

**LEVERAGING BIG DATA AND DEEP LEARNING FOR ECONOMICAL
CONDITION ASSESSMENT OF WASTEWATER PIPELINES**

by

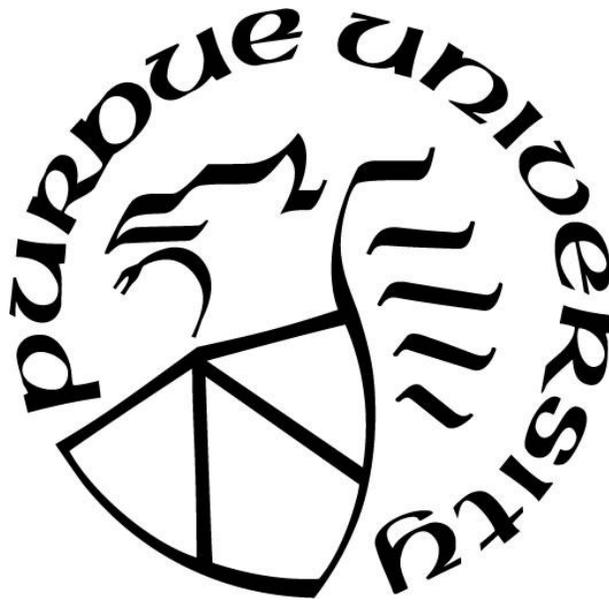
Srinath Shiv Kumar

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy



Lyles School of Civil Engineering

West Lafayette, Indiana

May 2020

**THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL**

Dr. Dulcy Abraham, Chair

Lyles School of Civil Engineering

Dr. Mohammad Jahanshahi

Lyles School of Civil Engineering

Prof. Edward Delp

School of Electrical and Computer Engineering

Prof. Tom Iseley

Trenchless Technology Center, Louisiana Tech University

Approved by:

Dr. Dulcy Abraham

Dedicated to my mother (Anuradha Kumar), father (Speedy Kumar), brother (Sohm Shiv Kumar), and dog (Marley).

ACKNOWLEDGMENTS

This work would not have been possible without the support and guidance from a network of brilliant people. First and foremost, I would like to thank my advisor Professor Dulcy Abraham for helping me grow as a researcher and as a human being. I consider myself extremely lucky to have been able to work with you and sincerely hope that someday you will write a book about mentoring PhD students (*after updating your personal website of course*).

I express my sincerest gratitude to my committee members Professor Tom Iseley, Professor Mohammad Jahanshahi, and Professor Edward Delp—you are my role models and I wish to emulate your success and groundedness.

I would like to thank Sean O'Rourke (Hazen and Sawyer), Justin Starr (formerly at RedZone Robotics), Eric Steinman (City of Fort Wayne Indiana), Jennifer Franz (Taylors Fire and Sewer District), Jeff Graham (Hydromax USA), and Matthew Rosenthal (SewerAI) for providing me with valuable data and insights, without which this work would not have been possible.

I owe a big thank you to my friends at Purdue University for distracting me from my studies just the right amount: Sharon Li, Troy Seberson, Shivam Gupta, Sreyansh Agarwal, Juyeong Choi, Saad Aljadhai, Mohamed Yamany, Hamed Zamenian, Jiannan Cai, Xin Xu, Seyedali Gahari, Tarutal Ghosh, Yunlan Zhang, Kevin Ro, Taka Yabe, Adwait Trikanad, Arnesh Daniel, Wonho Lee, and Prieston Lobo.

Finally, I would like to acknowledge the support of the Water Research Foundation (WRF) for funding my studies. This dissertation is based upon work supported by project WRF 4902. The contents of this dissertation reflect the views of the author, who is responsible for the facts and accuracy of the data presented herein. The content does not necessarily reflect the official views or policies of WRF at the time of publication.

TABLE OF CONTENTS

LIST OF TABLES	8
LIST OF FIGURES	9
ABSTRACT	12
CHAPTER 1. INTRODUCTION	15
1.1 Limitations of Manual CCTV Inspections.....	15
1.2 Limitations of Sewer Deterioration Modeling	17
1.3 Research Objectives.....	18
1.4 Anticipated Contributions of this Study	21
1.5 Chapter Organization	23
CHAPTER 2. BACKGROUND ON WASTEWATER PIPELINE INSPECTION AND ASSESSMENT.....	26
2.1 Related Studies on Automated Interpretation of CCTV Videos.....	28
2.1.1 Feature Engineering Approaches.....	29
2.1.2 Limitations of Feature Engineering Based Approaches	30
2.2 Deep Learning Approaches.....	31
CHAPTER 3. CNN-BASED AUTOMATED DEFECT CLASSIFICATION	33
3.1 Proposed Framework	34
3.2 Architecture of the CNN.....	35
3.2.1 Convolution	37
3.2.2 Activation Functions (ReLU and ELU).....	39
3.2.3 Max Pooling.....	39
3.2.4 Fully-Connected Layers.....	40
3.2.5 Output Layer	42
3.2.6 Mini-batch Gradient Descent.....	42
3.3 Techniques to Prevent Overfitting	43
3.3.1 Data Augmentation	43
3.3.2 Dropout	44
3.4 Experimental Results and Discussion.....	44
3.4.1 Preparation of Training, Validation and Testing Data Sets.....	44

3.4.2	Classification Accuracy, Precision, and Recall	47
3.5	Chapter Summary	51
CHAPTER 4.	CNN-BASED AUTOMATED DEFECT DETECTION	54
4.1	Evaluation of CNN-Based Object Detection Models	56
4.1.1	Single Shot Multibox Detector (SSD)	58
4.1.2	YOLOv3	59
4.1.3	Faster R-CNN	61
4.2	Experimental Results and Discussion.....	62
4.2.1	Preparation of Training and Evaluation Data Sets	63
4.2.2	Training and Evaluation of the Defect Detection Models	65
4.2.3	Performance of the Models.....	67
4.2.4	Discussion of Experimental Results	70
4.3	Demonstration Example.....	71
4.4	Limitations	73
CHAPTER 5.	TWO-STEP DEFECT DETECTION FRAMEWORK	74
5.1	Description of the Framework	75
5.2	Development of Anomaly Identification System	76
5.3	Results of Testing on CCTV Videos	77
5.4	Development of Supplementary Tool for Training Image Preparation	81
5.4.1	Description of ImgXtract.....	83
CHAPTER 6.	CNN INTERPRETATION TECHNIQUES	86
6.1	CNN Feature Visualization.....	87
6.2	Experiments and Discussions	88
6.3	Chapter Summary	94
CHAPTER 7.	VISION-BASED ORIENTATION ESTIMATION OF CCTV ROBOTS TO SUPPORT DEFECT LOCALIZATION AND AUTONOMOUS NAVIGATION.....	96
7.1	Related Studies.....	96
7.1.1	Automated Defect Interpretation	96
7.1.2	Autonomous Navigation of Sewer Robots	98
7.1.3	Contributions of this Study	100
7.2	Methodology	100

7.2.1	Image Frame Categorization.....	102
7.2.2	Vanishing Point Detection Module	105
7.2.3	Optical Flow Calculation Module	108
7.3	Experiments and Discussions	113
7.4	Conclusions.....	117
CHAPTER 8. A FRAMEWORK FOR MINING SPATIAL CHARACTERISTICS OF SEWER DEFECTS FROM INSPECTION DATABASES		120
8.1	Related Studies on Sewer Deterioration Modeling.....	121
8.2	Methodology	124
8.2.1	Defect Cluster Analysis (DCA).....	124
8.2.2	Defect Co-Occurrence Mining	130
8.3	Experiments and Discussions	132
8.3.1	Identification of Defect Clusters.....	133
8.3.2	Defect Co-Occurrence Mining	136
8.3.3	Validation of the Approach	138
8.3.4	Conclusions and Chapter Summary.....	141
CHAPTER 9. CONCLUSIONS AND RECOMMENDATIONS		143
9.1	Summary of the Research	143
9.2	Contributions to the Body of Knowledge	151
9.3	Contributions to the Body of Practice.....	153
9.4	Recommendations for Future Research	156
9.4.1	Autonomous UAVs for Rapid Inspection and Monitoring of Sewers.....	156
9.4.2	Data-driven Prediction of Water and Sewer Pipeline Failures	157
9.4.3	Smart, Secure, and Open Source Pipeline Asset Information Model.....	157
REFERENCES		158
APPENDIX A. ASCE REPRINT PERMISSION (JOURNAL OF COMPUTING IN CIVIL ENGINEERING)		166
APPENDIX B. ASCE REPRINT PERMISSION (INTERNATIONAL CONFERENCE ON COMPUTING IN CIVIL ENGINEERING)		168
APPENDIX C. ASCE REPRINT PERMISSION (CONSTRUCTION RESEARCH CONGRESS)		169

