AUTOENCODER-BASED KEY-FRAME IDENTIFICATION FOR WATER PIPELINE CCTV INSPECTION

by

Yutong Jiao

B.Eng., Beijing University of Chemical Technology, 2018

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE COLLEGE OF GRADUATE STUDIES

(Electrical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Okanagan)

February 2021

© Yutong Jiao, 2021
The following individuals certify that they have read, and recommend to the College of Graduate Studies for acceptance, a thesis entitled:

**AUTOENCODER-BASED KEY-FRAME IDENTIFICATION FOR WATER PIPELINE CCTV INSPECTION**

submitted by Yutong Jiao in partial fulfilment of the requirements of the degree of Master of Applied Science

Dr. Zheng Liu, School of Engineering
**Supervisor**

Dr. Kenneth Chau, School of Engineering
**Supervisory Committee Member**

Dr. Yang Cao, School of Engineering
**Supervisory Committee Member**

Dr. Joshua Brinkerhoff, School of Engineering
**University Examiner**
Abstract

Closed-circuit television (CCTV) is being widely adopted in water pipeline inspection. The inspector needs to spend a long time to watch the recorded video during the office-based survey and can get fatigue easily. An automated process can release the inspector’s workload and ensure the consistent quality of the survey. However, a fully automated survey of varied structural discontinuities still remains as a challenge. This study aims to first identify the key frames of the CCTV video, which contain the major anomalies captured from the internal surface of the pipe. Thus, the inspector can focus more on these key frames.

First, the key-frame identification task is performed by using an unsupervised learning-based framework with state-of-the-art autoencoders. Three popular autoencoders were implemented in this framework, and results were collected accordingly. The experimental results illustrate that this framework achieves up to 0.940 accuracy with a relatively lower precision metric of 0.901.

Secondly, a key-frame identification framework based on the steerable pyramid autoencoder (SPAE) is proposed in order to improve the accuracy of this task. Both the parameter optimization and comparative studies for the proposed SPAE were carried out in this research. Explicitly, the SPAE develops good capability of representation learning and extraction. The experimental results demonstrate that the SPAE-based approach can achieve 0.984 accuracy, which outperforms the other experimental methods selected for comparison.

Thirdly, a log-Gabor autoencoder (LGAE) based framework is proposed to further enhance the performance on key-frame identification in terms of representation learning. It can be regarded as an improved version of the SPAE-based approach. Comparative studies were conducted, and the results show that this novel LGAE takes advantage of the feature detection over the other methods with the accuracy of 0.988 and the recall metric of 0.996. Additionally, the LGAE has a superior performance for the generalization on varied datasets. Hence, LGAE-based framework will significantly assist the office-based survey, which will highly facilitate the pipeline condition assessment through the CCTV inspection.
Lay Summary

Closed-circuit television (CCTV) is being widely adopted in water pipeline inspection. To date, companies still assign inspectors to perform office-based survey on the recorded videos, which show the internal surface of the water pipeline. The inspectors need to spend a long time checking the videos manually, which is not only time-consuming but also strenuous. Also, the inspectors get fatigue easily. An automated process can release the inspectors’ workload and ensure the consistent quality of the survey. However, a fully automated survey of varied structural deficiencies still remains as a challenge. In this thesis, two key-frame identification frameworks are carried out based on the proposed steerable pyramid autoencoder (SPAE) and log-Gabor autoencoder (LGAE) to address this challenge. Thus, the inspector can focus more on these key frames. This will highly facilitate the pipeline condition assessment through the CCTV inspection. Experimental results demonstrate the effectiveness of the proposed methods.
Preface

This thesis is based on the research work conducted in the School of Engineering at The University of British Columbia, Okanagan Campus, under the supervision of Prof. Zheng Liu. Published work is contained in this thesis.

The following published paper is contained in the thesis and used with permission of Elsevier.


The author has contributed to the following published paper, which is used with permission of IEEE.


The author has contributed to the following published paper, which is used with permission of SPIE.


I am the principal contributor for the work in this thesis. Prof. Zheng Liu provided me with some suggestions to improve my research work. Besides, Mrs. Angie Wu, the senior manager at Pure Technologies, helped with the acquisition of the data used in the thesis.
# Table of Contents

Abstract ................................................................. iii

Lay Summary ............................................................. iv

Preface ................................................................. v

Table of Contents ......................................................... vi

List of Tables .......................................................... viii

List of Figures ........................................................ ix

Glossary .............................................................. xi

Acknowledgements ..................................................... xii

Chapter 1: Introduction ................................................ 1
  1.1 Background and Motivation ....................................... 1
  1.2 Literature Review ................................................ 3
    1.2.1 CCTV-based Condition Assessment of Water Pipeline .... 3
    1.2.2 CCTV Video Analytics Methodologies ....................... 4
  1.3 Research Challenges and Objectives ......................... 5
  1.4 Thesis Outline and Contributions .............................. 6

Chapter 2: Autoencoder Techniques: The State of the Art ..... 8
  2.1 Autoencoder Network ............................................ 9
  2.2 Convolutional Autoencoder Network ........................... 10
  2.3 Summary ....................................................... 13

Chapter 3: Autoencoder-based Key-Frame Identification Frameworks and Implementations ........................................ 14
  3.1 Introduction ..................................................... 14
TABLE OF CONTENTS

3.2 The Unsupervised Learning-based Framework .................................. 15
  3.2.1 Framework Structure ......................................................... 15
  3.2.2 Computational Methodologies ............................................ 15
3.3 The Steerable Pyramid Autoencoder based Framework ....................... 19
  3.3.1 Framework Structure ......................................................... 19
  3.3.2 Computational Methodologies ............................................ 20
3.4 The Log-Gabor Autoencoder based Framework .................................. 29
  3.4.1 Framework Structure ......................................................... 29
  3.4.2 Computational Methodologies ............................................ 30
3.5 Evaluation Metrics .................................................................... 35
3.6 Summary ................................................................................. 38

Chapter 4: Experimental Results and Discussions ................................... 39
  4.1 Data Description ..................................................................... 39
  4.2 Experimental Setup ............................................................... 39
  4.3 Results and Discussions ........................................................ 40
    4.3.1 The Unsupervised Learning-based Framework ....................... 40
    4.3.2 The Steerable Pyramid Autoencoder based Framework .......... 45
    4.3.3 The Log-Gabor Autoencoder based Framework ..................... 57
  4.4 Summary ................................................................................. 62

Chapter 5: Conclusion ........................................................................ 64

Bibliography ..................................................................................... 66
List of Tables

Table 3.1  The contingency table .................................................. 36
Table 4.1  Evaluation of three autoencoders ................................. 44
Table 4.2  Parameter optimization of the hidden layers .................. 46
Table 4.3  Parameter optimization of the steerable orientations .......... 46
Table 4.4  The comparison of the accuracy and recall from five-fold cross-validation 48
Table 4.5  The comparison of the precision and F1-score from five-fold cross-validation .................................................. 49
Table 4.6  Evaluation of key-frame identification on all the experimental methods ................................. 51
Table 4.7  The evaluation metrics from five-fold cross-validation on SPAE and LGAE .................................................. 58
Table 4.8  Computational costs of three frameworks on a five-minute video .................................................. 61
Table 4.9  Evaluation of key-frame identification between SPAE and LGAE on two varied datasets .................................................. 62
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The free-swimming robotic platform (PipeDiver®).</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Some illustrations of the data obtained from CCTV inspection videos.</td>
<td>2</td>
</tr>
<tr>
<td>2.1</td>
<td>The schematic diagram of an autoencoder network.</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>The schematic structure of a convolutional neural network.</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>The illustration of video analysis using convolutional autoencoder.</td>
<td>12</td>
</tr>
<tr>
<td>3.1</td>
<td>The research overview of the water pipeline CCTV inspection.</td>
<td>15</td>
</tr>
<tr>
<td>3.2</td>
<td>The unsupervised learning-based framework with state-of-the-art autoencoders.</td>
<td>16</td>
</tr>
<tr>
<td>3.3</td>
<td>The schematic structure of an autoencoder.</td>
<td>16</td>
</tr>
<tr>
<td>3.4</td>
<td>The schematic structure of a convolutional autoencoder.</td>
<td>17</td>
</tr>
<tr>
<td>3.5</td>
<td>The schematic structure of a variational autoencoder.</td>
<td>18</td>
</tr>
<tr>
<td>3.6</td>
<td>The SPAE-based framework for key-frame identification.</td>
<td>19</td>
</tr>
<tr>
<td>3.7</td>
<td>The structure of a basic autoencoder network.</td>
<td>21</td>
</tr>
<tr>
<td>3.8</td>
<td>System diagram of the steerable pyramid, illustrating the filtering and sampling operations, and the recursive subsystem in the dotted box.</td>
<td>22</td>
</tr>
<tr>
<td>3.9</td>
<td>The steerable pyramid.</td>
<td>23</td>
</tr>
<tr>
<td>3.10</td>
<td>The overall structure of the steerable pyramid autoencoder based framework.</td>
<td>25</td>
</tr>
<tr>
<td>3.11</td>
<td>The building block illustrating the lateral connection for the fusion of steerable pyramid in the SPAE.</td>
<td>26</td>
</tr>
<tr>
<td>3.12</td>
<td>The schematic structure of a nonlinear support vector machine.</td>
<td>27</td>
</tr>
<tr>
<td>3.13</td>
<td>The LGAE-based framework for key-frame identification.</td>
<td>29</td>
</tr>
<tr>
<td>3.14</td>
<td>Sample video frame and the golden pyramid of log-Gabor filter on the sample frame: (a) sample video frame; (b) the golden pyramid of the log-Gabor filter.</td>
<td>32</td>
</tr>
<tr>
<td>3.15</td>
<td>The overall structure of the log-Gabor autoencoder based framework.</td>
<td>33</td>
</tr>
<tr>
<td>Figure 3.16</td>
<td>The model construction of the proposed LGAE.</td>
<td>34</td>
</tr>
<tr>
<td>Figure 3.17</td>
<td>The diagram for k-fold cross-validation.</td>
<td>37</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>The training curves for the loss of the experimental models.</td>
<td>41</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>The reconstruction loss of the testing samples.</td>
<td>42</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>Loss distribution of the autoencoder.</td>
<td>42</td>
</tr>
<tr>
<td>Figure 4.4</td>
<td>Loss distribution of the convolutional autoencoder.</td>
<td>43</td>
</tr>
<tr>
<td>Figure 4.5</td>
<td>Loss distribution of the variational autoencoder.</td>
<td>43</td>
</tr>
<tr>
<td>Figure 4.6</td>
<td>The bar chart of the accuracy in the cross-validation of SPAE-based framework.</td>
<td>47</td>
</tr>
<tr>
<td>Figure 4.7</td>
<td>The training history of SPAE.</td>
<td>47</td>
</tr>
<tr>
<td>Figure 4.8</td>
<td>The comparison of the extracted features between the selected experimental methods.</td>
<td>51</td>
</tr>
<tr>
<td>Figure 4.9</td>
<td>Illustration of the comparative results.</td>
<td>52</td>
</tr>
<tr>
<td>Figure 4.10</td>
<td>ROC curves of the deep learning methods in the study.</td>
<td>53</td>
</tr>
<tr>
<td>Figure 4.11</td>
<td>Precision-recall curves of the deep learning methods in the study.</td>
<td>54</td>
</tr>
<tr>
<td>Figure 4.12</td>
<td>3-dimensional t-SNE feature distribution based on: (a) SPAE; (b) AE; (c) CAE; (d) VAE.</td>
<td>55</td>
</tr>
<tr>
<td>Figure 4.13</td>
<td>Temporal regularity of the CCTV inspection video.</td>
<td>57</td>
</tr>
<tr>
<td>Figure 4.14</td>
<td>The comparison of the evaluation metrics between SPAE and LGAE.</td>
<td>58</td>
</tr>
<tr>
<td>Figure 4.15</td>
<td>The hidden representations from SPAE and LGAE compared with raw frames.</td>
<td>59</td>
</tr>
<tr>
<td>Figure 4.16</td>
<td>ROC curves of the deep learning methods in the study.</td>
<td>60</td>
</tr>
<tr>
<td>Figure 4.17</td>
<td>Precision-recall curves of the deep learning methods in the study.</td>
<td>61</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>CCTV</td>
<td>Closed-Circuit Television</td>
<td></td>
</tr>
<tr>
<td>DT</td>
<td>Destructive Testing</td>
<td></td>
</tr>
<tr>
<td>NDT</td>
<td>Nondestructive Testing</td>
<td></td>
</tr>
<tr>
<td>DL</td>
<td>Deep Learning</td>
<td></td>
</tr>
<tr>
<td>SED</td>
<td>Sobel Edge Detection</td>
<td></td>
</tr>
<tr>
<td>CED</td>
<td>Canny Edge Detection</td>
<td></td>
</tr>
<tr>
<td>HOG</td>
<td>Histogram of Oriented Gradients</td>
<td></td>
</tr>
<tr>
<td>R-CNN</td>
<td>Region based Convolutional Neural Networks</td>
<td></td>
</tr>
<tr>
<td>CNN</td>
<td>Convolutional Neural Network</td>
<td></td>
</tr>
<tr>
<td>AE</td>
<td>Autoencoder</td>
<td></td>
</tr>
<tr>
<td>CAE</td>
<td>Convolutional Autoencoder</td>
<td></td>
</tr>
<tr>
<td>VAE</td>
<td>Variational Autoencoder</td>
<td></td>
</tr>
<tr>
<td>SP</td>
<td>Steerable Pyramid</td>
<td></td>
</tr>
<tr>
<td>ML</td>
<td>Machine Learning</td>
<td></td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial Neural Network</td>
<td></td>
</tr>
<tr>
<td>NN</td>
<td>Neural Network</td>
<td></td>
</tr>
<tr>
<td>MLP</td>
<td>Multi-Layer Perception</td>
<td></td>
</tr>
<tr>
<td>NLP</td>
<td>Natural Language Processing</td>
<td></td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
<td></td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function</td>
<td></td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver Operating Characteristic</td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Squared Error</td>
<td></td>
</tr>
<tr>
<td>TP</td>
<td>True Positive</td>
<td></td>
</tr>
<tr>
<td>FP</td>
<td>False Positive</td>
<td></td>
</tr>
<tr>
<td>TN</td>
<td>True Negative</td>
<td></td>
</tr>
<tr>
<td>FN</td>
<td>False Negative</td>
<td></td>
</tr>
<tr>
<td>TPR</td>
<td>True Positive Rate</td>
<td></td>
</tr>
<tr>
<td>FPR</td>
<td>False Positive Rate</td>
<td></td>
</tr>
</tbody>
</table>
Acknowledgements

First of all, I would like to present immense gratitude to my supervisor, Professor Zheng Liu, whose guidance and encouragement throughout my graduate study have been indispensable. I have attained valuable research skills and technical writing skills from his guidance. Additionally, I would also like to thank him for helping with the opportunity to work with industries, which has brought me precious working experience. I have developed professional attitude and skills in my graduate study with the help of my supervisor.

