of the manhole cover to the image size is more than 20%. Manhole covers in picture (a) and (b) could be distinguished visually, so the outputs are more accurate for them. But the texture of the manhole cover surfaces in picture (c) is the same as their surroundings, thus the result of that is predictably worse than those of picture (a) and (b). For picture (d), the size of manhole cover is appropriate but the texture feature seems quite similar to that of the surroundings influenced by the noise.

In Figure 4.2, there are several pictures in Medium group shown as examples. For these pictures the ratio is between 8% and 20%. In picture (a), (b), (d), (e) and (f) in Figure 4.2, the texture feature of manhole covers is distinguishable to that of the road surface, so the outputs for them are pretty good and acceptable accordingly. But for picture (c), the cracks on the road surface increases the difficulty of manhole cover detection due to the great similarity between the manhole cover and its surroundings. Therefore, the output is not as good as those of others shown below.

Some samples pictures in Small group are shown in Figure 4.3, in which the manhole covers in picture (c) has the similar texture feature to the road surface, while for the manhole covers in other pictures, there exists a great difference between their texture features and the road surface. For some pictures (e.g. picture (d)), both the color and the texture of the manhole cover are quite distinguishable to its surroundings. Accordingly, the result of picture (d) is much better than the output for picture (c),

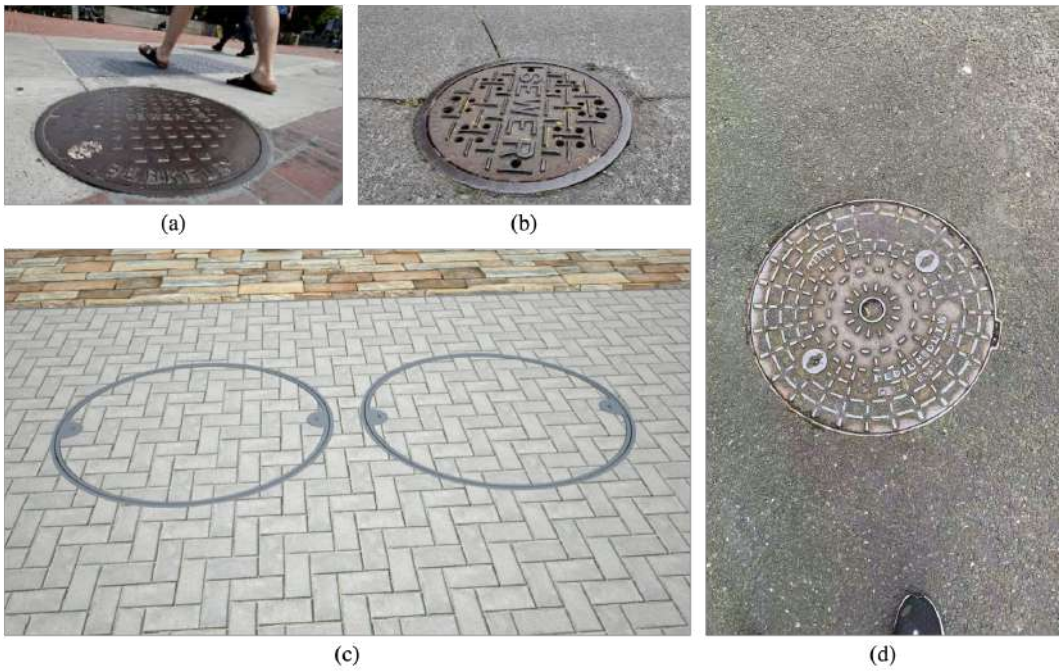


Figure 4.1: Samples of pictures in Large group

while the result of (b) is also acceptable. For picture (a), it is easier to extract the manhole cover based on the color information. But the texture information of the manhole cover is almost the same as the blocks around it, so the output for picture (a) is not as good as that of (b) using the method proposed.

In Figure 4.4, the pictures in Tiny group shown below cannot be detected due to the area condition used in filtering processing introduced in Section 3.5.2.

4.2 Evaluations

This dissertation aims at the detection of manhole covers, so for every single manhole cover in the natural image, a groundtruth is defined manually. The shape of the manhole cover is an ellipse, which could be defined with five parameters in a 2-dimensional space (i.e. the position of center $O(x_0,\ y_0)$, the length of major and minor axes ($2a$ and $2b$) and the rotation angle θ). Accordingly, the groundtruth is stored with five parameters mentioned above in the format of $((x_0,\ y_0),\ (a,\ b),\ \theta)$.