LIST OF TABLES

Table 2.1 Operational defects as defined by the NASSCO (2018) PACP.....	27
Table 2.2 Structural defects as defined by the NASSCO (2018) PACP.....	28
Table 3.1 Number of images in each category used for training, validation, and testing.....	47
Table 3.2 Number of positive and negative samples used.....	47
Table 3.3 Total number of parameters optimized during training.....	48
Table 3.4 Confusion matrix of testing sets (averaged across 5-folds).....	50
Table 4.1 Number of Images used for training and validation.....	65
Table 4.2 Summary of defect detection average precision values for the different IOU thresholds on the testing set.....	68
Table 5.1 Description of the dataset used for training and testing the automated anomaly identification system.....	77
Table 5.2 Description of the video files used for testing the automated system.....	78
Table 5.3 Defect detection results.....	79
Table 6.1 Number of images in each category used for training, validation, and testing.....	90
Table 6.2 Data augmentation options considered.....	91
Table 6.3 Data augmentation options considered.....	94
Table 7.1 Confusion matrix for camera orientation estimation.....	115
Table 8.1 Example PACP spreadsheet table that the Defect Cluster Identification Algorithm takes an input.....	128
Table 8.2 Execution steps of the cluster finding algorithm.....	130
Table 8.3 Defect clusters containing structural defects with ($S = 0.1\text{m}$ (0.3ft)).....	134
Table 8.4 Defect clusters containing structural defects with ($S = 0.3\text{m}$ (1ft)).....	134
Table 8.5 Defect clusters containing structural defects with ($S = 0.9\text{m}$ (3ft)).....	134
Table 8.6 Association rules with confidence greater than 0.5 and lift greater than 1.....	137
Table 8.7 Validation scores based on assessment by SMEs.....	139
Table 9.1 Summary of methodologies, datasets, findings, and limitations.....	148

LIST OF FIGURES

Figure 1.1 Overview of the CCTV inspection technique	16
Figure 1.2 Illustration to signify the importance of considering defect locations along a pipe....	18
Figure 1.3 Development of proposed automated CCTV inspection video interpretation system	19
Figure 1.4 Development of algorithms to facilitate autonomous navigation of sewer inspection robots.....	20
Figure 1.5 Mining large datasets of wastewater pipeline condition information	21
Figure 3.1 Description of the images used for training and testing the CNN classifiers.....	34
Figure 3.2 Overall training and classification pipeline.....	35
Figure 3.3 Architecture of the CNN for wastewater pipeline feature classification.....	37
Figure 3.4 Example of a convolution operation on an RGB image.....	38
Figure 3.5 Max pooling using a 2×2 window and a stride size of one	40
Figure 3.6 Example of two fully-connected (FC) layers with three neurons in each layer	41
Figure 3.7 Sample data augmentation through changes in brightness, contrast, and motion blur	43
Figure 3.8 Sample images used for training and testing.....	45
Figure 3.9 Training, validation, and testing accuracies on the 5 folds of root intrusion, deposit, and crack defect images.....	49
Figure 3.10 Examples of similarities in appearance between different categories of images	51
Figure 4.1 a) Example to demonstrate defect classification, b) Example to demonstrate defect detection.....	55
Figure 4.2 Illustration of intersection over union	58
Figure 4.3 Conceptual architecture of the SSD	59
Figure 4.4 An example of how YOLOv3 detects multiple objects in an image.....	61
Figure 4.5 Faster R-CNN conceptual architecture (Wang and Cheng, 2018)	62
Figure 4.6 Examples of images used for training and testing (Kumar et al. 2018)	64
Figure 4.7 Peak mAP curve used to find optimal training iterations for the three models.....	67
Figure 4.8 Comparison of the bounding box detections by the SSD (a – f), YOLOv3 (g – l), and Faster R-CNN (m – r) models.....	69
Figure 4.9 Example showing multiple ways to annotating a defect that spans the circumference of the pipe.....	71

Figure 4.10 Framework for producing defect detections on an input CCTV video	72
Figure 4.11 Example images showing the defects detected by the automated defect detection tool applied to a CCTV video in real-time.....	73
Figure 5.1 Illustration of the two-stage detection framework.....	76
Figure 5.2 Examples of correct detections by the system.....	79
Figure 5.3 Examples of incorrect detections by the system.....	79
Figure 5.4 Schematic diagram of the iterative procedure to aggregate incorrect detections	80
Figure 5.5 a) Original image with label; b) Image after addition of the mask to cover the label in the original image.	82
Figure 5.6 Schematic diagram of ImgXtract.....	84
Figure 5.7 a) ImgXtract displays a grid of nine images to the user; b) The tool automatically stores images in folders based on the defect type.	85
Figure 6.1 CAM applied to an example CCTV image.	88
Figure 6.2 Examples of images used for training and testing.....	89
Figure 6.3 CAM outputs for model 3 on validation images.	92
Figure 6.4 Example CAM of model 3 with and without the incorporation of rotations as a data augmentation technique	93
Figure 7.1 Example to illustrate the attributes of defects that are identified during sewer CCTV inspections.....	97
Figure 7.2 Example to illustrate how camera orientation affects the appearance of images.....	98
Figure 7.3 Illustration of sewer pipe vanishing point	101
Figure 7.4 Five camera orientations addressed by the methodology	104
Figure 7.5 Architecture of the SSD MobileNets model used for vanishing point detection	105
Figure 7.6 Example bounding box coordinates output by the SSD.....	107
Figure 7.7 Example to illustrate how the vanishing point center varies with camera orientation	108
Figure 7.8 Illustration of optical flow vectors computed between two image frames.....	109
Figure 7.9 Optical flow vectors corresponding to forward motion and right turn of a CCTV camera	111
Figure 7.10 Example optical flow vectors computed on image frames that depict a right turn .	112
Figure 7.11 Conceptual overview of experimental setup for accuracy evaluation.....	114
Figure 7.12 Images to illustrate errors originating from incorrect vanishing point detections...	116

Figure 7.13 Example image to demonstrate camera rotations about forward axis	117
Figure 8.1 Illustration to signify the importance of considering defect locations along a pipe..	121
Figure 8.2 Illustration of a pipe segment with seven defects.....	125
Figure 8.3 Illustration of three example defect clusters.....	126
Figure 8.4 Execution time of the algorithm for inspection spreadsheets.....	128
Figure 8.5 Number of pipe segments by: a) materials and b) diameters	132
Figure 8.6 Number of instances of structural defects	132
Figure 8.7 Visualization of the five clusters with the highest cluster severity score.....	135
Figure 8.8 Example cluster visualizations generated by the web-tool.....	136
Figure 9.1 Framework for defect interpretation—assigning minor defects to computer processes and severe defects for human interpretation	155

ABSTRACT

Sewer pipelines are an essential component of wastewater infrastructure and serve as the primary means for transporting wastewater to treatment plants. In the face of increasing demands and declining budgets, municipalities across the US face unprecedented challenges in maintaining current service levels of the 800,000 miles of public sewer pipes. Inadequate maintenance of sewer pipes leads to inflow and infiltration, sanitary sewer overflows, and sinkholes, which threaten human health and are expensive to correct. Accurate condition information from sewers is essential for planning maintenance, repair, and rehabilitation activities and ensuring the longevity of sewer systems. Currently, this information is obtained through visual closed-circuit television (CCTV) inspections and deterioration modeling of sewer pipelines. CCTV inspection facilitates the identification of defects in pipe walls whereas deterioration modeling estimates the remaining service life of pipes based on their current condition. However, both methods have drawbacks that limit their effective usage for sewer condition assessment. For instance, CCTV inspections tend to be labor intensive, costly, and time consuming, with the accuracy of collected data depending on the operator's experience and skill level. Current deterioration modeling approaches are unable to incorporate spatial information about pipe deterioration, such as the relative locations, densities, and clustering of defects, which play a crucial role in pipe failure. This study attempts to leverage recent advances in deep learning and data mining to address these limitations of CCTV inspection and deterioration modeling and consists of three objectives.

The first objective of this study seeks to develop algorithms for automated defect interpretation, to improve the speed and consistency of sewer CCTV inspections. The development, calibration, and testing of the algorithms in this study followed an iterative approach that began with the development of a defect classification system using a 5-layer convolutional neural network (CNN) and evolved into a two-step defect classification and localization framework, which combines a the ResNet34 CNN and Faster R-CNN object detection model. This study also demonstrates the use of a feature visualization technique, called class activation mapping (CAM), as a diagnostic tool to improve the accuracy of CNNs in defect classification tasks—thereby representing a crucial first step in using CNN interpretation techniques to develop improved models for sewer defect identification.