I would also like to thank Dr. Kenneth Chau and Dr. Yang Cao for their willingness to serve on the advisory committee. I really appreciate their valuable time and constructive comments on my thesis.

In addition, I would like to give my appreciation to all of my colleagues at the Intelligent Sensing, Diagnostic and Prognostic Research Lab (ISDPRL), who have encouraged me and tried to offer help with my research. Especially, I would thank Miss. Rakiba Rayhana for her support and constant encouragement during the past two years.

Finally, I owe my parents the greatest gratitude for their understanding, mental support, patience and love over all the past years. I could not possibly achieve the research work without their continuous encouragement and support, especially their kindness and care to me. I wish the happiest and healthiest life to my parents.
Chapter 1

Introduction

1.1 Background and Motivation

Water pipeline plays a crucial role in human life for providing a stable water supply. However, various defects are observed in the water pipelines such as water leakage, blockage, air pocket, debris, fracture, and so on [1, 2]. As a critical urban infrastructure, any damage can lead to severe distress of both the water mains and water quality [3]. These disruptions and damages caused by pipeline events are costly to repair [4]. For instance, one large-diameter failure may cost approximately 1.5 million dollars. Hence, it is essential to perform in-time condition assessment and maintenance of the water pipelines to avoid any massive disaster. Condition assessment commonly represents the analysis of the condition of the infrastructure with regard to many aspects, e.g., materials, age, deficiencies, and construction methods [5–7]. Specifically, condition assessment plays a vital role in surveillance and security applications, which allows robots or human operators to respond in time and take actions as needed to ensure the regular function of the equipment or platforms. With condition assessment, essential information can be provided to evaluate the condition of an infrastructure [8]. To date, multiple technologies have been performed for condition assessment of water mains. Closed-circuit television (CCTV) is being widely adopted to provide visual inspection without man-entry in water pipeline condition assessment.

![Figure 1.1: The free-swimming robotic platform (PipeDiver®).](image)

In this study, PipeDiver® [9] was employed as a platform to collect the inspection data. PipeDiver® is a water pipeline condition assessment tool with free-swimming capacity in a
long distance, which can operate while the water transmission and distribution system remains in service. As a result, it provides utility owners with an easier and less costly alternative to inspection methods that require shutdown or dewatering of the system. This robotic platform is developed by Pure Technologies, whose snapshot is presented in Figure 1.1. As can be seen, it is featured with collapsible pedals, which help passing through sharp bends, diameter-reduction, tees, and butterfly valves in the water pipelines. Besides, the robotic platform is equipped with a lighting array and multiple cameras to conduct inspection through the pipelines. It has high flexibility due to its deployment of the appurtenances, which allow it to travel through varied configurations of the pipes during the inspection. The benefits of this tool include providing actionable pipe wall data, no human disruption during the inspection, and long inspection distances. Specifically, several cameras are attached to this tool from different angles to conduct CCTV inspection and collect video data. Some illustrations of the data obtained from the CCTV videos are presented in Fig. 1.2. The first row and second row present the video frames obtained from the inspection videos of two different pipelines. The data presented on the first row shows a worse illumination setting, which results in the more difficult visual observation of the defects in the water pipes. As circled on the key frames, there are certain longitudinal cracking, offtake, air pocket, and debris observed on the internal surface of the water pipes.

Figure 1.2: Some illustrations of the data obtained from CCTV inspection videos.

The inspection by PipeDiver® through traveling along the pipelines creates a large number of inspection videos. To date, companies still assign inspectors to check each video frame to identify anomalies by watching the whole recorded videos during the office-based survey, which is tedious, strenuous, inefficient, and error-prone. Sometimes, the manual inspection is not accurate due to the operators’ fatigue. Above all, it is said that the inspectors can spend eight times as long as the video length to finish the manual labeling work, which means it is hard to realize water pipeline maintenance in time. Hence, it is necessary and critical to
perform automated key-frame identification to first identify the key frames of the videos for condition assessment instead of spending hours observing the videos. An automated process can release the inspectors’ workload and ensure the consistent quality of the survey. However, a fully automated survey of varied structural discontinuities still remains as a challenge to increase the accuracy and efficiency of water pipeline inspection on CCTV videos. Advanced data analytic capabilities are needed for inspection and interpretation of the system operation to empower the PipeDiver® platform.

1.2 Literature Review

Based on the background of the study, water pipeline condition assessment and the existing techniques are reviewed. Primarily, CCTV-based inspection of the water pipe is investigated in this chapter. Besides, CCTV video analytics methodologies are introduced with potential solutions to enhance the water pipeline inspection.

1.2.1 CCTV-based Condition Assessment of Water Pipeline

Condition assessment is necessary for water distribution and water transmission. Specifically, it is essential in surveillance and security applications to ensure the inspection and interpretation of the system operation. This enables human inspectors to respond and take actions in time to keep the consistent function of the equipment or platforms. Multiple technologies have been carried out for condition assessment of the water pipeline. To date, destructive testing (DT) and non-destructive testing (NDT) are two types of techniques conducted in the water pipeline inspection [10]. Destructive testing entails removal of a sample from a pipe to perform analysis. However, NDT offers a safe and reliable way of inspection, which is cost-effective and requires little disruption to the water system. NDT covers direct visual observation of defects as well as techniques that provide signals or signatures interpreted into distress indicators [10]. For example, the pressure in different nodes and the flow in different pipes have been measured to estimate leakage and its severity in the water distribution system[1]. Pit depth measurement is one of the NDT techniques carried out when the pipe is exposed, which is inconvenient to some extent. Other technologies include ultrasonic testing, radiographic testing, and thermographic testing [10, 11]. In addition, advances in sensor technologies give rise to the growing use of sensor networks in pipe inspection. Till-date, studies implemented different analysis and prediction techniques based on sensory data. Conditional monitoring across different fields has been widely investigated through multi-sensor data analysis [12]. For instance, wireless sensor network is applied for condition assessment of water-flood [13]. Leakage or
1.2. Literature Review

Blockage detection was performed through wireless sensor networks for water distribution system [14, 15]. Computational intelligent methods have also been applied to process sensory data for water transmission and distribution system [16]. Furthermore, visual inspection, electromagnetic inspection, and acoustic inspection are three other NDT techniques for water pipeline condition assessment [10].

As for visual inspection, the condition of the internal surfaces of the pipe can be assessed by a variety of vision aids. CCTV is one of the widely-adopted equipment to visualize the internal conditions of the water or sewer pipelines recently [10, 17]. CCTV inspection provides a direct visual representation of defects in the pipe, which highly facilitates the inspectors to understand and assess the pipe condition accurately [8]. This shows an advantage over other techniques; hence it is gaining attention from researchers. To date, the majority of the studies for CCTV inspection are based on the sewer pipeline. As a daily infrastructure, sewer pipe commonly has a low level of sewer water, which means the pipe is not full of sewer water. On the contrary, the water pipeline is usually in the condition of water transmission, namely, full of the water being transmitted. As introduced above, the video data acquired from the water pipeline is underwater data in this study, since the PipeDiver® for data collection does not require dewatering of the system in service. However, the data for sewer and water pipeline have similarities on the in-pipe condition to some extent. Hence, related techniques for the sewer pipeline are also reviewed and referred to in the study. On the basis of CCTV inspection videos, studies implemented image processing methods on sewer pipeline, such as Sobel edge detection (SED), canny edge detection (CED) [18] and histogram of oriented gradients (HOG) [19]. Moreover, a recent study by Yin et al. shows a deep learning-based framework for inspection on sewer pipes by using faster region based convolutional neural networks (R-CNN) [20]. However, few studies conduct water pipeline inspection using state-of-the-art deep learning (DL) methods, especially on CCTV videos. The objective of the thesis is to conduct key-frame identification, which is to detect key frames containing pipe anomalies (also known as anomaly frames) in the inspection videos. For instance, a study presents a key-frame extraction method for greenhouse monitoring video [21].

1.2.2 CCTV Video Analytics Methodologies

The last few years have seen an increased interest in CCTV video analytics, which has received a lot of research attention. CCTV video analytics can be addressed through image processing methodologies since each video frame can be regarded as an image. Conventional techniques include multiple detection methodologies such as Sobel edge detection (SED) and canny edge detection (CED), which have been implemented on sewer pipe video data [18]. In
the past few years, deep learning techniques have risen up, especially in the domain of image and video analytics. Deep learning methods have been well-adopted for various automated tasks, which were proved to be efficient and accurate. For instance, CNN is a well-known supervised deep learning model for image classification and detection, which can be implemented on video frames [22, 23]. R-CNN is a type of deep learning model derived from CNN, which is normally applied for object detection on video data [24]. Faster R-CNN is then an improved version of R-CNN [20, 25]. Moreover, autoencoder (AE) is one of the state-of-the-art unsupervised learning methods in the pattern recognition domain. The encoder structure of an autoencoder is widely applied for image representation learning and extraction, which can then be used to various prediction algorithms. Autoencoder was first proposed in the paper by Kramer [22]. Other autoencoder-based models include convolutional autoencoder (CAE) [26, 27], variational autoencoder (VAE) [28, 29], multimodal autoencoder [30], adversarial autoencoder [31], deep laplacian autoencoder [32] and so on. Another deep learning method called appearance and motion DeepNet was proposed for feature detection on surveillance videos in 2015 [33]. Besides, Chong and Tay carried out a spatio-temporal autoencoder for event detection of videos [34]. These methods were proved to have good performance, especially for anomaly detection on surveillance videos. However, few researchers have implemented these methods for water pipeline CCTV inspection so far.

### 1.3 Research Challenges and Objectives

From the perspective of water pipeline inspection, a fully automated office-based survey of varied structural deficiencies still remains as a challenge. Also, the majority of the studies are based on sensor technologies for the condition assessment of the water pipes. Few researches have carried out CCTV-based water pipeline inspection with a direct visual representation of defects in the pipe. For the office-based survey, it is important to first identify the probable key frames of the CCTV videos, which contain the major anomalies captured from the internal surface of the pipe. Thus, the inspector can focus more on these key frames to perform multiple tasks. This will significantly facilitate the office-based survey to help save time and labor.

On the other hand, few researchers have implemented deep learning methods for water pipeline CCTV inspection till date. Hard mining on the underwater video data needs to be empowered through advanced deep learning techniques. The methodologies for performing the key-frame identification can be further developed to adapt to the water pipeline dataset and achieve better performance in this study. To be more specific, accurate feature detection, namely, representation learning, is challenging, which has been extensively researched. The input image must be transformed into a feature representation, which is easier for an algorithm
to perform classification or detection tasks. Hence, it is crucial to develop a better algorithm to obtain outstanding capacity of feature extraction, especially on the specific in-pipe video data in the underwater condition in this study. As a result, a good set of representative features can be produced, which makes it easier for the classifier to perform the detection task on water pipeline video frames in the thesis.

Therefore, the objectives of the thesis include the application of a methodology to address key-frame identification for water pipeline CCTV inspection. This is to identify the key frames of the CCTV video, which contain the major anomalies captured from the internal surface of the water pipe. In addition, the methodology needs to be enhanced to adapt to the specific in-pipe data in the underwater condition. The hard mining capability of the methodology is to be developed to deeply exploit information on the video data.

1.4 Thesis Outline and Contributions

The thesis is organized into five chapters.

Chapter 1 presents the background of water pipeline condition assessment and the current solutions to enhance the inspection. In addition, this chapter provides a literature review for the water pipeline condition assessment and CCTV video analytics. Furthermore, the research challenges and objectives are also stated in this chapter.

Chapter 2 investigates the state-of-the-art autoencoder-based techniques relevant to the thesis. First, an overview of machine learning is presented. Secondly, the development history of autoencoder-based techniques is introduced. Finally, the state-of-the-art autoencoder-based technique, namely, convolutional autoencoder, is investigated, which is one of the most essential techniques in this thesis.

Chapter 3 introduces three autoencoder-based frameworks and their implementations on the task of key-frame identification. First, an unsupervised learning-based framework is carried out with state-of-the-art autoencoders. Secondly, a steerable pyramid autoencoder (SPAE) based framework is proposed for the purpose of enhancing accuracy. Thirdly, a log-Gabor autoencoder (LGAE) based framework is proposed, which is a further improved version of the SPAE-based framework.

In Chapter 4, the experimental results and discussions are demonstrated regarding three autoencoder-based frameworks conducted on key-frame identification. First, the unsupervised learning-based framework achieves up to 0.940 accuracy on the task with a lower precision metric of 0.901. Secondly, the SPAE-based framework shows improved performance with an averaged accuracy of 0.984 on the water pipeline video data in specific underwater condition. To the authors’ best knowledge, this is the first study to integrate steerable pyramid
with an autoencoder-based network, resulting in outperforming capability of representation learning and extraction. Finally, the LGAE-based framework achieves the accuracy of 0.988 and the recall metric of 0.996, which outperforms the other selected experimental methods. Explicitly, the superiority of LGAE comes from its compelling performance on feature generation and generalization capacity. Hence, LGAE-based framework will significantly assist the office-based survey on water pipeline CCTV inspection. The inspectors can focus on the most important information and conduct pipe assessment as well as maintenance effectively to avoid severe mistakes.

Chapter 5 concludes the thesis. Future work is also suggested.
Chapter 2

Autoencoder Techniques: The State of the Art

Autoencoder-based networks have found their applications on many complicated problems, especially for image or video analysis. They are proved to have good performance on feature detection, image classification, and anomaly detection on surveillance videos [33, 34]. Hence, autoencoder techniques are selected as the research methodologies to be further developed and implemented on the key-frame identification. The autoencoder techniques are investigated in this chapter in terms of their development history and fundamentals. Additionally, one subset of autoencoder networks, i.e., convolutional autoencoder network, is investigated since it is the basic approach based on which the proposed methodology is developed.