For every ellipse detected in a certain image, the groundtruth will be transferred

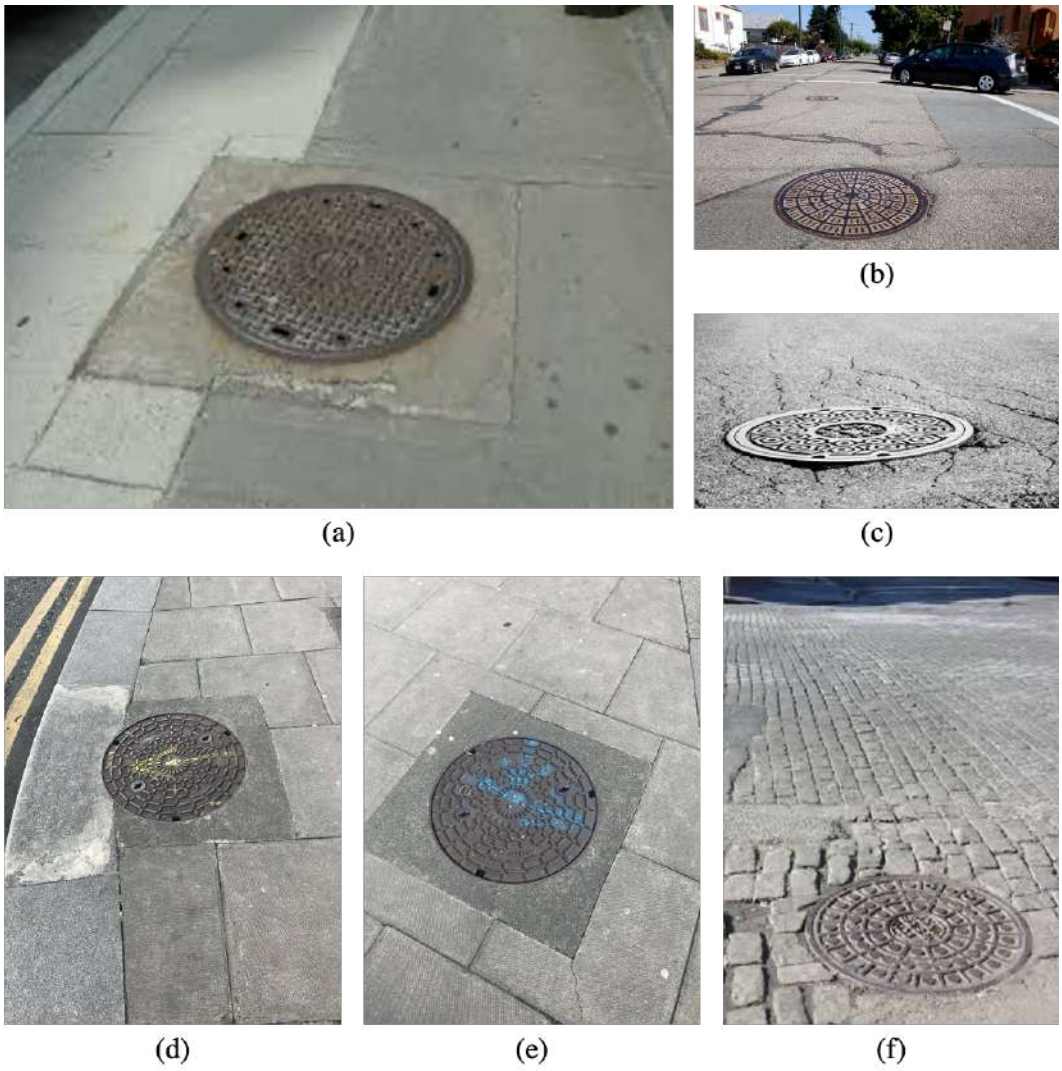


Figure 4.2: Samples of pictures in Medium group

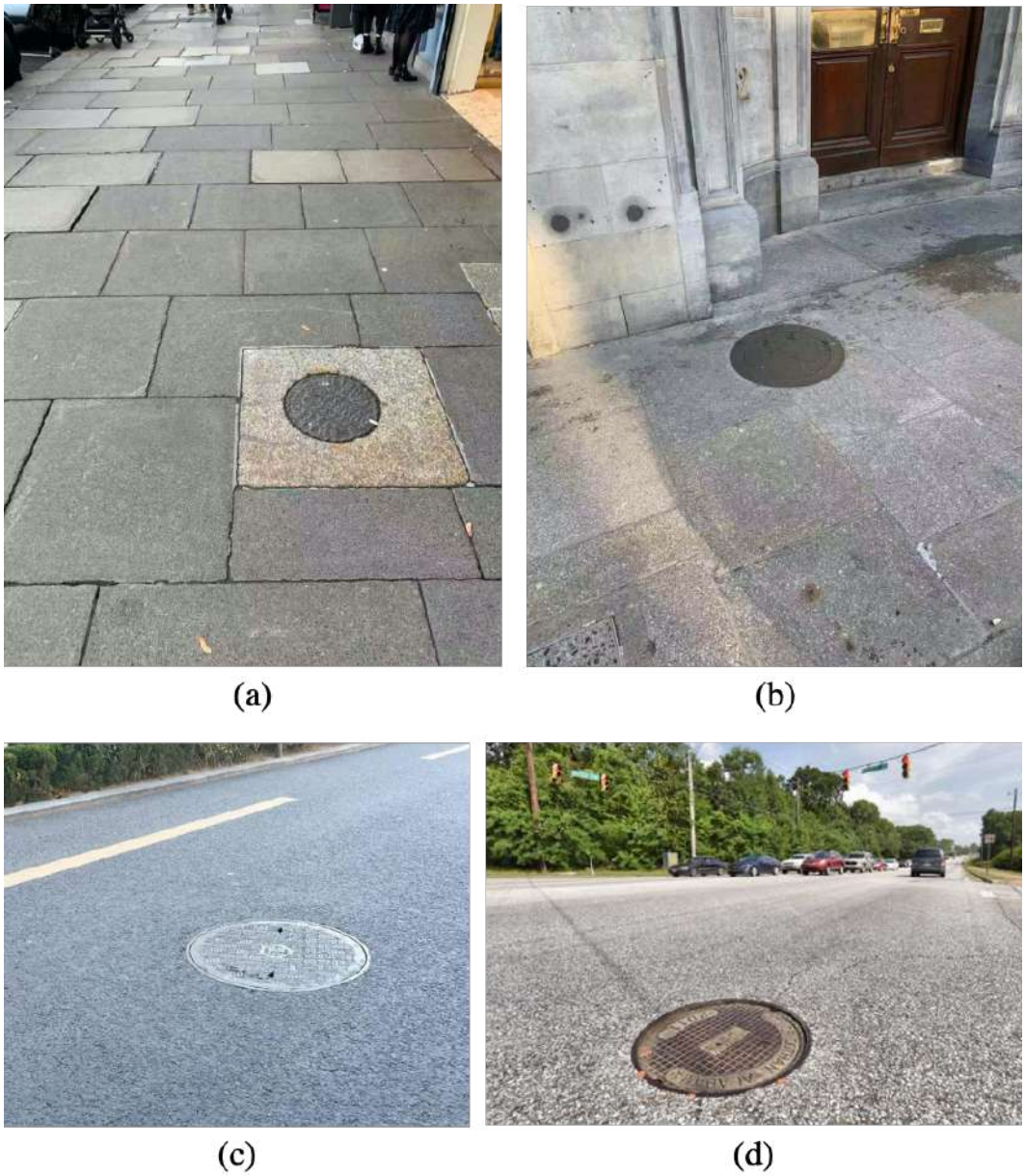


Figure 4.3: Samples of pictures in Small group

(a) (b)

Figure 4.4: Samples of pictures in Tiny group

into an image. Then the IoU (Intersection over Union) value will be calculated to determine if this ellipse is detected correctly. That is to say, for a single image, there are two sets of ellipses – groundtruth set (GT) and fitted set (FIT).

If one ellipse is in both GT and FIT, it will be marked as a True Positive (TP) result. If the ellipse is only in GT but not in FIT, the label of that will be False Negative (FN). On the contrary, ellipses only in FIT but not in GT will be marked as False Positive (FP) results. True Negative (TN) result represents an ellipse does not appear in neither GT nor FIT. But there is at least one ellipse in every single image in the dataset, so TN always equals to 0.

Based on the definitions of TP, FP, FN and TN results, the definitions of evaluations (i.e. *Recall*, *Precision*, *Accuracy* and F_1 *score*) could be explained well according to the study [33].