Extending upon the development of automated defect interpretation algorithms, the second objective of this study attempts to facilitate autonomous navigation of sewer CCTV robots. To overcome Global Positioning System (GPS) signal unavailability inside underground pipes, this study developed a vision-based algorithm that combines deep learning-based object detection with optical flow for estimating the orientation of sewer CCTV cameras. This algorithm can enable inspection robots to estimate their trajectories and make corrective actions while autonomously traversing pipes. Hence, considered together, the first two objectives of this study pave the way for future inspection technologies that combine automated defect interpretation with autonomous navigation of sewer CCTV robots.

The third and final objective of this study seeks to develop a novel methodology that incorporates spatial information about defects (such as their locations, densities, and co-occurrence characteristics) when assessing sewer deterioration. A methodology called Defect Cluster Analysis (DCA) was developed in order to mine sewer inspection reports and identify pipe segments that contain clusters of defects (i.e., multiple defects in proximity). Additionally, an approach to mine co-occurrence characteristics among defects is also introduced (i.e., identification of defects which occur frequently together). Together the two approaches (i.e., DCA and co-occurrence mining) address a key limitation of existing deterioration modeling approaches (i.e., the lack of consideration to spatial information about defects)—thereby leading to the generation of new insights into pipeline rehabilitation decision-making.

The algorithms and approaches presented in this dissertation have the potential to improve the speed, accuracy, and consistency of assessing sewer pipeline deterioration, leading to better prioritization strategies for maintenance, repair, and rehabilitation. The automated defect interpretation algorithms proposed in this study can be used to assign the subjective and error-prone task of defect identification to computer processes, thereby enabling human operators to focus on decision-making aspects, such as deciding whether to repair or rehabilitate a pipe. Automated interpretation of sewer CCTV videos could also facilitate re-evaluation of historical sewer inspection videos, which would be infeasible if performed manually. The information gleaned from re-evaluating these videos could generate insights into pipe deterioration, leading to

improved deterioration models. The algorithms for autonomous navigation could enable the development of completely autonomous inspection platforms that utilize unmanned aerial vehicles (UAVs) or similar technologies to facilitate rapid assessment of sewers. Furthermore, these technologies could be integrated into wireless sensor networks, paving the way for real-time condition monitoring of sewer infrastructure. The DCA approach could be used as a diagnostic tool to identify specific sections in a pipeline system that have a high propensity for failure due to the existence of multiple defects in proximity. When combined with contextual information (e.g., soil properties, water table levels, and presence of large trees), DCA could provide insights about the likelihood of void formation due to sand infiltration. The DCA approach could also be used to periodically determine how the distribution of defects and their clustering progresses with time and when examined alongside contextual data (e.g., soil properties, water table levels, presence of trees) could reveal trends in pipeline deterioration.

CHAPTER 1. INTRODUCTION

Wastewater infrastructure represents a significant investment in physical assets, with over 800,000 miles of public sewage pipes and 500,000 miles of private wastewater pipeline laterals in the US alone (ASCE 2017). While municipalities have invested heavily in wastewater pipeline system expansion, they have allocated a relatively smaller proportion of the budget to wastewater pipeline rehabilitation (AWWA 2012). As a result, many municipalities across the US face the problem of aging wastewater infrastructure in dire need of repair, rehabilitation or renewal. The lack of funding towards wastewater infrastructure rehabilitation is highlighted in the Clean Watersheds Needs Survey, which estimated the wastewater and stormwater treatment and collection requirements for the US at \$271 billion, as of January 1, 2012 (EPA 2012).

Sanitary sewer pipes are an essential component of wastewater infrastructure and serve as the primary means for transporting wastewater to treatment plants (Baah et al. 2015). Inadequate maintenance of sewer pipes leads to issues such as inflow and infiltration, sanitary sewer overflows, and sinkholes, which threaten human health and also tend to be very expensive to correct. For instance, the cost to repair the 2016 Fraser sinkhole in Michigan was estimated at over US \$78 million. Inflow and infiltration, which are often caused by cracks in wastewater pipeline walls, root intrusions, and leaking manholes, cost municipalities an additional treatment cost of \$2 to \$5 per thousand gallons of sewage (EPA 2014). The EPA also estimates between 23,000 and 75,000 sanitary sewer overflows each year in the US, which release large quantities of untreated sewage into the environment, exposing humans to a variety of illnesses (EPA 2016). Accurate condition information about sewer pipelines is essential to plan maintenance, renewal and rehabilitation activities. Currently, this information is obtained through visual closed-circuit television (CCTV) inspections and deterioration modeling. However, both methods have serious drawbacks limiting their usage for sewer condition assessment.

1.1 Limitations of Manual CCTV Inspections

Over the past 40 years, municipalities in North America have used closed-circuit television (CCTV) as the primary technique for inspecting non-man-entry wastewater pipes (see Figure 1.1). CCTV

inspection involves recording a video of the inner surface of a pipe using a camera equipped crawler. Trained inspectors review the recorded CCTV videos either in real-time (i.e., while navigating the camera crawler) or offline (i.e., after the inspection has been completed), and manually identify defects (e.g., roots, deposits, cracks, etc.). However, this manual process of reviewing CCTV sewer inspection videos relies on a subjective evaluation of defects and has the propensity to be error prone and inconsistent (Harvey and McBean 2014).

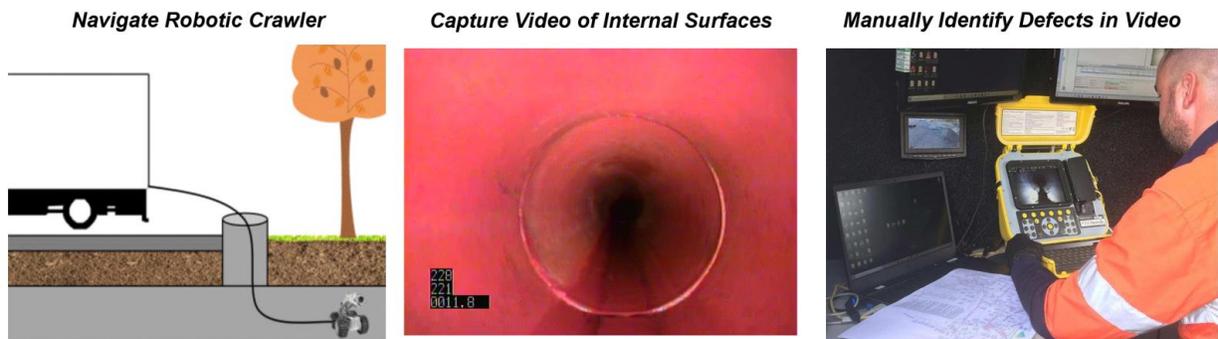


Figure 1.1 Overview of the CCTV inspection technique

Dirksen et al. (2013) showed CCTV images of sewer defects from 60 different sewer inspections to six trained operators in order to compare the operators' interpretations. Their study found that the operators' interpretations disagreed on approximately 40 percent of the images, highlighting the subjective nature of CCTV inspections. Based on data from 45,049 sewer inspections in the city of Braunschweig, Germany, Caradot et al. (2018) calculated that operators underestimated and overestimated the severity of 15 percent and 20 percent of the defects, respectively. Manual review of the inspection videos also tends to be slow and labor intensive since the operators must carefully inspect the videos to identify potential defects in the pipe. In order to ensure consistency in defect reporting, CCTV reports are frequently reviewed and audited offsite, leading to low overall inspection rates and higher inspection costs. According to an EPA study, US municipalities on average spend \$0.84 per linear foot for sewer CCTV inspections (Feeney et al. 2010). Due to the high inspection costs, municipalities typically inspect only a small fraction of their networks, owing to budgetary limitations. A survey of 75 US municipalities by the National Association of Clean Water Agencies found that about half of the municipalities inspect less than 10% of their

system each year. (AMSA 2003). Hence, there is a need for technologies to improve the defect interpretation consistency and speed of sewer CCTV inspections.