Machine learning (ML) technology evolves rapidly and powers many applications in life. For decades, it has been widely applied in a variety of domains such as computer vision, marketing, economics, and so on. Machine learning is a field of study on computer science, which is to apply computer algorithms on the data to discover patterns of interest. It is seen as a subset of artificial intelligence. Specifically, ML algorithms build a mathematical model based on a set of data in order to make predictions or decisions. It is achieved without the model being explicitly programmed to do so. The data is also known as “training data”. The model improves automatically through experience on the training data. Artificial neural network (ANN), which is commonly called neural network (NN), is a group of computing systems inspired by the biological neural networks that constitute animal brains [35]. It is extensively adopted in machine learning, which models the data through graphs of artificial neurons. Each neuron in an ANN works approximately in the way how a neuron works in a biological brain. To be more specific, each neuron can receive, process and transmit information in the ANN. The output of a neuron is computed by a specified function on the sum of its inputs. Explicitly, an ANN contains multiple layers with a certain number of neurons in each layer. Different layers could perform varied transformations on their inputs. Supervised learning, unsupervised learning, and reinforcement learning are three major types of learning paradigms of ANN. Convolutional neural network (CNN) is a typical supervised learning model, while autoencoder is a typical unsupervised learning model. An autoencoder is a type of artificial neural network, which is performed
2.1. Autoencoder Network

An autoencoder is a type of artificial neural network, which is performed to learn efficient data codings in an unsupervised manner [22]. The aim of an autoencoder is to learn a representation for a set of data for the purpose of performing specific tasks. An autoencoder consists of an encoder and a decoder, where the encoder generates a reduced encoding, and the decoder reconstructs an output from the encoding as close as possible to the raw input. The schematic diagram of an autoencoder is illustrated in Fig. 2.1. More specifically, an autoencoder is trained to encode the input data into a certain representation in the encoder, which is then decoded back to the original format of the input data. Essentially, the autoencoder attempts to optimize its hyper-parameters to improve the reconstruction outcome in the training process. Thus, the autoencoder is able to learn and extract useful representations of the input continuously, as well as tries to discard the useless information during the propagation. It is the major strength of an autoencoder network. In addition, the efficiency of the training procedure for an autoencoder can be relatively higher since a high dimensional input vector can be converted into a lower-dimensional representation in the encoder.

![Figure 2.1: The schematic diagram of an autoencoder network.](image)

During the past few decades, the autoencoder has been rapidly developed among researches regarding the ANN. In 1988, it is found out by Bourlard and Kamp [36] that a multi-layer perception (MLP) in auto-association mode could achieve dimensionality reduction and data
2.2 Convolutional Autoencoder Network

Deep learning is a subset of machine learning based on artificial neural network with representation learning. For decades, both shallow and deep learning of artificial neural networks have been researched in many fields [45, 46]. More precisely, deep learning algorithms apply multiple layers to learn and extract useful features progressively from the raw input [47]. For instance, the early layers can learn and extract lower semantic features from the input, while the deeper layers develop complex features of the input. The word “deep” indicates the large number of layers used in the deep learning model to conduct the task. An autoencoder can also be deeper with more hidden layers in the feature learning stage of the model. Deep autoencoder networks show higher potential for achieving good performance in multiple tasks. With the development of deep learning technique, convolutional neural network evolves rapidly in different domains. Convolutional neural network is a subset of deep neural networks in deep learning, which is most commonly conducted to explore pattern information. CNN is preferably adopted in the domains such as image processing, image classification, natural language processing (NLP), and so on. Meanwhile, the layered structure of CNN is extensively combined with other structures to form new models with the aim of performing different tasks. The schematic structure of a convolutional neural network is displayed in Fig. 2.2. A convolutional neural network is composed of an input layer, multiple hidden layers, as well as an output layer. As shown in Fig. 2.2, the hidden layers in a CNN typically consist of convolutional layers, subsampling layers, and fully-connected layers. Feature maps can be obtained after each hidden layer, which is presented in the schematic structure.

Specifically, convolutional layers are regularized versions of multi-layer perceptions, which are also known as fully-connected layers. For fully connected networks, each neuron in one
Figure 2.2: The schematic structure of a convolutional neural network.
Figure 2.3: The illustration of video analysis using convolutional autoencoder.
2.3. Summary

Specific layer is connected to all the neurons in the next layer. It means that these networks are prone to have overfitting problem on certain datasets. For the purpose of regularization, convolutional neural network takes advantage of the hierarchical patterns of the data to assemble complex patterns. Thus, CNN is less prone to be extremely constrained on the data. Due to the superiority of convolutional structure, it is widely adopted in other networks to achieve better performance on certain tasks. Convolutional autoencoder was proposed then, which is to implement convolutional layers in the form of an autoencoder. CAE takes advantage of the strength of convolutional layers, thus showing potential in multiple tasks. Also, a convolutional autoencoder can have multiple hidden layers with a deep structure as needed for performing a specific task on the dataset.

To be more precise, the illustration of video analysis using convolutional autoencoder is presented in the Fig. 2.3. As illustrated in Fig. 2.3, a convolutional autoencoder is composed of convolutional (Conv) layers, pooling layers, and upsampling layers. Each convolutional layer is followed by a pooling layer in the encoding part of the CAE. On the other side, each convolutional layer is constructed with an upsampling layer in the decoder. However, the pooling layer can also be replaced by the stride function of a convolutional layer. For instance, if the stride parameter of a convolutional layer is defined as 2, the features output from this layer will be downsampled by a factor of 2. Explicitly, the stride in the convolutional layer is able to change the types of features extracted from the layer, whereas the pooling layer is simply changing how much the data is downsampled. Above all, the convolutional autoencoder network has advantages for image or video analysis because it is able to learn the pattern of data accurately through the encoder-decoder structure. Explicitly, the model can extract representative features from the images, which can then be applied to an algorithm for prediction. The representative features detected are compressed and more separable, which will benefit varied application tasks.

2.3 Summary

In this chapter, the basic principles and the state-of-the-art techniques regarding the autoencoder network are introduced. Due to the superiority of the autoencoder-based network in feature learning and extraction for image or video analysis, it is adopted in this research to enhance the key-frame identification of water pipeline CCTV videos. Since the research majorly involves the development based on state-of-the-art CAE, the origins and principles of CAE are explicitly introduced in this chapter.
Chapter 3

Autoencoder-based Key-Frame Identification Frameworks and Implementations

3.1 Introduction

As introduced in the Section 1, the aim of the study is to perform key-frame identification on the water pipeline CCTV inspection data. Fig. 3.1 displays the research overview of the water pipeline CCTV inspection. First, CCTV data are acquired from the free-swimming robotic platform, i.e., PipeDiver® through travelling along the pipelines. The office-based survey is then conducted by the inspectors with the aim of performing pipe condition assessment, as demonstrated in Fig. 3.1. In the office-based survey, key-frame identification is performed with an autoencoder-based framework, after which the key frames with pipe anomalies can be collected. Then, the inspectors can focus more on these key frames to conduct the assessment report for further pipe assessment and maintenance by the utility manager. Three frameworks are introduced and implemented in the thesis. To be more specific, an unsupervised learning-based framework is first conducted to perform the task, which is built based on state-of-the-art autoencoder networks. Secondly, SPAE-based framework is carried out with the proposed autoencoder-based network, which is incorporated with steerable pyramid method. Precisely, this framework is developed with the combination of an unsupervised deep learning model and a supervised classifier to improve the performance. Furthermore, another framework, namely, LGAE-based framework, is then performed to improve the accuracy of key-frame identification on the dataset. This framework has a similar schematic structure to that of the SPAE-based framework, but with an integration of log-Gabor filter to enhance the capability of identifying key frames. Three frameworks and related evaluation metrics for the frameworks are introduced in detail in this chapter.
3.2 The Unsupervised Learning-based Framework

As stated above, key-frame identification is to detect the key frames containing pipe anomalies (also known as anomaly frames) in the CCTV inspection videos. The unsupervised learning-based framework is first performed for this task. It is composed of a state-of-the-art autoencoder and a threshold determinator for identifying the key frames out of the video frames.

3.2.1 Framework Structure

In this section, the structure of the unsupervised learning-based framework is shown in Fig. 3.2. Specifically, the framework is formed by an autoencoder-based network and a threshold determinator. Normal video frames are used for training the model and obtaining the threshold. The top of the flow diagram shows the training procedure, and the bottom of that presents the testing process. The pre-trained autoencoder model and the pre-defined threshold are obtained from the training process and applied to identify the key frames of the CCTV inspection videos. To be more explicit, the threshold is determined on the training result of the autoencoder-based network. After the training process of the network, the loss values of the normal frames from the network will fall onto a certain level, which is relatively lower since the network has learned the patterns of the normal frames accurately. However, the loss values of the key frames from the network will fall onto a much higher level because the network has not learned the patterns of the key frames in the training process. A threshold can be obtained from the averaged estimation for the loss values of normal frames. As a result, the threshold can be applied to detect the key frames out of the video frames.

3.2.2 Computational Methodologies

Unsupervised learning is conducted by using three autoencoder-based models, which are autoencoder, variational autoencoder, and convolutional autoencoder. Autoencoder-based mod-
3.2. The Unsupervised Learning-based Framework

Figure 3.2: The unsupervised learning-based framework with state-of-the-art autoencoders.

Autoencoders are capable of reconstructing images. Reconstruction loss can be comprehended as the difference between the input and output data, which is also known as reconstruction error. In the proposed framework, the model is only trained on normal frames of the CCTV videos to achieve a minimum reconstruction error. Hence, the model can reconstruct normal video frames accurately but not the frames with pipe anomalies. If a threshold is appropriately defined, key frames can be detected effectively based on the reconstruction loss.

Autoencoder

An autoencoder is a type of neural network, which is performed to learn efficient data codings in an unsupervised manner [22]. The aim of an autoencoder is to learn a representation (encoding) for a set of raw data. Autoencoder consists of an input layer, an encoder network, and a decoder network [48]. Specifically, it contains an input layer, multiple hidden layers, and an output layer. The raw data are fed into the model through the input layer. The encoder layer is for encoding the input data, learning the representative features. While the decoder layer is the reverse of the encoder layer, using the features to reconstruct the raw data. A basic autoencoder is formed by fully-connected layers in the model. Fig. 3.3 shows the schematic diagram of a basic autoencoder network, where the gray hidden layers represent the fully-connected layers.

Figure 3.3: The schematic structure of an autoencoder.
3.2. The Unsupervised Learning-based Framework

Assuming the autoencoder has $L$ layers, the input data is defined as $N$-dimensional vector $x$ and the corresponding values of the output layer is $x^L$. Based on the parameter optimization, an autoencoder network can learn representative features in the latent space and reconstruct the original data. The output data of an autoencoder has the same dimensions as that of the input data, which is fed into the network.

The loss function of an autoencoder can be described as:

$$
\text{loss} = \frac{1}{K} \sum_{k=1}^{K} (x^L_k - x_k)^2 
$$

(3.1)

where $K$ is the number of training samples. $x_k$ is the $k$th training sample, and $x^L_k$ is the output value of the $k$th training sample. $\text{loss}$ is the reconstruction loss of the autoencoder network (also known as mean squared error).

**Convolutional Autoencoder**

Convolutional autoencoder is developed from AE by adding convolutional neural network (CNN) structures [26] [27]. CAE takes advantage of CNN because it applies convolutional structure in the form of an autoencoder. Commonly, a convolutional layer is applied to extract the features from the input data, followed by a pooling layer to compress the features of the input. The representation in the latent space is then converted back to the original dimensions in reverse through convolutional and upsampling layers. Fig. 3.4 indicates the schematic structure of a convolutional autoencoder. To be more explicit, the first gray block of Fig. 3.4 represents the hidden convolutional layers with pooling layers for feature compression. In contrast, the second gray block shows the hidden convolutional layers and upsampling layers in the decoder part of the model. Obviously, the orange block demonstrates the representation extracted through the encoding process in the latent space.

![Figure 3.4: The schematic structure of a convolutional autoencoder.](image)
3.2. The Unsupervised Learning-based Framework

Variational Autoencoder

Variational Autoencoder (VAE) is a popular generative model, which allows representing data in a compact probabilistic latent space [49, 50]. Specifically, it produces an encoding of the raw input $x$ to a latent vector $v = E(x) \sim e(v|x)$, after which the latent vector is converted back to the origin dimensions of the raw input, i.e., $x' = D(v) \sim d(x|v)$. The $x$ and $x'$ represent the input data and output data of the VAE. The marginal log-likelihood of each pixel in the input sample is maximized for the purpose of image reconstruction. Explicitly, the reconstruction loss of the VAE $l_r$ is expected to be the negative log-likelihood of pixels in the raw input. Besides, VAE has a key component, that is, the capability of controlling the distribution of the latent vector. This is conducted through Gaussian random variable, i.e., $v \sim N(0, I)$. Furthermore, the distribution of a Gaussian distribution is also named KL divergence. The difference between the distribution of $e(v|x)$ and the KL divergence can be quantified and minimized by gradient descent algorithm [49]. Hence, the training process of VAE involves the optimization of the $l_r$ and KL divergence loss $l_{kl}$. Fig. 3.5 illustrates the schematic structure of a variational autoencoder. Specifically, the gray blocks of Fig. 3.5 demonstrate the hidden fully-connected layers in the encoder and decoder of the model. Moreover, the green block shows the distribution of the latent vector through the controlling of the encoding layers. Related equations are also displayed on the Fig. 3.5.

$$l_r = -E_{e(v|x)}[log(d(x|v))]$$  \hspace{1cm} (3.2)

$$l_{kl} = D_{kl}(e(v|x)||d(v))$$  \hspace{1cm} (3.3)

$$l_{vae} = l_r + l_{kl}$$  \hspace{1cm} (3.4)

To sum up, an unsupervised learning-based framework is proposed to conduct key-frame
identification for water pipeline CCTV inspection. Specifically, three state-of-the-art autoencoder networks are performed in this framework to achieve the objective.

### 3.3 The Steerable Pyramid Autoencoder based Framework

The steerable pyramid autoencoder (SPAE) based framework is on the basis of representation learning and extraction. It is an integration of unsupervised learning method and supervised learning method. Specifically, this framework consists of a feature extractor for extracting representative features and a classifier for performing the classification task. The feature extractor SPAE is trained in an unsupervised manner, while the classifier support vector machine (SVM) is trained in a supervised learning manner with labels.