Recall is the ratio of the number of TP results to the number of all ellipses in GT. It could be calculated with Equation 4.1.

$$Recall = \frac{TP}{TP + FN} \tag{4.1}$$

Precision is the ratio of the number of TP results to the number of all ellipses in FIT, which could be acquire with Equation 4.2.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

Accuracy is the ratio of the number of correctly predicted results to the sum of the results. In this dissertation, it could be calculated with Equation 4.3.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \overset{TN=0}{=} \frac{TP}{TP + FP + FN} \tag{4.3}$$

F1-score is the weighted average of Precision and Recall. It is the most important factor among these four evaluations because it removes the effect of absence of the number of TN results, and it could be acquired with Equation 4.4.

$$F_1\ score = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} = \frac{2TP}{2TP + FP + FN} \tag{4.4}$$

4.3 Results

As is introduced, Gabor filter and LoG filter are used respectively in the processing of texture feature extraction. The outputs of methods based on them are shown in Table 4.3 and Table 4.3 respectively in terms of *Recall*, *Precision*, *Accuracy* and F_1 *score*.

Table 4.2: Results of the approach with Gabor Filters used in texture feature extraction

Group ID	*Recall*	*Precision*	*Accuracy*	F_1 *score*
Large	44.4%	44.4%	28.6%	44.4%
Medium	**46.2**%	**46.2**%	**30.0**%	**46.2**%
Small	35.7%	29.4%	19.2%	32.2%
Tiny	0.0%	0.0%	0.0%	Nan

Table 4.3: Results of the approach with LoG Filters used in texture feature extraction

Group ID	*Recall*	*Precision*	*Accuracy*	F_1 *score*
Large	44.4%	44.4%	28.6%	44.4%
Medium	62.5%	**53.3**%	**40.0**%	**57.1**%
Small	**64.3**%	45.0%	36.0%	52.9%
Tiny	0.0%	0.0%	0.0%	Nan

In the tables above, the maximum of each column has been marked with bold. And there are 6 examples of the final results of the system in Figure 4.5.

4.4 Discussions

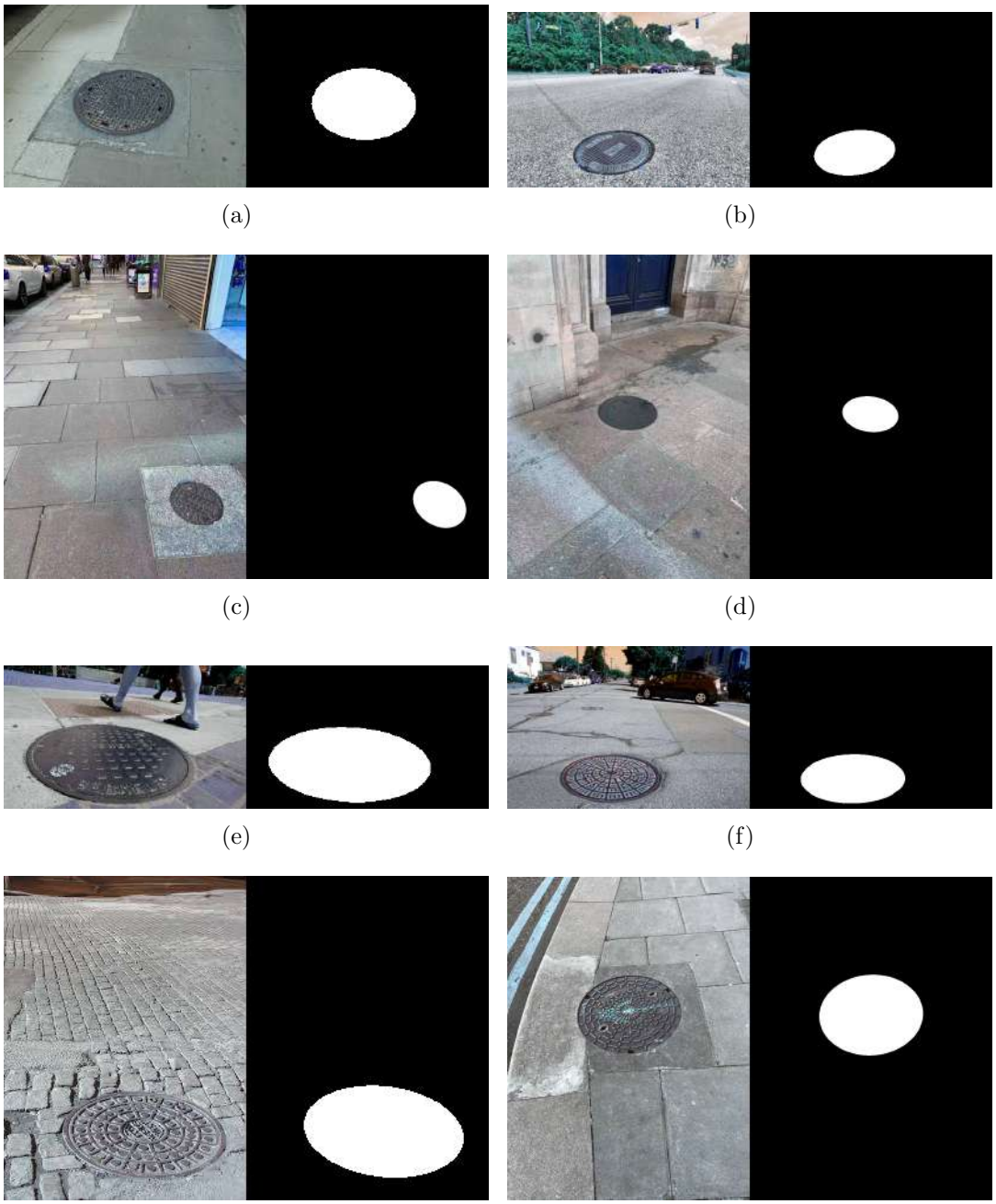
Comparing these Table 4.3 and Table 4.3, it could be seen that LoG filter performs better than Gabor filter in this system. Gabor and LoG Filter both perform well in Medium group, because most of maximums of different columns are lying in the row of Medium group.