1.2 Limitations of Sewer Deterioration Modeling

CCTV inspections provide a snapshot about the condition of a pipe on the date of inspection. Deterioration modeling aims to predict the future condition of pipes which is essential for developing long-term maintenance, renewal, and rehabilitation plans. However, a major limitation of current sewer deterioration modeling approaches (Chapter 8 provides a more in-depth discussion about the limitations of existing deterioration modeling approaches) is a lack of consideration for spatial information about defects in a pipe. For instance, upon completion of CCTV inspections, the identified defects in pipes are assigned numerical grades (typically between 1 and 5) to denote their severity. The grades of defects in a pipe are then aggregated (i.e., typically by calculating the sum of individual defect grades) into a single condition grade to represent the amount of deterioration of a pipe. The aggregated condition grade is then used as basis to estimate a pipe's likelihood of failure, remaining service life, and to determine the optimal choice and timing of maintenance or repairs. However, the aggregation of defect grades into a single condition grade leads to a loss of spatial information, i.e., information about the density, severity, and co-occurrence characteristics of defects—information which can play a crucial role in calculating a pipe's likelihood of failure. For instance, the approach of using a single aggregated condition grade overlooks the increased likelihood of failure of a pipe with defect clusters (i.e., areas with multiple defects in proximity). Figure 1.2 illustrates this problem. Under the conventional method of assigning a single grade to pipes, the pipes in Figure 1.2a and Figure 1.2b would both be assigned identical condition grades and hence be deemed to be equally prone to failure.

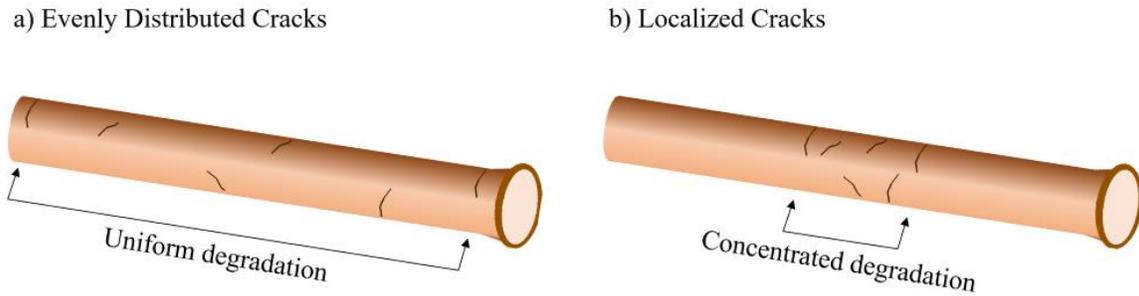


Figure 1.2 Illustration to signify the importance of considering defect locations along a pipe

However, it could be argued that the pipe in Figure 1.2b has a higher likelihood of failure for the following reasons: (1) defects, which are close to each other could propagate and coalesce into more severe defects; (2) multiple cracks and fractures may lead to soil infiltration leading to the formation of voids over the pipe. Voids over pipes are known to result in sinkholes; and (3) multiple defects in proximity can lead to a localized region of weakness, resulting in an increased likelihood of collapse. Hence, existing deterioration modeling approaches, which rely on aggregated condition scores, do not account for the relationships between defect locations and likelihood of failure. Thus, there is a need to develop techniques that also consider spatial information about defects when assessing pipe condition.

1.3 Research Objectives

This study aims to leverage recent advances in deep learning and data mining to address the challenges mentioned in Section 1.1 and Section 1.2 and consists of three research objectives. The first two objectives address the challenge of improving the defect interpretation consistency and speed of sewer CCTV inspections, whereas the third objective seeks to address the lack of consideration about spatial information of defects. The three objectives are as follows:

Objective 1 – Development of an Automated Interpretation System for Sewer CCTV Inspections

The primary focus of this study is on developing an automated interpretation system for sewer CCTV inspections, specifically, the development of algorithms for automated identification of

defects in CCTV videos (see Figure 1.3). The accuracy and speed of the defect identification algorithms will be evaluated in the detection of structural defects (i.e., cracks, fractures, and deformations), operational defects (i.e., root intrusions and deposits), and construction features (i.e., lateral connections), since these categories represent the majority of sewer defects and features. Additionally, the algorithms will be calibrated with emphasis on 8-inch, 10-inch, and 12-inch diameter vitrified clay pipes (VCPs), since these pipes comprise a majority of the sewer mains in the US (Tafari and Selvakumar 2002). However, the approaches described in this study can be extended to pipes of other materials and diameters.

The development, calibration, and testing, of the algorithms in this study follows an iterative approach, beginning with algorithms for defect classification and concluding with a two-step defect detection framework that incorporates neural network interpretation techniques. Chapters 2, 3, 4, 5, and 6 describe the various stages involved in the development and testing of the algorithms.

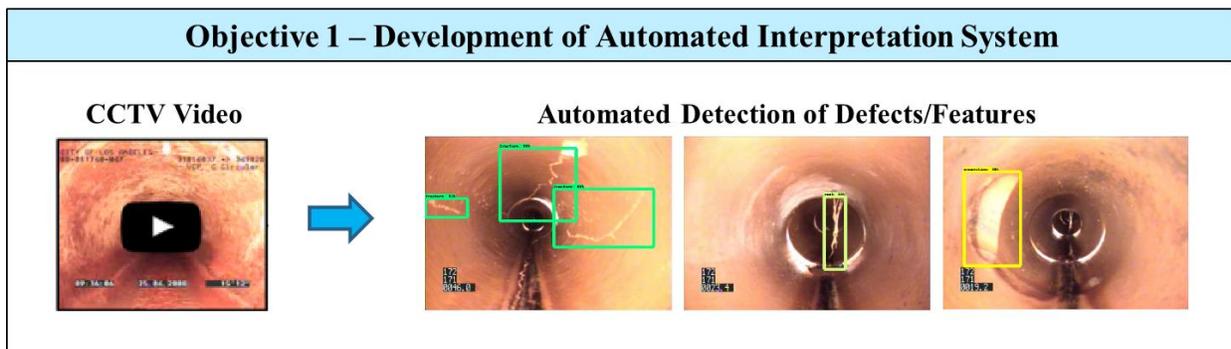


Figure 1.3 Development of proposed automated CCTV inspection video interpretation system

Objective 2 – Development of Algorithms to Facilitate Autonomous Navigation of Sewer Inspection Robots

Extending the development of the automated interpretation system, the second objective of this study focuses on the development of algorithms that can facilitate autonomous navigation of CCTV inspection robots. Since underground pipelines are GPS-denied environments, this study aims to develop vision-based techniques for localizing the position of sewer CCTV inspection robots in pipes—in order to facilitate ‘self-driving’ inspection robots (see Figure 1.3). The algorithms will be calibrated for sewer CCTV robots operating in 8-inch, 10-inch, and 12-inch

diameter VCPs because these pipes represent a majority of sewer mains in the US. However, the algorithms can be extended for other inspection platforms, such as indoor unmanned aerial vehicles (UAVs), and for pipes of other materials and diameters. Autonomous navigation in sewers could facilitate significantly faster inspections and lead to reductions in labor requirements. Chapter 7 describes the development and evaluation of the autonomous navigation algorithms.

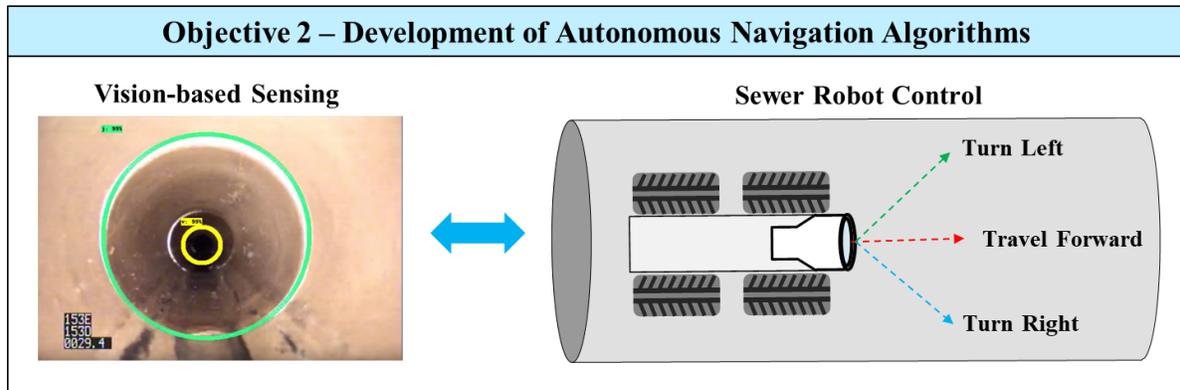


Figure 1.4 Development of algorithms to facilitate autonomous navigation of sewer inspection robots

Hence, considered together, the first two objectives of this study lay the foundation for the development of completely autonomous sewer pipeline inspection systems, that could significantly improve the speed, accuracy, and consistency of current condition assessment efforts. We envision that the future of sewer inspections would involve a network of robots that continuously and autonomously monitor the condition of pipes and relay the information to engineers, via wireless sensor networks, who make repair, rehabilitation, or replacement decisions. Although these research objectives encompass wastewater pipeline infrastructure, the approaches are adaptable to other horizontal infrastructure domains (such as water, oil, and gas pipelines) as well.