#### 3.3.1 Framework Structure

As introduced, key-frame identification is to detect frames containing pipe anomalies in the CCTV inspection videos. The proposed key-frame identification framework is formed by steerable pyramid autoencoder and support vector machine [51]. Steerable pyramid autoencoder is performed for learning as well as extracting representative features and support vector machine is used for classification. The structure of the framework for key-frame identification is shown in Fig. 3.6. Specifically, the top of the flow diagram shows the training procedure and the bottom of that presents the testing process. First, the video frames pre-extracted from the CCTV inspection videos are pre-processed to train the feature extractor and classifier. The pre-trained feature extractor and classifier are then obtained and applied to conduct the detection task, as shown at the bottom of Fig. 3.6. In this process, the videos are pre-processed and input into the model frame by frame, after which the frames predicted as key frames are collected.

![Figure 3.6: The SPAE-based framework for key-frame identification.](image)

In deep learning, feature learning, also known as representation learning is a variety of techniques that enables a system to automatically exploit the representations needed for feature
classification or detection from raw data. Autoencoder is one of the state-of-the-art methods for this work. Significantly, this allows a machine to both learn the features automatically and make use of them to perform a specific task. However, autoencoder is learning the features layer by layer in depth, not in multi-scale or multi-orientation. Multi-scale representation learning has drawn attention of researchers due to the different structures of objects at different scales. Image pyramid is an implementation of learning image features at scales with sub-sampling operations. Hence, it is considerable to integrate the image pyramid features with the automatically extracted features of a neural network to improve the performance of feature learning. In order to improve the representation learning of the autoencoder network, steerable pyramid is integrated into the encoder of the model to perform multi-orientation feature learning. In comparison with conventional image pyramids and feature pyramid networks, steerable pyramid is not only scale-selective, but also orientation-selective, which allows higher flexibility and deeper feature extraction for image analysis. In addition, steerable pyramid is also superior in edge feature enhancement of the objects on images thanks to its recursive transformation. Hence, we developed a steerable pyramid autoencoder, which is to fuse an autoencoder network with steerable pyramid, seeking to enhance representation learning and extraction. The SPAE takes advantage of both pre-defined and automatically extracted features, which is the strength of SPAE.

3.3.2 Computational Methodologies

The SPAE-based framework is developed on the basis of the steerable pyramid technique and a customized autoencoder-based network. The computational methodologies are introduced in succession below.

Autoencoder Network

As introduced in Section 3.2.2, an autoencoder is composed of an input layer, multiple hidden layers, and an output layer. The schematic structure of an autoencoder network is displayed in Fig. 3.7 with the aim of showing the training process of the model. The outputs of the layers denoted with the same color in Fig. 3.7 have the same dimensions of data. Precisely, each circle represents a node (also known as unit) in the specific layer, where the number of units is pre-defined. Each layer in the neural network has its own hyper-parameters, which will be optimized and updated in the training procedure of the autoencoder. The detailed interpretation of the model training is stated in the following paragraph.

Assuming the autoencoder has \( L \) layers, where \( l \) is defined as the \( l \)th hidden layer. The weights between the \((l - 1)\)th and \(l\)th layer of the network are defined as the matrix \( W^l \). More-
3.3. The Steerable Pyramid Autoencoder based Framework

![Figure 3.7: The structure of a basic autoencoder network.](image)

over, \( b^l \) is the matrix of biases in the \( l \)th layer, and \( \phi^l \) is the activation function in the \( l \)th layer. Activation functions are used to perform element-wise transformation and enable autoencoder to learn nonlinear reduction of the feature dimensions. Assuming input data as \( N \)-dimensional vector \( x \), the values of the \( l \)th hidden layer \( x^l (l = 2, 3, ..., L - 1) \) and output layer \( x^L \) are calculated using the following recursive formula [30]:

\[
x^l = \phi^l (W^l x^{(l-1)} + b^l), \quad l = 2, 3, ..., L
\]  

(3.5)

Based on the recursive formula, an autoencoder network can learn representative features in the latent space and reconstruct the original data. The output data of an autoencoder has the same dimensions as that of the input data, which is fed into the network.

Steerable Pyramid

A steerable pyramid (SP), proposed by Simoncelli et al., is an implementation of a multi-orientation and multi-scale bandpass filter bank used for applications, including image compression, object recognition, texture synthesis, as well as image registration and fusion [52]. It can be considered as an orientation-selective method, in which a set of steerable filters are used at each level of the pyramid instead of a single filter [53, 54].

On the basis, the steerable pyramid has a set of directional derivative operators with the number of \( n \), leading to \( n + 1 \) orientations. As for the derivatives, they span a rotation-invariant subspace, in which the derivatives are designed and sampled. Fig. 3.8 shows the system diagram of a steerable pyramid [52]. It illustrates the filtering and sampling operations, as well as the recursive subsystem in the dotted box. The frequency axis is defined as \( \theta \) ranging from \( -\pi \) to \( \pi \) in the spectral decomposition of steerable pyramid. The symbols, i.e., “2D” and “2U,” correspond to downsampling and upsampling by a factor of 2, respectively. The circles refer to the transform coefficients at the specific level [52]. The raw image is represented with highpass and lowpass subbands, using highpass filters \( H_0 \) and lowpass filter \( L_0 \), respectively.
3.3. The Steerable Pyramid Autoencoder based Framework

Furthermore, the lowpass subband is then divided into a group of oriented bandpass subbands and a lowpass subband. This lowpass subband is subsampled by a factor of 2 in two directions, namely, X and Y [55]. In the system diagram of the steerable pyramid illustrated in Fig. 3.8, the radial function is represented as $B_n(\theta)$, which is constrained by decomposition and aliasing reduction during the subsampling procedure. The filters indicated as $H_0(\theta)$ and $L_0(\theta)$ are implemented for the preparation of the recursion. The raw signal is divided by the subsystem of steerable pyramid into two parts, which are highpass signal and lowpass signal. Next, the lowpass signal is subsampled, and the recursion is performed by repeatedly applying the recursive transformation to the lowpass signal [55]. The initial shape of lowpass is formed through $L_0(\theta) = L_1(\theta/2)$, same as that used in the recursive subsystem. Here, $L_1(\theta)$ is strictly bandlimited, and $B_n(\theta)$ is power-complementary.

![System diagram of the steerable pyramid](image)

Figure 3.8: System diagram of the steerable pyramid, illustrating the filtering and sampling operations, and the recursive subsystem in the dotted box. “2D” and “2U” correspond to downsampling and upsampling by a factor of 2. The circles correspond to the transform coefficients.

The constraints performed on the filters in the system diagram are as follows:

**Bandlimiting:**

$$L_1(\theta) = 0 \quad \text{for } |\theta| > \pi/2$$  \hspace{1cm} (3.6)

**Flat system response:**

$$|B(\theta)|^2 = |B_0(\theta)|^2 + |B_1(\theta)|^2 + \ldots + |B_n(\theta)|^2$$  \hspace{1cm} (3.7)

$$|H_0(\theta)|^2 + |L_0(\theta)|^2 [|L_1(\theta)|^2 + |B(\theta)|^2] = 1$$  \hspace{1cm} (3.8)

**Recursion:**

$$|L_1(\theta/2)|^2 = |L_1(\theta/2)|^2 [|L_1(\theta)|^2 + |B(\theta)|^2]$$  \hspace{1cm} (3.9)

As introduced above, the steerable pyramid method has the capacity of extracting repre-
sentative features of an image, emphasizing the region of interest on an image from different orientations in multiple scales. Fig. 3.9 illustrates that the features of a sample video frame are extracted into pyramid scale with four orientations horizontally and four levels vertically. To be more precise, the feature maps for the first level, namely, level 0 have the same dimensions as the original input. With the increase of the depth of the pyramid, the dimensions of the feature maps are downsized by a factor of 2 with more discriminative features extracted from the raw input. Moreover, each small pyramid in Fig. 3.9 indicates the feature extraction in depth from one specific orientation on a sample.

![Figure 3.9: The steerable pyramid.](image)

**Steerable Pyramid Autoencoder**

The steerable pyramid autoencoder (SPAE) is developed based on the incorporation of steerable pyramid into a customized CAE-based network. In the steerable pyramid autoencoder, mean squared error (MSE) is applied for training the model. To be more specific, MSE is the average squared difference between the predicted value and the actual value. Symbol $L$ represents the output layer of the model. Assuming input data as $N$-dimensional vector $x$, $x_i$ and $x^L_i$ are the $i$th dimensional values of the input $x$ and output $x^L$ of the model ($i = 1, 2, ..., N$). MSE($x$) is the mean square error (also known as reconstruction loss) of the input sample $x$. Accordingly, the function of MSE can be described as:

$$\text{MSE}(x) = \frac{1}{N} \sum_{i=1}^{N} (x^L_i - x_i)^2$$

(3.10)

The training procedure of the autoencoder network involves parameter optimization of $W^l$ and $b^l$ for $l = 2, ..., L$, which are stated in the subsection 3.3.2. Assuming the training dataset as $x_1, ..., x_k$ ($k = 1, 2, ..., K$), the optimization procedure is to minimize the mean of the MSE score with this training set by looking for the suitable parameters:
min \frac{1}{L} \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{N} (x_{k,i}^l - x_{k,i})^2, \quad l = 2, \ldots, L \quad (3.11)

where $x_{k,i}^l$ and $x_{k,i}$ are the $i$th dimensional value of $x_{k}^l$ and $x_{k}$. The algorithm implemented for solving the (3.11) problem is adaptive moment estimation (also known as adam) [56].

The proposed steerable pyramid autoencoder is developed based on convolutional autoencoder network, where the encoding structure is used as feature extractor. The encoder of the model applies convolutional (Conv) layers to extract the image features, and the decoder applies convolutional layers with upsampling layers to reconstruct the original image. Fig. 3.10 shows the overall structure of the proposed framework. In Fig. 3.10, the grey layers represent convolutional layers in the SPAE model. The strides parameter is defined as 2 in the convolutional layer of the encoder to downsize the feature maps, while the size parameter is set as 2 in the upsampling layer (represented by the yellow layer in Fig. 3.10) to upsample the feature maps. The orientation is set to be 4 in each level of the steerable pyramid displayed in Fig. 3.10. Hence, four feature maps can be obtained at each level of the pyramid. On the basis of the availability of the extracted pyramid features, the third level of the steerable pyramid is selected to be incorporated into the autoencoder-based network. Specifically, this is to avoid the insufficient or excessive effect of the recursive transformation of the steerable pyramid. From the trial and error study, we found that the first two levels of the pyramid did not achieve useful feature extraction for the dataset, while the fourth level achieved excessive features with certain aliasing. Therefore, the third level of the pyramid is selected to integrate with the proposed network.

The diagram in Fig. 3.10 presents an example of the integration of four oriented steerable inputs with the proposed network. The fusion of steerable pyramid can be considered as a lateral connection, which originates from the input layers of multiple steerable features from varied orientations. As demonstrated, the layer with four purple nodes indicates the feature fusion layer of steerable features, which is orientation selective. One input layer with one convolutional layer (represented by each purple node) is used for each oriented steerable feature, followed by an addition layer to fuse them all. The input steerable features are then combined with encoded features with the same dimensions from SPAE through an addition layer. Thus, the dimensions of the selected intermediate layer for merging must match those of the steerable inputs.

Fig. 3.11 presents a detailed diagram of the model construction from the input layer to the layer used for representation extraction. The building block illustrates the lateral connection for the fusion of steerable pyramid in the SPAE. The input is a $1 \times 512 \times 512$ matrix with the pixel information from the video frame. And the input for the lateral connection is a $1 \times 128 \times 128$
Figure 3.10: The overall structure of the steerable pyramid autoencoder based framework.
3.3. The Steerable Pyramid Autoencoder based Framework

matrix. Furthermore, a small ResNet module is also adopted into the autoencoder-based network, which is within the backbone of the network. It consists of two Conv layers, one normalization layer between two Conv layers, and at last a skip connection to the first Conv layer in the module [57]. Precisely, the ResNet module is directly connected between two layers in the proposed network, without performing dimensionality reduction. It is represented as the green block in the model structure of Fig. 3.10. As is known, ResNet has several compelling advantages such as alleviating the vanishing-gradient problem, powerful representational ability, and strengthening feature propagation with a ”shortcut connection” [58]. Many computer vision applications have been boosted by taking advantage of the powerful representational ability of ResNet, such as object detection and face recognition. As for normalization, group normalization (GN) is used in this model since it proves to be better than batch normalization when batch size is small, which equals to 4 in our case [57].

Figure 3.11: The building block illustrating the lateral connection for the fusion of steerable pyramid in the SPAE.
Support Vector Machine

Support vector machine, also known as support vector network, is a type of supervised learning model, which aims to analyze data for classification or regression prediction [59]. To be more precise, a support vector machine constructs a hyper-plane or a set of hyper-planes in the space, which are then used to perform various kinds of tasks, e.g., classification, regression, or outlier detection. Explicitly, a hyper-plane constructed will be considered as a good one if it has the largest functional margin, which means the distance to the nearest training sample of any class. As a result, the classifier would have better generalization performance, that is, lower generalization error [60]. However, it often happens that the problem on the raw samples cannot be solved simply in a low-dimensional space. Therefore, the idea rose up that a mapping could be constructed from the original space to a higher-dimensional space, which is targeted to perform the task more easily in the new space. The mapping is designed by a kernel function to define the computation regarding the input data vectors [61]. Radial basis function (RBF) is a type of nonlinear kernel functions, which is widely applied to any distribution of the samples. The schematic structure of a nonlinear SVM is shown in Fig. 3.12. It takes a binary classification in two-dimensional space as an example, where the orange scatters and blue scatters belong to two different classes. The black curve represents the decision function constructed to separate the training data points of these two classes. Besides, the dotted curves demonstrate the tolerance of the decision boundary on the misclassifications as introduced above.

![Figure 3.12: The schematic structure of a nonlinear support vector machine.](image)

The RBF kernel function can be expressed as:

\[ K(x_m, x_n) = \exp(-\gamma \| x_m - x_n \|^2) \]  

(3.12)

where \( x_m \) denotes the data base point, and \( x_n \) represents the testing data point for the mapping.
3.3. The Steerable Pyramid Autoencoder based Framework

\( \gamma \) is a parameter defined for the kernel function, which will be introduced in detail in the following.