For images in Large group, their performances are quite similar to each other. The reason of the values being less than those in Medium group is that if the ratio is large enough, the effect of noise to the texture feature tends to increase a lot, which will influence the results of segmentation. In other words, the image will be over-segmented because the noise adds more details in the region of the manhole cover.

For images in Small group, K-means algorithm will focus more on the other regions rather than the ellipse with its surroundings. It means there might be not any center of cluster lying in the region of the manhole cover, and in these images, manhole covers tend to be mixed up with their surroundings.

For images in Tiny group, as is discussed in Section 3.5.2, the regions with an area smaller than one percent of the image size will be removed, so it is reasonable to get no output.

Besides, the results in Large group and Medium group in Table 4.3 are quite close to each other, while in Table 4.3, it happens on the Medium group and Small group. That reflects the difference of performance of Gabor and LoG filter in texture feature extractions for different images.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 4.5: Final results of the system

Chapter 5

Conclusion and Future Work

The conclusion and contributions of this dissertation is discussed in Section 5.1, while the limitations and improvements that could be conducted in the future are discussed in Section 5.2.

5.1 Conclusions

The objective of this dissertation is to detect manhole covers from natural images in the dataset. For every single imaged in the dataset, a groundtruth is established to describe the information of the ellipses in this image in terms of the position of center point, the major and minor axes and the rotation angle.

The dataset consists of 47 pictures in total, and it is divided into four groups according to the ratio of the area of each ellipse to the image size. The performance of the method proposed in this dissertation is determined by the relationship between the results and the ellipses of groundtruth.

For every single natural image, the texture feature is extracted by calculating the convolution of this image and a set of filters pre-defined. Three channels of L*a*b* color space have been denoised using medium filter, and Gabor filter and Laplacian of Gaussian filter are used respectively to be convolved with the gray-level image of the input. The outputs are used as the feature images.

A window with a pre-defined size is used to traverse the all the feature images, and the texture feature of the center point of the window is represented by the distribution

of the bins in histogram of the feature images.

According to the texture features, the input image is segmented into several regions with K-means algorithm, of which the number of clusters is determined by elbow method based on Sum of Square Error. Flood fill algorithm is used to distinguish unconnected regions with similar features, and meaningless tiny regions are removed to reduce the effects.

Every single region is fitted to an ellipse with Least-Square method, but the ellipses will be removed if they do not meet pre-defined conditions in terms of geometric properties of ellipse and actual situations of manhole covers.

The final results of approaches using Gabor filter are compared with those of method using LoG filter. Although they both performs well in Medium group, the performance of LoG filter is better than the other.

This dissertation provides a method to detect the manhole covers in natural image, which will be helpful to locate the position of the vehicles and plan routes.