Objective 3 – Mining Spatial Characteristics of Defects from Sewer Inspection Records

Given the large extent of wastewater pipeline infrastructure and the burgeoning of rapid inspection technologies, the volume and velocity of pipe condition data is likely to grow at an unprecedented pace. Moreover, many municipalities have begun sharing information about their pipeline assets

through initiatives such as Data.gov (<https://www.data.gov>), thereby ushering in the creation of large datasets of publicly accessible sewer pipeline condition information. Hence, the third objective of this study seeks to develop techniques to mine these large datasets and discover insights that can be used to guide maintenance prioritization efforts.

Specifically, this study seeks to develop a novel methodology for assessing sewer deterioration by incorporating spatial information, such as the locations, densities, and co-occurrence characteristics of defects in pipes (see Figure 1.4). A methodology called Defect Cluster Analysis (DCA) was developed in order to mine sewer inspection reports and identify pipe segments which contain defect clusters. Additionally, an approach to mine co-occurrence characteristics among defects is also introduced (i.e., identification of defects which occur frequently in pairs). Together the two approaches (i.e., DCA and co-occurrence mining) address the limitations of existing deterioration modeling approaches (i.e., the lack of consideration to spatial information about defects) thereby leading to new insights in pipe asset management and rehabilitation decision-making. Chapter 8 describes the development and evaluation of the DCA and co-occurrence mining approaches.

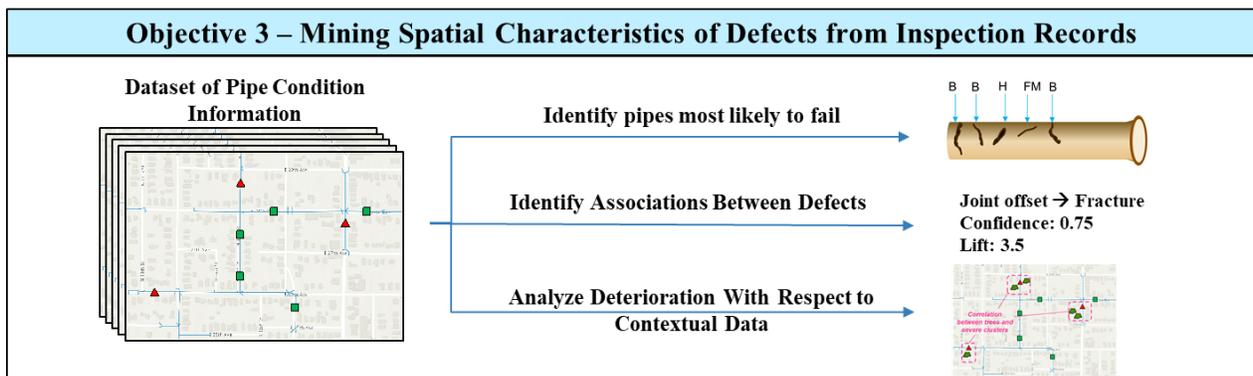


Figure 1.5 Mining large datasets of wastewater pipeline condition information

1.4 Anticipated Contributions of this Study

The algorithms for automated interpretation of CCTV videos, which are an outcome of Objective 1, could facilitate consistent, accurate, and quick condition assessment of sewers by minimizing inspection errors due to fatigue, biases, and differing skill levels. The repetitive and error-prone task of identifying defects in videos could be assigned to computer processes, thereby enabling

inspectors to shift their focus on decision-making aspects, such as deciding to repair or rehabilitate a pipe. Automated interpretation of sewer CCTV videos could also facilitate re-evaluation of historical sewer inspection videos, which is a task that would be infeasible if performed manually. The information obtained from re-evaluating these videos could be used to develop statistical models for predicting sewer deterioration. For instance, the rate of deterioration of a pipe can be calculated by comparing multiple inspection videos of the same pipe, which were recorded at different instances of time. Automated defect detection algorithms can be incorporated into existing CCTV inspection reporting software, facilitating adoption by municipalities and condition assessment contractors. Moreover, these algorithms can also be extended to facilitate automated defect detection in water pipelines. Furthermore, this study aims to leverage deep learning to develop algorithms and approaches that improve upon the accuracy and speed of current sewer defect identification approaches.

The algorithms for autonomous navigation, which are an outcome of Objective 2, could facilitate the development of completely autonomous inspection platforms for sewers. For instance, these algorithms could be integrated onboard unmanned aerial vehicles (UAVs) or similar technologies, to facilitate rapid inspection of sewers. Constrained by the speed at which the human eye can interpret videos, most sewer CCTV inspections are limited to a maximum speed of 1 foot/second. Inspection platforms that combine autonomous navigation with automated defect interpretation could facilitate inspections at significantly higher speeds, enabling a larger percentage of pipes to be inspected each year.

The DCA approach developed as part of Objective 3, could provide asset managers with a tool to consider defects and their proximity in a way that has not been previously possible. DCA could be used as a diagnostic tool to identify sections of the pipe that are likely to fail or have the propensity to progress into severe defects. When combined with contextual information, DCA could provide additional insights, such as the likelihood of void formation due to sand infiltration. The DCA approach could be used to periodically determine how the distribution of defects and their clustering progresses with time and could reveal pipeline deterioration patterns. The identification of defect clusters could also inform the choice of rehabilitation option. For instance, the identification of defect clusters could lead to insights into whether whole length rehabilitation or

local patch repairs should be pursued. Furthermore, the mining of co-occurrence characteristics among sewer defects could highlight groups of defects which occur frequently together, allowing for the creation of customized maintenance plans. For instance, the identification of pipes which contain high water marks and fractured walls could indicate a high propensity for sewage exfiltration during sewer surges. Additionally, the identification of successive exfiltration and infiltration locations along a pipe could indicate the presence of external water flows that have the potential to compromise the pipe bedding, leading to sewer collapses.

1.5 Chapter Organization

This dissertation is divided into nine chapters and follows the ‘multiple publications’ format. Each chapter is self-contained and has its own introduction, review of related studies, methodology, analysis, and conclusion sections. Significant portions of these chapters have been published or submitted for review and publication in peer reviewed journals and/or refereed conferences. Chapter 1 discusses the motivations and objectives of this study and provides an overview of the methodological approach.

Chapter 2 provides an overview of sewer CCTV inspections and reviews existing studies on automated interpretation of sewer CCTV inspections. This chapter discusses the limitations of feature engineering approaches and explains the rationale behind utilizing deep learning approaches for automated defect interpretation.

Chapter 3 describes a framework that uses an ensemble of deep convolutional neural networks (CNNs) to classify root intrusions, deposits, and cracks in CCTV images. The framework also enables the classification of multiple instances of defects in images. This chapter is reprinted in part from Kumar, S. S., Abraham, D. M., Jahanshahi, M. R., Iseley, T., and Starr, J. (2018). Automated Defect Classification in Sewer Closed Circuit Television Inspections using Deep Convolutional Neural Networks. *Automation in Construction*, 91, 273-283. Permission to reprint not required by Elsevier. Tables and figure captions have been modified to maintain the form of the dissertation.

Chapter 4 extends the automated system proposed in Chapter 3, to facilitate detection (i.e., classification and localization) of defects. To this end, the three state-of-the-art CNN models—single-shot detector (SSD), you only look once (YOLO), and faster region-based convolutional neural network (Faster R-CNN)—are evaluated for speed and precision in detecting sewer defects. This chapter is reprinted in part from Kumar, S. S., Wang, M., Abraham, D. M., Jahanshahi, M. R., Iseley, T., and Cheng, J. C. (2020). Deep Learning–Based Automated Detection of Sewer Defects in CCTV Videos. *Journal of Computing in Civil Engineering*, 34(1), 04019047. With Permission from ASCE (see Appendix A). Tables and figure captions have been modified to maintain the form of the dissertation.

Chapter 5 describes a two-step CNN framework, which uses a pre-processing step to determine whether images contain defects or not and applies a defect detection algorithm only to those images that contain defects. The pre-processing step helps avoid unnecessary computations enabling faster and more accurate detection of sewer defects. This chapter is reprinted in part from Kumar, S. S. and Abraham, D. M. (2019). A Deep Learning Based Automated Structural Defect Detection System for Wastewater Pipelines, ASCE International Conference on Computing in Civil Engineering (i3CE 2019), Atlanta, Georgia, USA. With Permission from ASCE (see Appendix B). Tables and figure captions have been modified to maintain the form of the dissertation.