With the definition of the kernel function, the problem the support vector machine aims to solve is the minimization problem denoted as follows:

\[
\min_{\alpha_m} \frac{1}{2} \sum_{m=1}^{N} \sum_{n=1}^{N} y_m y_n \alpha_m \alpha_n \exp(-\gamma \|x_m - x_n\|)^2 - \sum_{m=1}^{N} \alpha_m \tag{3.13}
\]

Among s.t. \( \sum_{m=1}^{N} y_m \alpha_m = 0, \quad 0 < \alpha_m < C \)

where each \( x_m \) is a multi-dimensional vector and \( y_m \) is the corresponding label of the data point \( x_m \). Also, it is the same for \( x_n \) and \( y_n \). \( N \) is the number of the training samples in the dataset. The minimum value of the problem (3.13) depends on the choice of the parameters (C, \( \gamma \)) [62].

Intuitively, the C parameter can be regarded as a penalty parameter, which imposes a penalty for each misclassified sample. The purpose of defining C is to seek for a trade-off between the correct prediction of the training samples against the maximization of the hyper-plane margin. In other words, smaller value of C leads to lower penalty on the misclassified samples, and thus a larger margin of the decision hyper-plane is prone to be selected with certain prediction errors accepted. On the contrary, larger value of C results in higher penalty on the misclassifications, that is, the classifier tries to minimize the misclassification rate. Hence, the decision boundary constructed by the classifier will encourage a smaller margin. On the other side, the \( \gamma \) parameter defines the distance of the influence of a single training sample. To be more specific, lower \( \gamma \) indicates that the decision hyper-plane has a larger radius, which is able to group more training samples. However, higher \( \gamma \) means the radius of the decision boundary is smaller. In this case, only the samples close to each other are prone to be grouped in the same class. Therefore, a support vector machine with much higher \( \gamma \) tends to be over-constrained on the training data, which means it tends to be overfitting. As a result, this classifier will not achieve promising accuracy for the prediction on the testing data.

As illustrated in Fig. 3.10, the representative features extracted from the feature extractor are then fed into SVM classifier. Radial Basis Function (RBF) kernel is applied in the SVM conducted in this experiment. The \( \gamma \) parameter of the classifier is selected according to different experimental methods. Specifically, it is adjusted on the basis of varied dimensions of the features extracted from the feature extractors. In this study, the \( \gamma \) is set as 0.001 for SED and CED as well as 0.003 for HOG and AE. As for VAE, the \( \gamma \) of the classifier is selected as 0.1. Additionally, the \( \gamma \) is adapted as \( 6 \times 10^{-5} \) for CAE and SPAE. The penalty parameter C is
3.4 The Log-Gabor Autoencoder based Framework

The proposed log-Gabor autoencoder (LGAE) based framework involves unsupervised feature extraction and supervised classification, which is similar to the SPAE-based framework. Log-Gabor filter is integrated into the autoencoder-based network to enhance the hard mining capability. The detailed structure is introduced in succession below.

3.4.1 Framework Structure

The proposed log-Gabor autoencoder (LGAE) based framework is formed by log-Gabor autoencoder and support vector machine (SVM) [19], which is similar to the schematic structure of SPAE-based framework. Log-Gabor autoencoder is performed for extracting features, which are then fed into the support vector machine for prediction. The schematic structure of the LGAE-based framework is shown in Fig. 3.13. The top flow of the diagram demonstrates the training procedure, while the bottom flow illustrates the testing process of the pre-trained LGAE and SVM classifier.

Figure 3.13: The LGAE-based framework for key-frame identification.

As introduced in Section 3.3.1, autoencoder is learning the features layer by layer in depth, not in multi-scale or multi-orientation. It is crucial to strengthen the feature learning and extraction, namely, to obtain more representative features from the images. Hence, it can help assist the establishment of the decision hyper-plane of the classifier. In this study, it is proved that enhancing edge feature detection of the anomalies captured on the key frames can be helpful for
3.4. The Log-Gabor Autoencoder based Framework

The identification work. To date, image feature detection has gained a lot of research attention in many fields. It has been extensively researched in computer vision, pattern recognition, and content-based image retrieval. For image feature detection, multi-scale representation learning is widely implemented for image analysis and feature detection. Hence, multi-scale image analysis can be considered to improve the accuracy of image edge detection. Log-Gabor filter has led to a renewed interest in the detection of image edge features. It has outstanding performance in many fields such as pattern recognition and computer vision, because the response of the filter may form a good representation for the application purpose. Hence, it is considerable to integrate the log-Gabor filter with the neural network to gain higher accuracy than that of the SPAE proposed in Section 3.3. Similarly, log-Gabor filter is integrated into the encoder part of the autoencoder to perform multi-scale feature learning. Hence, a log-Gabor autoencoder is developed, which is to fuse an autoencoder network with log-Gabor filter, aiming to enhance the hard mining on video frames. The LGAE takes advantage of both pre-defined and automatically extracted features, which is the superiority of LGAE.

3.4.2 Computational Methodologies

The LGAE-based framework is developed on the basis of the log-Gabor technique and a customized autoencoder-based network. The computational methodologies are investigated in succession below.

Log-Gabor Filter

In signal processing, space and frequency are two characteristics of a signal. It is impossible to locate the part of a signal, which produces a specific frequency based on Fourier transform. Although a short-time Fourier transform can be helpful on this, it limits the basis functions to be sinusoidal. In this case, log-Gabor filter has been proposed to provide a space-frequency signal decomposition with higher flexibility, which is an improvement upon the original Gabor filter. Compared with other alternative filters, it is suggested that log-Gabor filter can better encode natural images because it can better fit the statistics of those images. In pattern recognition, a representation, also known as extracted features will be obtained from the input images, which can then be fed into a classifier for a classification task. Log-Gabor filter can be widely applied in this field since the response of the filter may form a good representation, that is, a good set of representative features for the application purpose. For instance, the log-Gabor filter has been successfully implemented in face expression classification [63].

To be more specific, log-Gabor filter has two important features. The functions of log-Gabor filter have no DC component (the mean amplitude of the wavelet function). In other
3.4. The Log-Gabor Autoencoder based Framework

words, the DC component of the log-Gabor filter equals to zero. On the other hand, there is an extended tail at the high-frequency end of log-Gabor transfer function. Precisely, the function of the original Gabor filter includes Gaussian transfer when viewed on the linear frequency scale, while that of log-Gabor filter contains Gaussian transfer when viewed on the logarithmic frequency scale [64]. Thus, log-Gabor function can better reflect the frequency response of a natural image compared with the original Gabor filter, which is demonstrated on the linear frequency scale [65]. The transfer function of log-Gabor function has the form:

\[ G(f) = e^{-\frac{(\log(f/f_0))^2}{2(\log(\omega_f/f_0))^2}} \]  

(3.14)

where \( f_0 \) and \( \omega_f \) are the parameters of the log-Gabor filter. \( f_0 \) shows the filter’s centre frequency and \( \omega_f \) represents the bandwidth of the filter. To maintain consistent shape, the \( \omega_f/f_0 \) should be kept constant across varied frequency parameter \( f_0 \). Accordingly, \( G(f) \) is the frequency response of a log-Gabor filter.

The last few years have seen an increased interest in log-Gabor filter for image processing. Hence, the log-Gabor filter is considered for the 2-dimensional extension [66]. To be more precise, the extended filter is designed for a particular frequency as well as a particular orientation together. The orientation component of the filter is defined as a Gaussian distance function with regard to the angle in polar coordinates. The radial component of the 2-dimensional filter is described the same as 3.14, while the orientation component has a response denoted by:

\[ G(\theta) = e^{-\frac{(\theta-\theta_0)^2}{2\omega_\theta^2}} \]  

(3.15)

where \( \theta_0 \) represents the center orientation, and \( \omega_\theta \) is the width parameter of the orientation.

With 3.14 and 3.15, the response of the 2-dimensional log-Gabor filter can then be described as 3.16 in succession below. The filter can be divided into two components, namely radial filter and angular filter (also known as orientation filter). Thus, the 2-dimensional log-Gabor filter results from the combination of the functions for two components.

\[ G(f, \theta) = G(f) \cdot G(\theta) \]  

(3.16)

where \( (f, \theta) \) indicates the polar coordinates.

A golden image pyramid of a sample frame is represented in Fig. 3.14(b) with eleven successive levels, which affects the edges of the objects on the input frame at varied scales by using oriented log-Gabor filter. Meanwhile, the raw frame is also shown in Fig. 3.14(a). The aspect ratio, namely, golden section, is selected as \( \varphi = \frac{5}{2} + 1 \) and each level of the pyramid corresponds to the orientation performed by log-Gabor filter. The down-scaling factor is de-
3.4. The Log-Gabor Autoencoder based Framework

fined as \( \phi^2 \). The deeper the pyramid level is, the more evident the effect of the log-Gabor filter is. In this study, the first level of the golden pyramid is selected to be incorporated into the autoencoder-based network in case of excessive effect of the filter.

Figure 3.14: Sample video frame and the golden pyramid of log-Gabor filter on the sample frame: (a) sample video frame; (b) the golden pyramid of the log-Gabor filter.

From the formulas above, it can be seen that the log-Gabor filter is determined by four parameters: \( f_0, \theta_0, \omega_f \) and \( \omega_\theta \). Precisely, \( f_0 \) and \( \theta_0 \) correspond to the local frequency and orientation of the specific dataset. Besides, \( \omega_f \) and \( \omega_\theta \) are determined based on the experience or filter fitting on the dataset [67]. Regarding the detailed definition, the \( \omega_f \) is set to be 1.8, while \( \omega_\theta \) is defined as 0.1589 according to the filter fitting and empirical information on the dataset. Besides, \( f_0 \) is 92.29 and \( \theta_0 \) is 0.4053 for the dataset of this study. Above all, the first level of the golden image pyramid is selected to be incorporated into the autoencoder-based neural network.

Log-Gabor Autoencoder

In the log-Gabor autoencoder, mean squared error (MSE) is applied for training the model, which is the same as the SPAE introduced in Section 3.3.2.

The proposed log-Gabor autoencoder is also developed based on a customized convolutional autoencoder network, where the encoding structure is used as feature extractor. Fig. 3.15 illustrates the model structure of LGAE, where the layer with one purple node indicates the fusion layer of log-Gabor filtered features. As mentioned, the first level of the log-Gabor image pyramid is selected to be incorporated into the autoencoder-based network according to the quality of the extracted pyramid features as well as in case of excessive effect of the filter. Hence, each input sample will come with one log-Gabor filtered feature map to be input into the network. To be more precise, one convolutional layer and one batch normalization layer are applied after the input layer of log-Gabor features in the feature fusion. The input log-Gabor features are then combined with encoded features from the autoencoder network through an addition layer. Also, LGAE consists of the small ResNet module represented by a green block.
3.4. The Log-Gabor Autoencoder based Framework

Figure 3.15: The overall structure of the log-Gabor autoencoder based framework.
3.4. The Log-Gabor Autoencoder based Framework

Figure 3.16: The model construction of the proposed LGAE.
3.5 Evaluation Metrics

In Fig. 3.15 to enhance the feature propagation. Group normalization (GN) is used in the small ResNet module since it proves to be good during the feature propagation when batch size is small. The yellow layer represents the upsampling layer with the size parameter defined as 2. Fig. 3.16 demonstrates a detailed diagram of the model construction for the LGAE. The diagram displays the backbone of LGAE, the lateral connection for the fusion of log-Gabor, and the structure of the small ResNet module.

As for the classifier, the representation extracted from the LGAE is fed into the SVM classifier with the Radial Basis Function (RBF) kernel. The penalty parameter C is defined as 1.0. As for the $\gamma$ parameter, it is adapted as $6 \times 10^{-5}$ for LGAE, which is the same as that of SPAE.

To summarize, LGAE-based framework is proposed to conduct key-frame identification for water pipeline CCTV inspection. Precisely, a feature extraction model LGAE is carried out based on the integration of log-Gabor filter with a customized autoencoder-based network to enhance the hard mining on the video frames for the purpose of improving the accuracy of key-frame identification.

### 3.5 Evaluation Metrics

The performance of the proposed key-frame identification framework is validated using the contingency table of the binary classifier. Precisely, the contingency table is shown in the following Table 3.1. For the classification task in the thesis, there are two classes of the pipeline data, one of which is normal frame, and the other is key frame (also known as anomaly frame). The classifier will predict each input as belonging to one class, namely, predicted class, while the input also has its own actual class. The contingency table aims to compare the prediction performance for each class of the data. Specifically, it provides a basic picture of the interrelation between two variables and can help find interactions between them. Assuming that one dataset is labeled into two categories, the categories can be denoted as variables, namely, ‘0’ and ‘1’. As is demonstrated, the first column shows the actual variable, and the first row presents the predicted variable. Thus, the interrelation between the actual variable and the predicted variable can be illustrated through the calculation. Normally, the variable 1 is regarded as positive, while 0 is considered as negative. Accordingly, true positive is denoted as TP, and false positive is indicated as FP. Besides, true negative is represented as TN, while false negative is denoted as FN.

Accuracy, precision, recall, and F1-score are four important evaluation metrics calculated from the contingency table. The equations of these four evaluation metrics are given below:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (3.17)$$
3.5. Evaluation Metrics

<table>
<thead>
<tr>
<th>Actual/Predicted</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>TN</td>
<td>FP</td>
</tr>
<tr>
<td>1</td>
<td>FN</td>
<td>TP</td>
</tr>
</tbody>
</table>

Precision = \( \frac{TP}{TP + FP} \)  \hspace{3cm} (3.18)

Recall = \( \frac{TP}{TP + FN} \)  \hspace{3cm} (3.19)

\[
\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \hspace{3cm} (3.20)
\]

where \( TP \) is the true positive (correctly detected positive ones). \( FP \) is the false positive (falsely predicted as positive). Similarly, \( TN \) is the true negative (correctly detected negative ones) and \( FN \) is the false negative (falsely predicted as negative).

Explicitly, accuracy represents the ratio of the correctly predicted observations to the total observations. Precision shows the ratio of the correctly predicted anomaly observations to the total predicted anomaly observations. Recall denotes the ratio of the correctly predicted anomaly observations to the all observations in actual class. F1-score indicates the weighted average of precision and recall. Hence, recall is one of the most essential metrics on account of the purpose of the study, which is to identify the anomaly frames contained in the videos as many as possible.