5.2 Future Works

As is introduced and discussed above, the method proposed in this dissertation could still be improved in several aspects.

The filters used to extract the texture feature determined the output. The parameters used, the number of filters, and even the type of filters could be considered and determined more appropriately. The combination of filters provides unlimited methods that could be used in different fields.

Another element that could be improved is the selection of clustering algorithm and the determination of number of the clusters. K-means is used in this dissertation because of the high efficiency, but there are also several other clustering algorithms that could be tried.

Besides, the number of the images in the dataset is limited, so more images shall be added into the dataset to increase the reliability and robustness of this approach.

Bibliography

[1] B. Schoettle and M. Sivak, "Public opinion about self-driving vehicles in china, india, japan, the u.s., the u.k., and australia," Tech. Rep. UMTRI-2014-30, The University of Michigan, Transportation Research Institute, 2901 Baxter Road, Ann Arbor, Michigan 48109-2150 U.S.A., oct 2014.

[2] S. Lambert, "5 top autonomous vehicle companies to watch in 2020," *MES Insights*, Mar. 25, 2020.

[3] J. D. Lee and K. Kolodge, "Exploring Trust in Self-Driving Vehicles Through Text Analysis," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 62, pp. 260–277, Mar. 2020.

[4] M. Hirz and B. Walzel, "Sensor and object recognition technologies for self-driving cars," *Computer-Aided Design and Applications*, vol. 15, pp. 501–508, July 2018.

[5] A. Raza, S. Hameed, and T. Macintyre, "Global Positioning System – Working and its Applications," in *Innovations and Advanced Techniques in Systems, Computing Sciences and Software Engineering* (K. Elleithy, ed.), pp. 448–453, Dordrecht: Springer Netherlands, 2008.

[6] M. R. Islam and J.-M. Kim, "An Effective Approach to Improving Low-Cost GPS Positioning Accuracy in Real-Time Navigation," *The Scientific World Journal*, vol. 2014, pp. 1–8, 2014.

[7] A. Y. Lin, K. Kuehl, J. Schöning, and B. Hecht, "Understanding "Death by GPS": A Systematic Study of Catastrophic Incidents Associated with Personal Navigation Technologies," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, (Denver Colorado USA), pp. 1154–1166, ACM, May 2017.

[8] L. L. Arnold and P. A. Zandbergen, "Positional accuracy of the Wide Area Augmentation System in consumer-grade GPS units," *Computers & Geosciences*, vol. 37, pp. 883–892, July 2011.

[9] J. Levinson, J. Askeland, J. Dolson, and S. Thrun, "Traffic light mapping, localization, and state detection for autonomous vehicles," in *2011 IEEE International Conference on Robotics and Automation*, pp. 5784–5791, May 2011. ISSN: 1050-4729.

[10] D. A. Alghmgham, G. Latif, J. Alghazo, and L. Alzubaidi, "Autonomous Traffic Sign (ATSR) Detection and Recognition using Deep CNN," *Procedia Computer Science*, vol. 163, pp. 266–274, 2019.

[11] R. Maini and H. Aggarwal, "Study and comparison of various image edge detection techniques," *International Journal of Image Processing*, vol. 3, 2009.

[12] L. G. Shapiro and G. C. Stockman, "Image segmentation," in *Computer Vision*, pp. 279–325, Englewood Cliffs, NJ, USA: Prentice Hall, 2001.

[13] M. H. Bharati, J. Liu, and J. F. MacGregor, "Image texture analysis: methods and comparisons," *Chemometrics and Intelligent Laboratory Systems*, vol. 72, pp. 57–71, June 2004.

[14] S. Singh, D. Srivastava, and S. Agarwal, "GLCM and its application in pattern recognition," in *2017 5th International Symposium on Computational and Business Intelligence (ISCBI)*, (Dubai, United Arab Emirates), pp. 20–25, IEEE, Aug. 2017.

[15] S. K. Kashyap and J. K. Mishra, "A Review Paper on Gabor Filter Algorithm for The application of Texture Segmentation," *IJARCCE*, vol. 6, pp. 476–480, Apr. 2017.

[16] R. Roslan and N. Jamil, "Texture feature extraction using 2-d gabor filters," *ISCAIE 2012 - 2012 IEEE Symposium on Computer Applications and Industrial Electronics*, pp. 173–178, 12 2012.

[17] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, pp. 530–49, 06 2004.