Chapter 6 discusses how class activation mapping (CAM) can be used for interpreting CNNs and how the insights from CAM can lead to the development of a more generalizable automated defect detection system. This chapter is reprinted in part from Kumar, S. S. and Abraham, D. M. (2020). Leveraging Visualization Techniques to Develop Improved Deep Neural Network Architectures for Sewer Defect Identification, ASCE Construction Research Congress (CRC 2020), Tempe, Arizona, USA. With Permission from ASCE (see Appendix C). Tables and figure captions have been modified to maintain the form of the dissertation.

Chapter 7 describes the development of a vision-based algorithm to determine the position of inspection robots in sewer pipelines. The algorithm proposed in this chapter serves two purposes: (1) facilitating autonomous navigation of sewer robots in pipelines and (2) automated localization of defects (i.e., longitudinal and circumferential) in pipelines. The work presented in this chapter

will be prepared as a journal paper and submitted to the Automation in Construction Journal in May 2020.

Chapter 8 presents methods for mining spatial characteristics of defects from sewer pipeline inspection reports. Two approaches are presented in this chapter: (1) DCA, for identifying regions of high defect concentrations and (2) association rule mining of sewer defects to identify co-occurrence patterns among different categories of defects. This chapter is reprinted from a manuscript that is currently under preparation: Kumar, S. S., Abraham, D. M., Choi, J. (2020). A Framework for Mining Spatial Characteristics of Sewer Defects from Inspection Databases. Tables and figure captions have been modified to maintain the form of the dissertation.

Chapter 9 concludes the dissertation with a summary of the work, the contributions to the body of knowledge and practice, the limitations of the research, and recommendations for future research.

CHAPTER 2. BACKGROUND ON WASTEWATER PIPELINE INSPECTION AND ASSESSMENT

Wastewater pipeline CCTV inspections typically involve multiple steps. First, the access points of a pipeline are located, typically through a review of as-built records and/or aerial imagery. For the types of pipes addressed in this study (i.e., vitrified clay sewer pipes with diameters ranging between 8 inches and 12 inches), the access points are typically manholes. Once the access points are located, the operators lower a CCTV camera-equipped inspection crawler down the access point into the pipe. The crawlers are connected with coaxial cables that supply power to the electronics on-board the crawler and also transmit the video feed from the cameras back to the operator. Inspection of the pipe involves navigating the inspection crawler from one end of the pipe to the other end and back, while simultaneously monitoring the video-feed. When defects (e.g., cracks, fractures, roots, etc.) are encountered in the pipe, the operator typically stops the motion of the robot and codes the defect (usually with the help of inspection reporting software). Coding a defect involves classifying the defect using an accepted convention such as the North American Society of Sewer Service Companies (NASSCO) Pipeline Assessment and Certification Program (PACP®), which requires the operator to identify the defect's category (e.g., spiral crack, medium roots, large joint offset, hole, etc.), the defect's longitudinal location, i.e., its linear distance from the access point, and its circumferential location, i.e., its location relative to the cross section of the pipe. Additionally, the NASSCO PACP also requires operators to code any lateral connections to the pipe. Once the pipe has been traversed and all defects and laterals identified, the crawler is navigated in the reverse direction towards the access point of entry, where it can be retrieved from.

Defects are typically a function of ageing, environmental loading, quality of construction, inappropriate material usage, soil conditions, hydraulic properties, increased overburden loads, unanticipated user demands, or natural forces. The most common defects in wastewater pipeline systems are categorized as structural defects or operational defects according to the NASSCO PACP. Table 2.1 and Table 2.2 provide definitions of common operational and structural defects, respectively.

Table 2.1 Operational defects as defined by the NASSCO (2018) PACP

Category	Definition	Types
Deposits	This code is used to report a wide range of deposits in wastewater pipeline systems. Deposits can cause flow turbulence and partial blockages and may reduce the hydraulic capacity of the wastewater pipeline.	Attached deposits, settled deposits, ingressed deposits
Roots	This code is used to describe the ingress of roots through defects in the wastewater pipeline, pipe connections, or manholes.	Fine roots, tap roots, medium roots, ball roots
Infiltration	Infiltration is the ingress of groundwater through a defect or porous area of pipe wall.	Infiltration stain, infiltration weeper, infiltration dripper, infiltration runner, infiltration gusher
Obstacles	This code is used to record the presence of large and medium sized obstacles that are likely to cause a serious obstruction to the flow.	Brick or masonry, pipe material in invert, object protruding through wall, rocks

Table 2.2 Structural defects as defined by the NASSCO (2018) PACP

Category	Definition	Types
Crack	The crack code is used where a crack line is visible on a surface, but it is not visibly open, i.e., there is no gap between the edges of the crack.	Longitudinal crack, circumferential crack, multiple cracks, spiral cracks, hinge cracks
Fracture	A fracture is a crack which is visibly open although the sections of the wall are still in place and not able to move.	Longitudinal fracture, circumferential fracture, multiple fractures, spiral fractures, hinge fractures
Hole	This code is used when the pipe material is missing, and the surrounding soil exposed. This occurs where the pipe pieces have been completely dislodged from the pipe wall.	Hole soil visible, hole void visible
Deformed	This code is used when the pipe is damaged to a point that the original cross-section is noticeable altered. Deformation is the last stage of severity before collapse.	Deformation vertical, deformation horizontal
Collapse	A collapse is termed as a deformation where there is complete loss of structural integrity with greater than 40 percent of the cross-sectional area lost. Collapses prevent the camera from passing this defect and hence surveys stop once a collapse is encountered.	Collapse in pipe, collapse in brick wastewater pipeline
Joint Offsets	This defect code indicates a defective displacement at joints.	Joint offset, joint separated, joint angular

2.1 Related Studies on Automated Interpretation of CCTV Videos

An automated method for interpreting wastewater pipeline CCTV videos could assist inspectors in quickly identifying defects and provide a more accurate and consistent method of evaluating pipe condition. The repetitive and error-prone task of identifying defects in videos could be assigned to computer processes, thereby enabling inspectors to shift their focus on decision-making aspects, such as deciding to repair or rehabilitate a pipe.

Depending on the quality of video footage, an automated interpretation system could perform the following tasks:

1. Classify defects by category (i.e., crack, fracture, etc.)
2. Classify defects by type (i.e., circumferential crack, longitudinal crack, etc.)
3. Determine the longitudinal location of defects (i.e., distance from the point of entry of the camera)
4. Determine the circumferential location of defects (i.e., position relative to the cross-section)

Prior research on automated interpretation of wastewater pipeline CCTV videos focuses primarily on defect classification by category. That is given an image from a wastewater pipeline, algorithms are developed to categorize the type of defects that are visible in the images. Between 1997 and 2018 most research studies typically attempted to perform automated defect interpretation using feature engineering approaches. However, recent studies (i.e., 2018 onwards) have demonstrated improved defect detection accuracies and generalization capabilities when utilizing feature learning approaches (such as deep neural networks).

2.1.1 Feature Engineering Approaches

Early studies by Xu et al. (1998) and Moselhi and Shehab (2000) used image processing techniques such as edge detection, binary image thresholding, and Fourier transforms to extract features from wastewater pipeline CCTV images. Xu et al. (1998) used the extracted features to measure the cross-section profile of a pipe and identify deformations in plastic pipes, whereas Moselhi and Shehab (2000) used the extracted features to identify cracks, joint displacements, cross-section reduction, and spalling in 305mm (12-inch) diameter concrete and clay pipes. More recent studies used a two-step methodology which first segmented images into regions of interest and then extracted features from the regions of interest. This method of extracting features only from the regions of interest reduces the number of pixels that have to be analyzed, which reduces the processing time. Guo et al. (2009) used the concept of frame differencing to identify regions of interest, where candidate images are subtracted from a reference image of a healthy pipe section. The regions of interest are then inputted to feature extraction and classification modules to classify the type of defects. Guo et al. (2009) demonstrated their approach on approximately 300 images

extracted from 305mm (12 inches) diameter vitrified clay and concrete pipes located in Pittsburgh, Pennsylvania. Halfawy and Hengmeechai (2014a) and Halfawy and Hengmeechai (2014b) used morphological operations to identify regions of interest and histograms of oriented gradients (HOG) to extract features from the regions of interest. Their method used a support vector machine classifier to classify defects based on the extracted features. Their method was evaluated using images from concrete and vitrified clay pipes having diameters ranging from 250mm to 900mm, which were located in Regina and Calgary, Canada. Su and Yang (2014) used morphological segmentation based on edge detection to extract regions of interest and demonstrated on a test set consisting of images from vitrified clay pipes. This method had a higher accuracy than conventional morphological operations. Moradi and Zayed (2017) extracted spatio-temporal features from CCTV videos and trained a Hidden Markov Model to identify images that showed defects or anomalies in the pipe. They evaluated their method on images extracted from 610mm (24 inches) diameter concrete pipes located in Quebec, Canada. Hawari et al. (2018) proposed a method that used image processing techniques including thresholding, Gabor transformation, ellipse fitting, and segmentation to classify the defects contained in CCTV images of wastewater pipelines. Their method was tested on a sample of 32 images extracted from vitrified clay pipelines that were located in Qatar and had condition ratings ranging from very good to poor. Their method achieved precision scores, i.e., the ratio of true positives to the sum of true positives plus false positives of 73.7%, 52.9%, and 65.3% in classifying images of cracks, deposits, and displaced joints, respectively.