Receiver operating characteristic (ROC) curve and precision-recall (P-R) curve are considered as another two important evaluation plots of the prediction model skill. ROC curve is a plot of the false positive rate (x-axis) versus the true positive rate (y-axis) for a group of different candidate threshold values between 0.0 and 1.0. AUC is the area under the ROC curve. The precision-recall curve is a plot of the recall (x-axis) and the precision (y-axis) for a set of different thresholds.

True positive rate (TPR) and false positive rate (FPR) can be described as following equations:

\[
\text{TPR} = \frac{TP}{TP + FN} \hspace{3cm} (3.21)
\]

\[
\text{FPR} = \frac{FP}{FP + TN} \hspace{3cm} (3.22)
\]

where \( TP \) is the true positive value as mentioned above. \( FP \) is the false positive value and \( TN \)
3.5. Evaluation Metrics

is the true negative value. Besides, $FN$ is the false negative value.

For deep neural networks, there is a concern of overfitting problem, which is frequently encountered in the model training process. Therefore, $k$-fold cross-validation is conducted to evaluate the experimental methods. In $k$-fold cross-validation, the raw dataset is randomly split into $k$ equal sized subsamples. Among the $k$ subsamples, one subsample is retained as the testing data of the model, and the remaining $k - 1$ subsamples are used as training data. The cross-validation process is then repeated $k$ times, where each subsample is used as the testing data one by one, which is displayed in the diagram of Fig. 3.17. $k$ results can be collected and then be calculated to produce an average estimation. The goal of cross-validation is to test the model’s ability to predict new data that was not used in estimating it, in order to prove against problems like overfitting. The $k$ value can be larger if the raw data is a very small amount of dataset. For the dataset with 5,000 samples in our study, $k$ value is selected as 5. Thus, five-fold cross-validation is implemented in this study to provide an accurate estimate of the model performance.

Figure 3.17: The diagram for $k$-fold cross-validation.

Temporal regularity is another evaluation metric applied in the visualization of key-frame identification. It is represented as the probability value from the classifier to detect each frame as key frame. For the classifier, the frame with a probability value larger than the threshold will be classified as key frame, whereas the frame with a probability value less than the threshold is identified as normal frame. Therefore, the pattern of the video is presented, where a peak in the pattern probably indicates there are key frames identified.
3.6 Summary

In this chapter, an unsupervised learning-based framework is first implemented for the key-frame identification on the water pipe dataset. Related state-of-the-art autoencoder networks are introduced, which are applied in the framework. For the purpose of further improving accuracy, SPAE-based framework is then proposed to conduct key-frame identification for water pipeline CCTV inspection. Explicitly, a feature extraction model SPAE is carried out based on the integration of steerable pyramid with a customized autoencoder network to achieve outperforming performance of representation learning. Thirdly, LGAE-based framework is proposed to enhance the hard mining on video frames. Precisely, the LGAE-based framework is composed of a log-Gabor autoencoder and a classifier, which has a similar schematic structure to that of the SPAE-based approach. The log-Gabor filtered features are incorporated into the autoencoder-based network, constituting the LGAE to achieve superior feature learning and generation. Finally, the crucial evaluation metrics are introduced for the estimation of the experimental methods in the study.
Chapter 4

Experimental Results and Discussions

4.1 Data Description

In this study, PipeDiver® [9] was employed as a platform to collect the inspection data. The robotic platform is equipped with a lighting array and multiple cameras. The cameras are attached to this tool from different angles to conduct CCTV inspection and collect video data. Specifically, this study employed the video data collected from the PipeDiver® through traveling along the pipelines. There are two datasets collected from two different water pipelines. The first dataset contains 5,000 video frames obtained from water pipeline CCTV videos in 2018, which were labeled into two categories, normal frames, and key frames, respectively. The key frames include various kinds of pipe anomalies such as air pocket, longitudinal crack, and different types of outlets. This dataset is mainly applied for fitting and testing the selected experimental methods. The second dataset contains 880 labeled video frames obtained in 2015, among which the key frames include several types of pipe anomalies, i.e., debris, outlet, and crack. This dataset is only used for estimating the generalization performance of the proposed methods, which will be introduced in the corresponding section below.

4.2 Experimental Setup

The experiments were conducted on the computer with i7 – 8750H processor and NVIDIA GeForce GTX 1070 Graphic Processing Unit (GPU). The training dataset contains 5,000 video frames obtained from water pipeline CCTV videos in 2018, which were labeled into two categories, normal frames (2,560) and key frames (2,440), respectively. The key frames include various kinds of pipe anomalies, i.e., air pocket and different types of outlets.

For the unsupervised learning-based framework, the raw frames were pre-processed and resized to $512 \times 512$ pixels with one channel for the purpose of controlling computational memory. The channel is selected appropriately with acceptable quality. Since only the normal frames were used for training the autoencoder-based networks, the normal frames were shuffled and split into training set (60%), validation set (20%), and testing set (20%). In addition, the key frames were used only for testing the models. Hence, the key frames were shuffled,
4.3 Results and Discussions

Among which 512 frames were randomly selected for testing the model performance. The initial epochs of training the neural networks were set to be 50 with batch size as 4. Furthermore, adam [56] was configured to be the optimizer of training the neural networks.

For the SPAE and LGAE based frameworks, five-fold cross-validation was conducted to avoid overfitting problem in model selection. The raw frames were pre-processed and resized to $512 \times 512$ pixels with one channel, which is the same as that of previous framework. In the five-fold cross-validation, the dataset was shuffled and split into five subsamples, where each subsample is retained as the testing data one by one, and the remaining subsamples are used for training. Namely, 4,000 video frames were used for training the model, and 1,000 frames were used for testing in each iteration of the cross-validation. To be more specific, there are 2,048 normal frames and 1,952 key frames in the training set. The testing set consists of 512 normal frames and 488 key frames. The initial epochs of training the neural networks were set to be 50 with batch size as 4. Furthermore, adam was configured to be the optimizer of training the neural networks. To avoid overfitting problems, early-stopping was applied in the training procedure of the neural networks. In addition, dropout layer with the rate of 0.2 was implemented in the SPAE and LGAE. Furthermore, group normalization with the groups of 4 is also able to help prevent overfitting in practice. As for CNN, weight regularization was applied for the layer’s kernel and bias. Regularizer with both L1 and L2 penalties was created with the factor of 0.001 for each penalty. Meanwhile, dropout layer with the rate of 0.3 was configured in the model.

4.3 Results and Discussions

The study involves three different frameworks for key-frame identification on water pipeline CCTV videos. For each framework, comparative studies were conducted, and the experimental results were collected.

4.3.1 The Unsupervised Learning-based Framework

As introduced in the Section 3.2, this framework is based on unsupervised learning, where the state-of-the-art autoencoders are only trained with normal frames. Specifically, the autoencoders will learn and extract the representative features of a normal frame, outputting a reconstructed frame with corresponding reconstruction loss. In the training process, the autoencoders try to reconstruct the raw frames as accurately as possible, namely, with lower reconstruction loss. In other words, the autoencoders do not learn to extract the features of the key frames. Thus, they are not able to reconstruct the key frames as accurately as the normal
4.3. Results and Discussions

ones, which means the loss values of the key frames are much higher than those of the normal ones. In the study, normal video frames are fed into the autoencoders, and the training iterations are set to be 50.

**Comparative Analysis**

The training loss curves of three experimental methods are illustrated in Fig. 4.1. As can be observed, the training loss of each method shows a stable decrease along with the training iterations. In Fig. 4.2, the models are tested on 1,024 samples, among which the first 512 samples are normal frames, while the last 512 ones are anomaly frames. It can be observed that the normal and anomaly frames are more prone to be separated by AE through applying an appropriate threshold on the loss values. However, the losses of normal and anomaly frames from CAE and VAE show certain overlapping, which results in the lower identification accuracy of the anomaly frames.

![Figure 4.1: The training curves for the loss of the experimental models.](image)

For better visualization, a loss distribution graph is plotted for each experimental method. On the x-axis of the graph, the first 512 samples are normal frames, while the last 512 ones are anomaly frames. To be more specific, blue scatters represent the reconstruction errors of normal video frames, while the orange ones are those of the anomaly frames. A threshold is defined for each method to perform the detection task, which is indicated by the red line in the loss distribution graph. Hence, it can be observed in Fig. 4.3 that the majority of the loss
4.3. Results and Discussions

Figure 4.2: The reconstruction loss of the testing samples.

Figure 4.3: Loss distribution of the autoencoder.

values of anomaly frames distribute approximately above the threshold line, while those of the normal ones distribute under the threshold line for AE. Apparently, the distributions of the loss values of two classes have larger aliasing for CAE in Fig. 4.4. The loss distribution of VAE is similar to that of AE, which is promising in Fig. 4.5.
4.3. Results and Discussions

The evaluation indices for the key-frame identification are shown in Table 4.1. Since recall represents the ability of predicting the anomaly frames correctly, it is one of the most important metrics for evaluating the experimental methods. Based on the comparison, the accuracy of the
AE is 0.940, higher than 0.833 of CAE and 0.921 of VAE. Regarding the F1-score, the AE also holds 0.943, which is higher than 0.849 of the CAE and 0.923 of the VAE. The fact that the F1-score of CAE is obviously lower is due to the low precision of CAE with the value of 0.772. As for the recall, AE shows better performance with the value of 0.988, however, the precision of AE is relatively low with the value of 0.901. CAE does not perform well because it can reconstruct the anomaly frames well, which leads to lower loss values of the anomaly frames. Hence, the conclusion can be drawn that AE outperforms the other two models in this framework on the detection task.

Discussions

In this unsupervised framework, one factor is crucial in terms of affecting the model performance, which is the defined threshold. The threshold determines the trade-off between the ratio of the correctly predicted normal frames and that of the correctly predicted key frames to the total observations of either class. To be more specific, the reconstruction losses of normal and anomaly frames have certain difference. If a threshold is defined properly, the frames with the losses below the threshold can be regarded as normal frames, whereas the frames with the losses above the threshold can be identified as key frames. Certainly, there are sample points distributed close to the threshold defined. Hence, it is not hard to understand that lower threshold leads to higher ratio of the correctly detected key frames to the total observations of this class. On the contrary, higher threshold results in an increase of the accuracy for identifying normal frames, but meanwhile, it leads to a lower recall metrics. Accordingly, the trade-off between these two metrics, i.e., precision and recall, is hard to some extent, which is a drawback of this framework. This is also the reason why the framework with higher recall above 0.940 shows relatively lower precision, which is below 0.910. On the other side, this framework is on the basis of unsupervised learning, where the feature learning model only learns the pattern of normal frames. Without the supervision of the data labels, it is difficult to have control over the learning process of the model. Therefore, the unsupervised learning-based framework does not show competitive performance in this task.
4.3. Results and Discussions

To summarize, comparative results regarding four evaluation metrics are analyzed and discussed to estimate the model performance. Additionally, loss distribution plots are provided to have a better visualization of the model performance on the key-frame identification. According to the experimental results, the unsupervised learning-based framework does not achieve promising performance on this task.

4.3.2 The Steerable Pyramid Autoencoder based Framework

The experimental results are illustrated regrading the experiments conducted on the CCTV inspection videos of the internal surface of pipe. Parameter optimization of the SPAE was also performed. Furthermore, the SPAE framework was compared with three image processing methods, and state-of-the-art autoencoder models. The proposed framework was validated with labeled video frames and the comparative results are presented. The SPAE framework was also tested on a video clip of the internal surface of water pipe.

Parameter Optimization

Parameters of the proposed SPAE were optimized by comparing the model performance regarding different numbers of hidden layers. On the other hand, varied numbers of orientations regarding the steerable pyramid features were also considered to optimize the SPAE model.

Hidden Layers in the SPAE  As shown in Fig. 3.11, the hidden representation of $4 \times 64 \times 64$ in SPAE was adopted to train the classifier, which meant each sample was represented by $4 \times 64 \times 64$ features. Hence, the representations of samples were predicted by the classifier, and accuracy can be computed. There are hidden layers for down-sampling in the encoder of the model between the layer of $64 \times 64 \times 64$ and the hidden representation of $4 \times 64 \times 64$. Dropout layer and early-stopping are applied as well in the training process to prevent from overfitting. From the perspective of reducing parameters, convolutional layer (Conv) is adopted in the latent space of SPAE.

Results of model performance regarding different numbers of hidden layers are compared in Table 4.2. It can be observed that SPAE performs best with two Convs in the latent space, achieving the highest accuracy of 0.975. For one Conv and three Convs, the former one wins with an accuracy of 0.970 compared with 0.955 of the latter one. The fact that they show worse results indicates that neither more layers or less layers achieve improvement for the model performance. Therefore, two Convs are used for down-sampling to the representation of $4 \times 64 \times 64$. 
### 4.3. Results and Discussions

#### Table 4.2: Parameter optimization of the hidden layers

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Evaluation Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
</tr>
<tr>
<td>One Conv</td>
<td>0.970</td>
</tr>
<tr>
<td>Two Convs</td>
<td><strong>0.975</strong></td>
</tr>
<tr>
<td>Three Convs</td>
<td>0.955</td>
</tr>
</tbody>
</table>

#### Orientations of Steerable Pyramid

In a steerable pyramid, two types of filters are applied for extracting features; one of which is high-pass filter and the other is low-pass filter. Basically, features can be extracted from one to four orientations with corresponding filters at $0^\circ (180^\circ)$, $45^\circ (225^\circ)$, $90^\circ (270^\circ)$ or $135^\circ (315^\circ)$, respectively. The filters are defined accordingly to extract features from different orientations. Table 4.3 shows the evaluation of the model performance with different steerable orientations. It is concluded that SPAE performs best on key-frame identification with four steerable orientations, achieving the highest F1-score of 0.988 and AUC of 0.989.

#### Table 4.3: Parameter optimization of the steerable orientations

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Evaluation Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
</tr>
<tr>
<td>One orientation</td>
<td>0.975</td>
</tr>
<tr>
<td>Two orientations</td>
<td>0.969</td>
</tr>
<tr>
<td>Three orientations</td>
<td>0.980</td>
</tr>
<tr>
<td>Four orientations</td>
<td><strong>0.989</strong></td>
</tr>
</tbody>
</table>

#### Comparative Analysis

Five-fold cross-validation is conducted to avoid overfitting problems. The validation results were collected across five subsamples and the mean value as well as standard deviation of each method were calculated. The results of accuracy and recall are displayed in the Table 4.4 as they are two of the most important evaluation indices in the experiment. As can be observed, the CED and HOG achieve lower accuracy across five subsamples. Although SED performs higher accuracy, it fails to detect key frames accurately, which results in the lower recall. Specifically, there should be a balance between precision and recall for the method selection. As for the neural networks. They perform well with a much higher accuracy over 0.950. Nevertheless, the performance on recall should be considered as recall is one of the most important evaluation
4.3. Results and Discussions

It can be observed that SPAE outperforms the other selected methods across five subsamples with the recall higher than 0.980. In terms of accuracy, CNN performs better than SPAE on the first subsample with the accuracy of 0.976, whereas SPAE wins on the other four subsamples.