[18] J. Yuan, D. Wang, and A. M. Cheriyadat, "Factorization-Based Texture Segmentation," *IEEE Transactions on Image Processing*, vol. 24, pp. 3488–3497, Nov. 2015.

[19] X. Liu and D. Wang, "A spectral histogram model for texton modeling and texture discrimination," *Vision Research*, vol. 42, pp. 2617–2634, Oct. 2002.

[20] X. Liu and D. Wang, "Image and Texture Segmentation Using Local Spectral Histograms," *IEEE Transactions on Image Processing*, vol. 15, pp. 3066–3077, Oct. 2006.

[21] S. Mannor, X. Jin, J. Han, X. Jin, J. Han, X. Jin, J. Han, and X. Zhang, "K-Means Clustering," in *Encyclopedia of Machine Learning* (C. Sammut and G. I. Webb, eds.), pp. 563–564, Boston, MA: Springer US, 2011.

[22] J. Jamesmanoharan, S. Ganesh, M. Felciah, and A. Shafreenbanu, "Discovering students' academic performance based on gpa using k-means clustering algorithm," *Proceedings - 2014 World Congress on Computing and Communication Technologies, WCCCT 2014*, pp. 200–202, 02 2014.

[23] M. A. Syakur, B. K. Khotimah, E. M. S. Rochman, and B. D. Satoto, "Integration K-Means Clustering Method and Elbow Method For Identification of The Best Customer Profile Cluster," *IOP Conference Series: Materials Science and Engineering*, vol. 336, p. 012017, Apr. 2018.

[24] F. Wang, H.-H. Franco-Penya, J. D. Kelleher, J. Pugh, and R. Ross, "An Analysis of the Application of Simplified Silhouette to the Evaluation of k-means Clustering Validity," in *Machine Learning and Data Mining in Pattern Recognition* (P. Perner, ed.), vol. 10358, pp. 291–305, Cham: Springer International Publishing, 2017.

[25] S. Torbert, *Applied Computer Science*. Cham: Springer International Publishing, 2016.

[26] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, pp. 381–395, June 1981.

[27] X. Tan, J. Yang, and X. Deng, "Filtering method of star control points for geometric correction of remote sensing image based on RANSAC algorithm," in *Ninth International Conference on Graphic and Image Processing (ICGIP 2017)* (H. Yu and J. Dong, eds.), (Qingdao, China), p. 209, SPIE, Apr. 2018.

[28] A. Kaehler and G. R. Bradski, "Contours," in *Learning OpenCV 3: computer vision in C++ with the OpenCV library*, pp. 424–425, Sebastopol, CA: O'Reilly Media, first edition, second release ed., 2017. OCLC: ocn968936315.

[29] A. Fitzgibbon and R. Fisher, "A Buyer's Guide to Conic Fitting.," in *Procedings of the British Machine Vision Conference 1995*, pp. 51.1–51.10, British Machine Vision Association, 1995.

[30] H. Liu and Z. Wang, "Geometric property based ellipse detection method," *Journal of Visual Communication and Image Representation*, vol. 24, pp. 1075–1086, Oct. 2013.

[31] L. Libuda, I. Grothues, and K.-F. Kraiss, "Ellipse Detection in Digital Image Data Using Geometric Features," in *Advances in Computer Graphics and Computer Vision* (J. Braz, A. Ranchordas, H. Araújo, and J. Jorge, eds.), vol. 4, pp. 229–239, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[32] C.-T. Ho and L.-H. Chen, "A fast ellipse/circle detector using geometric symmetry," *Pattern Recognition*, vol. 28, pp. 117–124, Jan. 1995.

[33] C. Goutte and E. Gaussier, "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation," in *Advances in Information Retrieval* (D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, D. E. Losada, and J. M. Fernández-Luna, eds.), vol. 3408, pp. 345–359, Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

Appendix

Abbreviations

AD	Autonomous Driving
ASM	Angular Second Moment
CNN	Convolutional Neural Network
FN	False Negative
FP	False Positive
GLCM	Gray Level Co-occurrence Matrix
GPS	Global Positioning System
IDM	Inverse Differential Moment
IoU	Intersection over Union
LoG	Laplacian of Gaussian
LS	Least-Squares
ML	Machine Learning
RANSAC	Random Sample Consensus
SDV	Self-Driving Vehicle
SSE	Sum of Square Error
TN	True Negative
TP	True Positive