2.1.2 Limitations of Feature Engineering Based Approaches

The use of morphologies and feature extraction approaches limits the generalization capability of previously developed automated systems. Generalization capability of a classifier can be defined as the ability to classify images that exhibit significant variations (i.e., in terms of shape, color, texture, etc.) from the images used for training. For instance, morphological operations require structuring elements (e.g., simple shapes used to convolve images with) to be manually calibrated through repeated trials on training images. As a result, structuring elements which may work well with training images captured under specific conditions (e.g., focal length of camera, distance between object and camera, focal length of lens, illumination conditions, etc.) may not be optimal for images captured under different conditions. Furthermore, morphological approaches are

susceptible to generating false positives in wastewater pipeline that have variations in internal surface colors due to surface staining, relining of pipes, change of material, etc. As a result, morphological approaches cannot be successfully used to extract ROIs in images that differ significantly from the training images, which the structuring elements are based on.

Feature extraction methods, which use pre-engineered (or manually specified) features, have worked well in areas of pattern recognition, such as face detection, pedestrian tracking, etc. However, classifying images based on pre-engineered features results in a poorer generalization capability than recent deep learning-based automatic feature extractors (LeCun et al. 2015). As a result, the use of feature extraction methods for classifying images leads to a reduction in performance if the images used for testing vary significantly from the images used for training. Traditional feature extraction methods thus lack the generalization capability to deal with wastewater pipeline CCTV images that are known to exhibit large variations arising from differences in pipe geometry, materials, nature of defects, presence of internal linings, camera specifications, etc. Previous automated wastewater pipeline CCTV image classification methods have yielded high defect classification accuracies when applied to small datasets of images collected from a few sources. For instance, the method proposed by Halfawy and Hengmeechai (Halfawy and Hengmeechai 2014c) yielded an average classification accuracy of 86% in classifying root intrusion defects, when tested on a set of 100 wastewater pipeline CCTV images collected from Regina and Calgary, in Canada. However, in order to develop an automated system for interpreting CCTV inspection videos, the classification performance, as measured by the accuracy, precision and recall of the classifier, should be tested on significantly larger datasets (i.e., few thousands) of images collected from multiple pipeline inspections.

2.2 Deep Learning Approaches

In recent years, there have been rapid improvements in image classification tasks due to the advent of deep learning. The breakthrough in deep learning occurred when Krizhevsky et al. (2012) created AlexNet, a deep convolutional neural network (CNN) with five convolutional layers, which won the ImageNet contest to classify 1.2 million high-resolution images. This breakthrough demonstrated that CNNs had a better generalization capability than feature extraction methods such as speeded up robust features (SURF) by Bay et al. (2008) and histograms of oriented

gradients (HOG) by Dallal and Triggs (2005). Generalization capability in this context refers to the ability to correctly classify images that exhibit significant variations in size, shape, color, or texture. Deep learning-based methods have a better generalization performance compared to previous methods since they do not use pre-engineered (i.e., manually specified) features for classifying images. Previous methods such as SURF and HOG use manually specified features (e.g., edges, corners, and gradients) as indicators for classifying images. Deep learning-based methods instead utilize a technique called feature learning, which does not rely on manually specified features. Instead, the optimal features for classifying images are automatically learned through an algorithm called backpropagation. Deep learning-based image classifiers have the capability to learn tens of thousands of optimal image features for classifying images, and these features are not limited to edges, corners, or gradients. As a result, deep learning-based methods can discern intricate patterns in images resulting in substantial improvements in generalization capability over previous methods such as SURF and HOG (LeCun et al. 2015).

Chapter 3 of this dissertation describes the development of a CNN-based method for defect classification and has been published as Kumar et al. (2018). The automated system proposed by Kumar et. (2018) achieved an average classification accuracy of 87% on a test set comprising of 2,000 images extracted from CCTV inspections of 8-inch and 10-inch diameter VCPs, prestressed concrete cylinder pipes, and ductile iron pipes. Their method took sewer images as input and used an ensemble of binary CNNs to identify whether the images contained root intrusions, deposits, or cracks. Hence, Kumar et al. (2018) demonstrated that CNNs could classify sewer CCTV images more accurately than the previous methods which used feature engineering.

CHAPTER 3. CNN-BASED AUTOMATED DEFECT CLASSIFICATION

[A version of this chapter has been published in the journal *Automation in Construction*]¹

In the last 5 years, there has been an emergence of deep learning algorithms for image classification, especially CNNs. CNNs discover tens of thousands of optimal pixel signatures to classify images with, resulting in significantly higher classification accuracies and a better generalization capability than previous methods (LeCun et al. 2015). A previous shortcoming of CNNs was the need for large datasets of training images and a subsequent high computational cost. However, this shortcoming has been overcome through the establishment of well-annotated databases such as ImageNet; and through advances in parallel computations using graphic processing units (GPUs) (Russakovsky et al. 2015). CNNs are capable of differentiating between a large number of object categories, as evidenced by their state-of-the-art classification performance on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) datasets which contain 1000 object categories (Krizhevsky et al. 2012).

Recently, CNNs have made large strides in the automated interpretation of endoscopic images, which display large variations in illumination, shape, and texture (Greenspan et al. 2016). In the civil engineering domain, Soukup and Huber-Mork (2014) used CNNs for classifying defects in rail surfaces, and Cha and Choi (2017) used CNNs for classifying deposits in concrete surfaces. Kim et al. (2017) used a region-based CNN for detecting equipment on construction sites. The images in these previous studies, however, were obtained under relatively controlled conditions, making them more homogenous than images collected from wastewater pipeline CCTV inspections. In order to develop a wastewater pipeline CCTV image classifier with a high generalization capability, it is essential to carefully configure a CNN architecture by using training

¹ Kumar, S. S., Abraham, D. M., Jahanshahi, M. R., Iseley, D. T., and Starr, J. (2018). Automated Defect Detection in Sewer Closed Circuit Television Inspections using Deep Convolutional Neural Networks, *Automation in Construction*, 91, 273 – 283, <https://doi.org/10.1016/j.autcon.2018.03.028>.

images collected from multiple sources under a wide variety of conditions. The next section describes the development of a CNN-based automated defect classification system.

3.1 Proposed Framework

The proposed framework uses an ensemble of binary CNNs where each CNN is trained to classify images as containing or not containing a particular type of defect and supports the classification of multiple defect/feature instances in images. The proposed system takes CCTV inspection videos as input and identifies the presence of two operational defects: root intrusions and deposits, and one structural defect: cracks. Training each CNN to classify a particular defect i consists of feeding the CNN large datasets of positive samples (i.e., images containing defect i) and negative samples (i.e., images not containing defect i). The CNN uses the positive and negative samples to discover features or pixel signatures to distinguish between images. The selection of features to distinguish between images is iteratively optimized through a process called backpropagation. In order to improve the generalization capability of the classifier, it is vital to use training samples that are representative of the commonly occurring variations in wastewater pipeline CCTV images. As a result, training images were collected from over 200 CCTV inspections of pipelines with varying materials (i.e., VCP, DIP, and PCCP) and geometries (i.e., diameter 8-in, 10-in, and 12-in) that were captured using different brands of CCTV equipment (see Figure 3.1).

Images for Automated Defect Classification using CNNs (Chapter 3)

Task	Operational Defects			Structural Defects				Other (including no defects)	Total
	Root	Deposit	Debris	Crack	Fracture	Joint	Infiltration		
Training	3000	1800	800	1200	-	-	800	2400	10000
Testing	500	300	200	200	-	-	200	600	2000

Pipe locations: Georgia and California **Pipe diameters:** 8-in, 10-in, and 12-in **Pipe materials:** DIP, PCCP, and VCP

Figure 3.1 Description of the images used for training and testing the CNN classifiers

The overall CNN classification framework used in this study is as follows: first CCTV inspection videos are converted into a sequence of RGB image frames. Each image frame is then passed through multiple CNNs, with each CNN being trained to classify a particular type of defect (see Figure 3.2). This approach enables the classification of multiple types of defects in a given image. While the focus of this study is detecting the presence of defects in images, techniques to determine

the camera position, such as the optical flow method proposed by Halfawy and Hengmeechai (2014c) can be integrated with the system. Knowledge of the camera position facilitates localization of defects, which is mandatory according to wastewater pipeline defect reporting standards such as the Pipeline Assessment and Certification Program (PACP) (NASSCO 2010).