Figure 4.6: The bar chart of the accuracy in the cross-validation of SPAE-based framework.

Figure 4.7: The training history of SPAE.

The bar chart of the accuracy across five subsamples with the error bar of the mean accuracy is displayed in Fig. 4.6, where several neural networks show good performance. The black line on top of each error bar denotes the standard deviation of the corresponding method in five-fold cross-validation. SPAE shows better performance than the other selected neural networks. In addition, the comparative results of the precision and F1-score is shown in Table 4.5 with
Table 4.4: The comparison of the accuracy and recall from five-fold cross-validation

<table>
<thead>
<tr>
<th>Method</th>
<th>Fold Index</th>
<th>Mean</th>
<th>Recall</th>
<th>Fold Index</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SED [18]</td>
<td>0.948 0.948 0.946 0.949 0.947±0.001</td>
<td>0.893 0.893 0.889 0.889 0.895</td>
<td>0.892±0.002</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CED [18]</td>
<td>0.836 0.860 0.848 0.829 0.851±0.011</td>
<td>0.668 0.727 0.702 0.659 0.700</td>
<td>0.692±0.025</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HOG [19]</td>
<td>0.902 0.893 0.886 0.911 0.896±0.008</td>
<td>0.862 0.854 0.854 0.827 0.821</td>
<td>0.844±0.016</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNN [22] 0.976</td>
<td>0.968 0.958 0.973 0.960 0.967±0.007</td>
<td>0.950 0.934 0.913 0.952 0.918</td>
<td>0.934±0.016</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AE [22] 0.975</td>
<td>0.940 0.960 0.963 0.964 0.960±0.011</td>
<td>0.948 0.877 0.918 0.924 0.926</td>
<td>0.919±0.023</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAE [26]</td>
<td>0.961 0.945 0.948 0.949 0.970</td>
<td>0.920 0.887 0.893 0.895 0.938</td>
<td>0.907±0.019</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VAE [29]</td>
<td>0.953 0.945 0.950 0.951 0.949</td>
<td>0.903 0.889 0.897 0.905 0.903</td>
<td>0.900±0.006</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPAE</td>
<td>0.975</td>
<td>0.987 0.968 0.995 0.995</td>
<td>0.984±0.010 0.995 0.981 0.991 0.989</td>
<td>0.990±0.005</td>
<td></td>
</tr>
</tbody>
</table>
Table 4.5: The comparison of the precision and F1-score from five-fold cross-validation

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fold Index</td>
<td>Mean</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>SED [18]</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>CED [18]</td>
<td>0.993</td>
<td>0.980</td>
</tr>
<tr>
<td>HOG [19]</td>
<td>0.931</td>
<td>0.920</td>
</tr>
<tr>
<td>CNN [22]</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>AE [22]</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>CAE [26]</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>VAE [29]</td>
<td>1.000</td>
<td>0.997</td>
</tr>
<tr>
<td>SPAE</td>
<td>0.954</td>
<td>0.991</td>
</tr>
</tbody>
</table>
regard to eight experimental methods. It can be observed that SED, AE, and CAE all achieve the best performance on the precision index with the value of 1.000 on five subsamples in the cross-validation. Also, CNN shows good precision of 1.000 on four subsamples. Explicitly, these methods is prone to detect the normal frames correctly, that is, less likely to misidentify normal frames as key frames. However, they also have a larger probability of falsely predicting the key frames as normal frames. Thus, based on the weighted average of precision and recall, the SPAE has outperforming performance on the F1-score compared with other methods. The training history of SPAE is also illustrated in Fig. 4.7, which presents the training loss and validation loss in different training iteration. It can be seen that the training and validation loss are descending stably. The curves reach the optimal level, keeping flat after around 25 iterations.

The proposed method is compared with existing state-of-the-art methods in Table 4.6 with respect to five important evaluation metrics. The results show the average performance of the experimental methods with standard deviation based on five-fold cross validation. Since recall represents the ability of predicting the key frames correctly, it is one of the most important metrics for evaluating the experimental methods. First, SED, CED and HOG were conducted and classification was performed by SVM with the same kernel. From Table 4.6, it can be found that CED achieves the lowest accuracy of 0.844 in this work. Although, SED and HOG have higher accuracy than CED with 0.947 and 0.896, respectively, the results regarding the recall of these two methods are not very promising. The poor performance indicates that these three image processing methods are not suitable on this dataset, which could result from the low illumination problem of the dataset. For better visualization, the feature extractions by SED, CED and HOG are tested on one testing sample and the feature maps are illustrated in Fig. 4.8(a). As observed, the testing sample shows a small and unclear anomaly (outlet) on the internal surface of the pipe. However, the features extracted by SED are not clear due to the illumination. The CED detect a few features of the anomaly but not necessarily enough for the detection task. As for HOG, it fails to extract adequate features of the unclear anomaly on the frame. On the contrary, the CNN performs much better than the previous methods with an accuracy of 0.967, which implies that the deep learning model is preferable in this work. CNN performs well especially on precision with a value of 0.998. Nevertheless, the recall of CNN is reported as 0.934 with the SD of 0.016, which is relatively lower compared with the other metrics of it.

In this study, three state-of-the-art autoencoders were implemented, which were AE, CAE and VAE, respectively. After that, the hidden representations of the models were adopted to train the SVM and the prediction results were computed and illustrated in Table 4.6. Among three encoder-based models, the AE is better than CAE and VAE with the accuracy and AUC
4.3. Results and Discussions

Table 4.6: Evaluation of key-frame identification on all the experimental methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Evaluation Metrics</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SED [18]</td>
<td></td>
<td>0.947±0.001</td>
<td>1.000±0.000</td>
<td>0.892±0.002</td>
<td>0.943±0.001</td>
<td>0.946±0.001</td>
</tr>
<tr>
<td>CED [18]</td>
<td></td>
<td>0.844±0.011</td>
<td>0.986±0.005</td>
<td>0.692±0.025</td>
<td>0.813±0.016</td>
<td>0.841±0.011</td>
</tr>
<tr>
<td>HOG [19]</td>
<td></td>
<td>0.896±0.008</td>
<td>0.938±0.028</td>
<td>0.844±0.016</td>
<td>0.889±0.008</td>
<td>0.895±0.009</td>
</tr>
<tr>
<td>CNN [22]</td>
<td></td>
<td>0.967±0.007</td>
<td>0.998±0.003</td>
<td>0.934±0.016</td>
<td>0.965±0.008</td>
<td>0.966±0.007</td>
</tr>
<tr>
<td>AE [22]</td>
<td></td>
<td>0.960±0.011</td>
<td>1.000±0.000</td>
<td>0.919±0.023</td>
<td>0.958±0.013</td>
<td>0.959±0.012</td>
</tr>
<tr>
<td>CAE [26]</td>
<td></td>
<td>0.954±0.009</td>
<td>1.000±0.000</td>
<td>0.907±0.019</td>
<td>0.951±0.011</td>
<td>0.953±0.010</td>
</tr>
<tr>
<td>VAE [29]</td>
<td></td>
<td>0.949±0.002</td>
<td>0.996±0.003</td>
<td>0.900±0.006</td>
<td>0.946±0.003</td>
<td>0.948±0.003</td>
</tr>
<tr>
<td>SPAE</td>
<td></td>
<td>0.984±0.010</td>
<td>0.978±0.023</td>
<td>0.990±0.005</td>
<td>0.984±0.011</td>
<td>0.984±0.011</td>
</tr>
</tbody>
</table>

Figure 4.8: The comparison of the extracted features between the selected experimental methods.

of 0.960 and 0.959. However, these three autoencoders did not achieve improvement on recall compared with that of CNN. From the perspective of further improving the recall, we developed a steerable pyramid autoencoder that sought to further enhance the representation learning by integrating steerable pyramid into the autoencoder network. As shown in Table 4.6, the proposed model achieves best results on accuracy, F1-score and AUC. The fact that SPAE outperforms other methods proves the improvement by this fusing strategy, which strengthens the representation learning and extraction. Our proposed model has a much higher recall of
4.3. Results and Discussions

Figure 4.9: Illustration of the comparative results.

0.990 and accuracy of 0.984, because it explicitly extract more accurate representation, thus improves the prediction accuracy. It is worth noticing that the F1-score of SPAE is 0.984, which is much higher than all the other tested methods.

For better visualization of Table 4.6, bar charts are reported on three evaluation metrics, which are accuracy, F1-score and AUC. From Fig. 4.9(a), it can be noticed that the SPAE remains the highest bar on accuracy, whereas the CED has the lowest accuracy under 0.850. As stated in Fig. 4.9(b), the highest F1-score of SPAE indicates that this model performs best across precision and recall compared with other methods. Moreover, Fig. 4.9(c) shows that SPAE achieves the best result on AUC with 0.984, where CED performs worst with a low AUC of 0.841. Based on the comparison among all the experimental methods, the three image processing methods are not applicable for key-frame identification on our dataset. The poor
4.3. Results and Discussions

performance can be caused by the low illumination problem of the dataset. Additionally, these methods rely on more clear and useful information in data, which could be the reason why their performance is not good, especially in the aspect of recall. Furthermore, CNN shows promising results, however, it is not suitable in this work due to the relatively low recall. Among all the tested encoder-based models, the proposed steerable pyramid autoencoder outperforms the other models especially in terms of recall. It has explicitly outstanding prediction results on three essential metrics, namely, accuracy, F1-score and AUC.

![ROC Curve](image)

Figure 4.10: ROC curves of the deep learning methods in the study.

As discussed above, the deep learning methods conducted in this study show good performance in this work. According to the ROC curves in Fig. 4.10, it can be found that the star line (SPAE) stays highest compared with those of the others, accounting for the largest area under curve, which corresponds to the largest AUC value of 0.984. It proves that SPAE achieves the best result on true positive rate, leading to the best performance on identifying the key frames. As for the other tested models, the curves imply that they achieve higher false positive rates compared with SPAE, which means they are more prone to falsely predict the normal frames as key frames. In the precision-recall curves from Fig. 4.11, the curve from SPAE remains highest, above all the curves of other models, namely AE, VAE, CAE and CNN. The fact that SPAE shows the best performance in precision-recall curves implies that it has an advantage of maintaining good precision over other models, while achieving high recall close to 1.000.
4.3. Results and Discussions

The hidden representation of SPAE is compared with those of CNN and CAE in Fig. 4.8(b). It is worth noticing from the feature maps that CNN and CAE achieve good performance of feature learning from our raw data, but they do not generate sufficient representative features for samples with small and unclear defects. However, it can be found that SPAE is able to extract more edge features of the anomaly regions on the key frames. Thus, the proposed SPAE has compelling capacity of representation learning and extraction. It can be beneficial for the classifier to predict varied key frames out of the normal frames.

As is known, the early layer of an autoencoder technically learns the lower semantic information extracted from the input. Skip connections is able to compromise between short and long networks. Specifically, complex features can be developed in the layers close to the centre of the model with skip connections and lower semantic features can also be accessed at the same time from the earlier layers. The purpose of applying skip connection is to enhance the development of complex features in the latent space. Hence, the hidden representations of the SPAE can capture the complexity of the raw input and be more representative. A machine learning algorithm called t-distributed stochastic neighbor embedding (t-SNE) is always used for better visualization of high-level representations learnt by an artificial neural network. Specifically, it is a technique for embedding high-dimensional data for visualization in a low-dimensional space of two or three dimensions. Hence, 3-dimensional t-SNE feature distri-
4.3. Results and Discussions

Figure 4.12: 3-dimensional t-SNE feature distribution based on: (a) SPAE; (b) AE; (c) CAE; (d) VAE.

Figure 4.12 illustrates the feature generation of the aforementioned autoencoder-based models in the latent space. As can be observed in Fig. 4.12, the pink round scatters represent normal frames, whereas the blue star scatters represent the key frames in the space. It can be seen in the Fig. 4.12(a) that the SPAE is prone to extract the features of the normal frames into close clusters, while the features of the key frames distributed outside the clusters of normal frames. The scatters of key frames fall outside the relatively centralized clusters of normal ones, which makes hyper-plane prone to be established to identify the anomaly ones. In other words, the features extracted by SPAE from normal frames are aggregated to some extent, which helps assist the establishment of the decision hyper-plane of...
As for the other three models, they also generate separable features with certain degree of aliasing. Specifically, the features of the normal frames are generated into separate clusters with certain distance in the space through AE, which is displayed in Fig. 4.12(b). Nevertheless, the aggregation performance of AE is not very promising in terms of the feature extraction on normal frames, resulting in the decentralized feature distribution in the cluster of normal frames. As for the CAE, it achieves similar performance to that of the AE, as demonstrated in Fig. 4.12(c). Moreover, VAE shows stronger aggregation capability on the feature generation of normal frames than AE and CAE. Explicitly, VAE generates two separate clusters for normal frames, but the clusters of key frames have certain overlapping with those of the normal ones in Fig. 4.12(d). The other reason of implementing SPAE rather than other methods is that it outperforms the other selected methods on edge feature detection of the pipe anomaly on the video frame, which was illustrated in Fig. 4.8(b) above. The superiority can be further developed to perform soft proposal networks for weakly supervised anomaly localization in the future work. Hence, SPAE has practical advantage to be implemented on this water pipe dataset.

**Visualization of Key-Frame Identification**

The proposed SPAE-based framework was also tested on one video clip of the internal surface of water pipe. In water pipeline system, various types of pipe anomalies contain outlet, air pocket, debris, crack and so on. One type of pipe anomaly is included in the video clip, namely, outlet. The temporal regularity of the video frames is displayed through the probability value from the classifier. It is observed in Fig. 4.13 that the key frames highlighted by the pink background can be detected as large probability values. This framework shows compelling performance with most of the key frames correctly identified and only a few normal frames falsely predicted as key frames. With this offline video processing, the inspectors are able to assess only the key frames automatically obtained, which cost less time than manually checking the whole inspection video. Consequently, it will save time and labor for the operators to conduct further evaluation and assessment of the water pipeline.

To summarize, parameter optimization was conducted first to find the best structure of the proposed model SPAE. Five-fold cross-validation was then implemented, and results are illustrated to estimate the model performance. Bar charts are provided to have a better visualization of the comparative results. ROC curve and precision-recall curve are also applied to compare the models. Additionally, the feature extraction capacities of the selected experimental methods are discussed based on the feature maps and t-SNE distribution. Eventually, a visualization of the key-frame identification is presented. On the basis of the comparative results and
4.3. Results and Discussions

Figure 4.13: Temporal regularity of the CCTV inspection video.

discussions, the proposed SPAE shows good performance in the study.

4.3.3 The Log-Gabor Autoencoder based Framework

The LGAE-based framework was validated on the CCTV inspection data of the internal surface of the pipe, and the experimental results are demonstrated accordingly. Since it is a further improved version of the proposed SPAE-based framework, the experimental results and discussions are focused on the comparison between the LGAE as well as SPAE based frameworks. Also, the precision-recall curve and ROC curve are plotted to evaluate the newly proposed LGAE-based framework.

Comparative Analysis

Five-fold cross-validation is conducted to avoid overfitting problems. The validation results were collected across five subsamples, and the mean value was thus obtained accordingly. Standard deviation of each method was calculated as well. The results of five crucial evaluation metrics are demonstrated in the Table 4.7 with standard deviation. It is known that SPAE outperforms the other selected experimental methods in the Section 4.3.2. Also, LGAE-based framework is the further improved version of SPAE-based framework. Hence, the comparison is conducted only between SPAE and LGAE across five subsamples of the cross-validation in the table. As can be observed, LGAE outperforms SPAE across the first three subsamples with the accuracy over 0.990. Obviously, LGAE reports the impressing results with the 1.000 accuracy on the first subsample. As for the precision, LGAE does not perform very well on the last two subsamples compared with 1.000 precision reported by SPAE. Additionally, LGAE presents outstanding performance on the metric of recall. To be more explicit, LGAE
4.3. Results and Discussions

achieves 1.000 recall on three subsamples, which implies promising performance on the key-frame identification.

Table 4.7: The evaluation metrics from five-fold cross-validation on SPAE and LGAE

<table>
<thead>
<tr>
<th>Fold Index</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SPAE</td>
<td>LGAE</td>
<td>SPAE</td>
<td>LGAE</td>
<td>SPAE</td>
</tr>
<tr>
<td>1</td>
<td>0.975</td>
<td>1.000</td>
<td>0.954</td>
<td>1.000</td>
<td>0.995</td>
</tr>
<tr>
<td>2</td>
<td>0.987</td>
<td>0.994</td>
<td>0.991</td>
<td>0.989</td>
<td>0.981</td>
</tr>
<tr>
<td>3</td>
<td>0.968</td>
<td>0.992</td>
<td>0.945</td>
<td>1.000</td>
<td>0.991</td>
</tr>
<tr>
<td>4</td>
<td>0.995</td>
<td>0.982</td>
<td>1.000</td>
<td>0.964</td>
<td>0.989</td>
</tr>
<tr>
<td>5</td>
<td>0.995</td>
<td>0.975</td>
<td>1.000</td>
<td>0.951</td>
<td>0.989</td>
</tr>
<tr>
<td>Mean</td>
<td>0.984</td>
<td><strong>0.988</strong></td>
<td>0.978</td>
<td><strong>0.980</strong></td>
<td>0.990</td>
</tr>
<tr>
<td>SD</td>
<td>0.010</td>
<td>0.008</td>
<td>0.023</td>
<td>0.019</td>
<td>0.005</td>
</tr>
</tbody>
</table>

Figure 4.14: The comparison of the evaluation metrics between SPAE and LGAE.

For better visualization, the bar chart of the evaluation metrics averaged from five subsamples in the cross-validation is displayed in Fig. 4.14. The standard deviation is also shown on each bar with the black line on the top. As can be observed, LGAE achieves good results on each metric, which slightly outperforms SPAE. The fact that SPAE outperforms the other selected experimental methods implies the enhancement by this incorporation strategy, which strengthens the representation learning and extraction. This indicates that the SPAE explicitly extracts more accurate representation, thus improves the prediction accuracy. The log-Gabor autoencoder (LGAE) was proposed to further improve the accuracy of key-frame identification.
and obtain better generalization on varied water pipeline datasets. LGAE outperforms SPAE with the integration of log-Gabor filtered features into the autoencoder network. Precisely, the accuracy of LGAE is 0.988 higher than 0.984 of SPAE. In addition, the recall is reported as 0.996, which is improved by 0.006 compared with that of SPAE. Furthermore, the hidden representation of LGAE is extracted from the intermediate hidden layer and compared with that of SPAE in Fig. 4.15. It is worth noticing from the feature maps in Fig. 4.15 that SPAE achieves good performance of edge feature generation from our raw data, even though on the samples with small and unclear defects. Nevertheless, it can be observed that the LGAE is able to provide more frequency information of the frames with less aliasing. Hence, the proposed LGAE-based framework has compelling capacity of representation extraction, which is more separable for the classifier to predict varied key frames out of the normal frames.

![Feature Map](image)

Figure 4.15: The hidden representations from SPAE and LGAE compared with raw frames.

As mentioned in section 4.3.2, the deep learning methods show good performance in the key-frame identification. According to the plot of ROC curves in Fig. 4.16, it can be found that the cross orange line (LGAE) remains highest, accounting for the largest area under the curve, which corresponds to the largest AUC value of 0.988. The result proves that LGAE has the best performance on TPR, which means that LGAE achieves the best performance on key-frame identification. As for the other selected experimental methods, the curves imply that they achieve higher false positive rates than that of LGAE. Precisely, they are more prone to falsely predict the normal frames as anomaly frames. From the plot of precision-recall curves in Fig. 4.17, LGAE presents more competitive performance than the other models, namely AE, VAE, CAE, CNN, and SPAE. Especially, it is worth noticing that LGAE is slightly better than the proposed SPAE, which achieves the recall with a 0.006 higher value. The LGAE maintains
4.3. Results and Discussions

Figure 4.16: ROC curves of the deep learning methods in the study.

good recall and, in the meantime, achieves higher precision of 0.980 than 0.978 of the SPAE. Thus, the proposed LGAE is more superior in representation learning and extraction to the proposed SPAE. The fact results from that the features extracted by LGAE are more separable. In other words, the features obtained from LGAE are prone to be classified by the prediction algorithm.

**Computational Cost**

The computational costs of the methodologies were also investigated in the experiments. Table 4.8 presents the computational costs of three frameworks on a five-minute video. As can be observed, the unsupervised learning-based framework takes 69s, which is around one-fifth of the video length, to accomplish key-frame identification. For the SPAE-based framework, it takes longer time to process the video, which is 299s. Moreover, the LGAE-based framework takes 280s for the computational task on the video, which is close to the video length. The reason why the first framework spends less time on the task is that the computational model is shallower compared with the SPAE and LGAE.
4.3. Results and Discussions

Figure 4.17: Precision-recall curves of the deep learning methods in the study.

Table 4.8: Computational costs of three frameworks on a five-minute video

<table>
<thead>
<tr>
<th>Framework</th>
<th>Computational Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Unsupervised Learning-based Framework</td>
<td>69s</td>
</tr>
<tr>
<td>The SPAE-based Framework</td>
<td>299s</td>
</tr>
<tr>
<td>The LGAE-based Framework</td>
<td>280s</td>
</tr>
</tbody>
</table>

Discussions

Furthermore, the LGAE and SPAE based frameworks were tested on the second dataset, namely, the dataset from a different pipeline, for the purpose of discussing the generalization capability of the proposed models. To be more precise, the dataset contains 440 normal frames and 440 key frames, which were shuffled and split into training set (60%), validation set (20%), and testing set (20%). The second dataset was only used to fit a SVM classifier. Explicitly, the pre-trained feature extractor on the first dataset was used; however, a new SVM classifier was trained and fit on the second dataset. As can be observed in the Table 4.9, both the SPAE and LGAE based frameworks can achieve encouraging results on two datasets in the study. Especially, they report 1.000 accuracy on the second dataset. It implies that these two frameworks have good generalization capacity on these two datasets. Above all, due to
the lack of quantity and variations of the samples, it is prone to encounter overfitting problems while using a small amount of dataset to fit a neural network. However, the proposed LGAE has outstanding performance on the generalization capability. Specifically, there is no need to refit the existing feature extractor or fit a new feature extractor to address the task on the new dataset from another pipeline. The pre-trained LGAE can be applied directly on the new data, while a new SVM classifier needs to be fit. Generally, the LGAE-based framework can be conducted to perform key-frame identification on another dataset with only fitting a new SVM classifier on the dataset. This would save energy and time for the inspectors to carry out key-frame identification on varied pipeline datasets.

Table 4.9: Evaluation of key-frame identification between SPAE and LGAE on two varied datasets

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPAE</td>
<td>0.984</td>
<td>1.000</td>
<td>0.978</td>
<td>1.000</td>
<td></td>
</tr>
<tr>
<td>LGAE</td>
<td>0.988</td>
<td>1.000</td>
<td>0.980</td>
<td>1.000</td>
<td></td>
</tr>
</tbody>
</table>

To summarize, five-fold cross-validation was implemented on LGAE-based framework, and corresponding results are illustrated to estimate the model performance. Besides, the bar chart, ROC curves, and P-R curves are provided to have a better visualization of the comparative results. In addition, the feature extraction capabilities of the proposed LGAE and SPAE are compared based on the feature maps extracted from the intermediate layers. A discussion is also conducted on the generalization performance of two proposed methods. On the basis of the comparative results and discussions, the proposed LGAE-based framework achieves the best performance in this study.

4.4 Summary

In this chapter, the experimental results of three frameworks are presented and discussed with the proof of different evaluation plots. For the unsupervised learning-based framework, comparative results are analyzed regarding four essential evaluation metrics. Moreover, loss distribution plots are discussed to estimate the model performance on the key-frame identification. This approach is easy to conduct with less dataset required. For the SPAE-based framework, parameter optimization was conducted first to find the best structure of the proposed model SPAE. Five-fold cross-validation was then implemented, and results are illustrated to estimate the model performance. Bar charts are provided to have a better visualization of the
comparative results. ROC curve and precision-recall curve are also applied to compare the models. Additionally, the feature extraction capabilities of the selected experimental methods are discussed based on the feature maps and t-SNE distribution. Eventually, a visualization of the key-frame identification is presented. On the basis of the comparative results and discussions, the proposed SPAE shows good performance in the study. As for LGAE-based framework, the evaluation tables and related visualization plots are also illustrated. Discussions are majorly included in the comparison of SPAE and LGAE based frameworks. This framework achieves the best results among the three frameworks in the study.
Chapter 5

Conclusion

In this thesis, the autoencoder-based frameworks are proposed to address the challenges that exist in the office-based survey of water pipeline CCTV inspection. To the authors’ best knowledge, this is the first study to present autoencoder-based approaches to perform key-frame identification on water pipeline CCTV videos, which will facilitate the condition assessment of water pipeline. The main contributions can be listed as follows:

- To tackle the challenge of water pipeline CCTV inspection, the thesis first carries out an unsupervised learning-based framework for key-frame identification. This framework is easy to conduct with less dataset required, which achieves up to 0.940 accuracy with a relatively lower precision metric of 0.901.

- To improve the accuracy of key-frame identification, the thesis proposes a steerable pyramid autoencoder (SPAE) based framework, which is the first to incorporate the steerable pyramid into the autoencoder network. In addition, ResNet module and group normalization are employed to further enhance powerful representation learning and strengthen feature propagation of the SPAE framework. The SPAE enables representative and discriminative features to be generated and applied to the classifier. The experimental results show SPAE outperforms the other state-of-the-art methods in terms of selected evaluation metrics, namely, 0.984 accuracy, 0.990 recall, and 0.984 F1-score. The superiority of SPAE comes from its outstanding capability of representation learning and extraction by integrating features from multi-scale image pyramid. Thus, this framework will assure accuracy and greatly improve the efficiency for the survey of water pipeline CCTV inspection videos. The inspectors can focus on the information from the key frames to conduct water pipeline condition assessment.

- To further empower the capability of hard mining on the video frames, a log-Gabor autoencoder (LGAE) based framework is proposed to perform the identification task. The schematic structure of this framework is similar to the proposed SPAE-based framework, with the integration of log-Gabor filtered features instead of steerable pyramid features. LGAE takes advantage of feature detection over SPAE on the anomalies of the video frames. Comparative studies demonstrate that LGAE has an outperforming performance.
Chapter 5. Conclusion

with the accuracy of 0.988, the recall of 0.996, and the F1-score of 0.988. The fact that LGAE achieves better results than the other experimental methods implies that LGAE can generate a better set of features, which will then be employed into the classifier. Besides, LGAE shows superiority in the deep exploitation of the information on the video frames. Hence, LGAE-based framework will significantly assist the office-based survey on water pipeline CCTV inspection.

To conclude, in terms of water pipeline CCTV inspection, this study proposes an automated office-based survey, which is key-frame identification to investigate varied pipe deficiencies. Hence, the inspection of PipeDiver® platform is empowered for the preparation of further operation in the water mains system. Additionally, the workload of the inspectors is reduced, namely, the inspectors are less prone to get fatigue on account of watching hours of recorded videos. The manual inspection can not ensure the consistency and quality of the office-based survey. However, with the proposed framework, they can focus more on the information of the identified key frames for further pipe assessment. From the perspective of the methodologies, two autoencoder-based frameworks are proposed based on the fusing strategy of pre-defined features and customized deep neural networks. They are both superior in the deep exploitation on the underwater video data, which has overcome the illumination limitation of the raw data. Besides, the proposed SPAE and LGAE have the encouraging capacity of generalization for feature learning on different datasets.

The thesis proposes promising approaches for water pipeline CCTV inspection, which is accurate and robust. Nonetheless, the research is not intact in the thesis. There are many possible future extensions for the proposed method. The potential future studies for the researches are discussed and listed below:

- In future studies, transfer learning can be used to make the framework more generalized to the datasets from varied water pipelines.

- Besides, the superiority of the proposed LGAE-based method can also be employed to perform weakly supervised anomaly localization using soft proposal networks on the dataset.

- Furthermore, multi-class anomaly classification on the identified key frames can be investigated to facilitate the automated CCTV data analysis for water pipeline inspection.
Bibliography