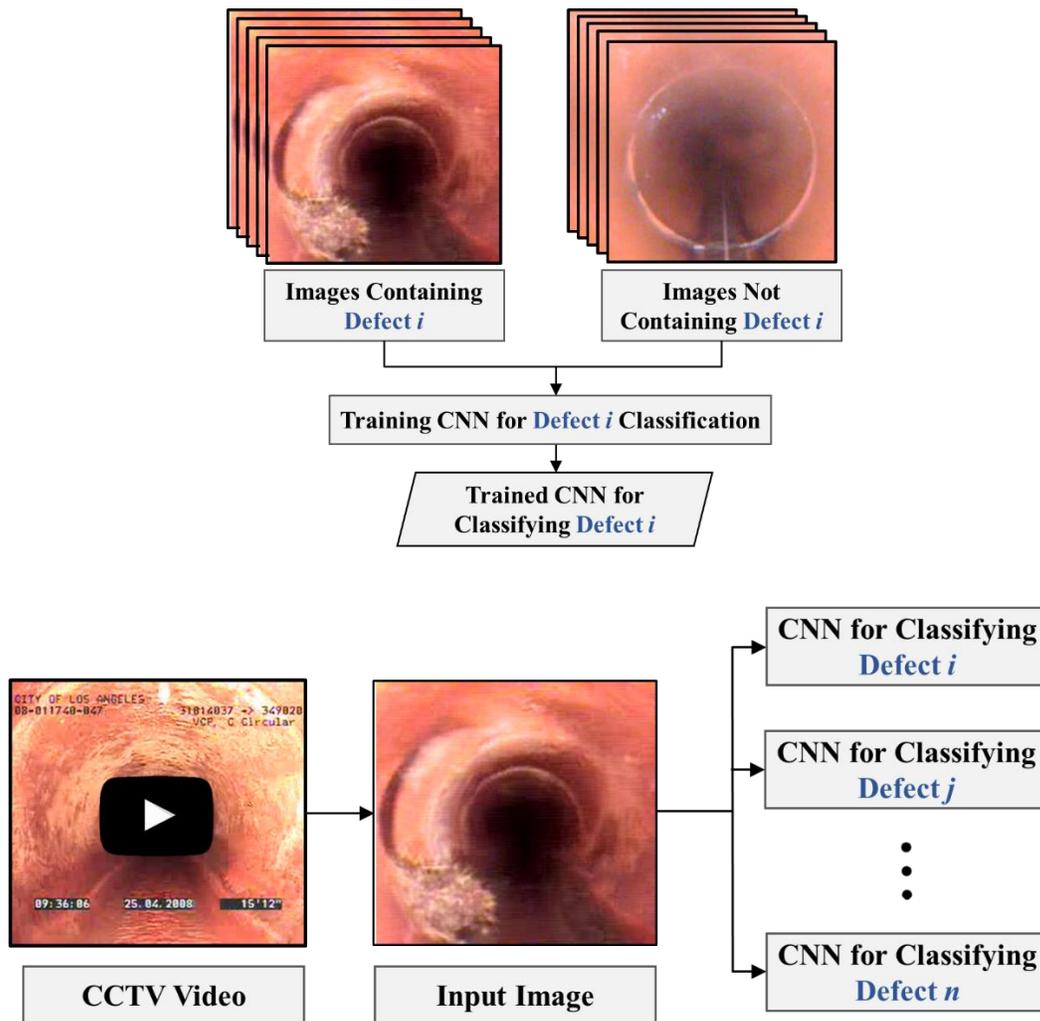


Figure 3.2 Overall training and classification pipeline

3.2 Architecture of the CNN

CNNs are an extension of traditional neural networks in that they are constructed using artificial neurons with weights, biases, and activation functions. Artificial neurons are mathematical functions that receive one or multiple inputs and output their weighted sum. There are four main operations in a CNN: convolution, non-linearity (or activation), pooling (or sub-sampling), and

fully-connected (or classification). The CNN used in this study consists of two convolutional layers, two pooling layers, and two fully-connected layers. The input image, which is a matrix of pixel values, undergoes convolution, activation, pooling, and passes through the fully-connected layer, resulting in an output vector. Binary CNNs, use two output channels (or a two-dimensional vector) to indicate whether an image contains or does not contain a particular type of defect (see Figure 3.3). The outputs are one-hot encoded, such that a final output value of $[1, 0]^T$ indicates the presence of a defect, whereas an output of $[0, 1]^T$ indicates the absence of a defect, in an image.

CNNs perform the spatial decomposition of images, over multiple stages (van Noord and Postma 2017). This spatial decomposition is achieved through alternating convolution and pooling operations. Convolutions transform images by a set of filters with limited spatial extent, whereas pooling reduces the dimensionality of the inputs. At each convolution-pooling step, the output of the previous stage is convolved with another set of filters and subsequently pooled to further reduce the dimensionality of the feature maps (LeCun and Bengio 1995). As a result, relatively simple features in the image, with small spatial extents are processed in the initial layers, and more complex visual patterns are discovered in the later layers of the CNN (LeCun and Bengio 1995).

A CNN can be viewed as a large network of interconnected weights, non-linearities, and down-samplings, which transforms an input image into a single output value. The weights of the network are adjusted through backpropagation, such that the CNN outputs the desired value for a set of images known as the training set. This process of adjusting the weights is performed in iterations or epochs, such that the CNN can correctly classify almost all of the images in the training dataset. Training of the CNN is stopped when successive epochs cease to produce improvements in the classification accuracies on the training sets. Another dataset of images known as the validation set is used to measure the classification accuracy of the trained CNN, based on which, the number of layers, size of convolution kernels, activation function, and learning rate are selected. For example, in the proposed CNN switching from one fully-connected layer to two fully-connected layers resulted in a four percent improvement in classification accuracy on the validation sets. As a result, two fully-connected layers were used in the final CNN architecture. A third dataset of images called the testing set is used to measure the actual classification accuracy of the trained CNN. CNNs use the training images to adjust the parameters (weights and biases) of the network,

whereas validation sets are used for selecting the hyper-parameters such as the number of hidden layers, learning rate, loss function, batch size, etc.

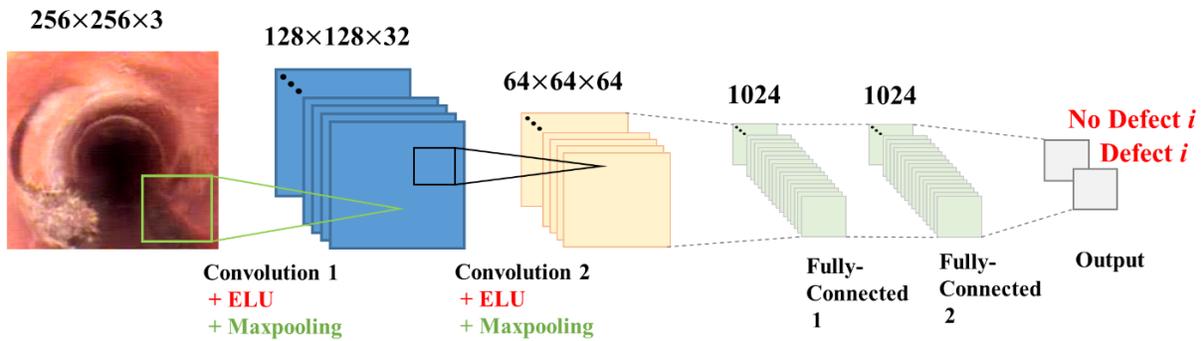


Figure 3.3 Architecture of the CNN for wastewater pipeline feature classification

3.2.1 Convolution

CNNs get their name from the convolution operation, which extracts local features from an input image. The convolution operation is the weighted sum of pixels over local regions in the input. Note: in signal-processing this operation represents cross-correlation and not convolution. However, most deep learning articles refer to this operation as convolution and hence we use the same name. The weights used by the convolution operation are stored in matrices called kernels. As shown in Figure 3.4, computing the weighted sum of a $5 \times 5 \times 3$ patch of pixels with a kernel of $5 \times 5 \times 3 = 125$ weights, results in a single value. Sliding the $5 \times 5 \times 3$ patch of pixels one pixel at a time, to cover all $5 \times 5 \times 3$ regions of the input and computing the weighted sum with the kernel results in a single output layer. A single bias value is then added to all of the values in the output layer, and the resulting output is passed through an activation function. The kernel weights and bias values are initialized with random values, which are then optimized during training.

A convolution operation on an RGB image, using a kernel of size $5 \times 5 \times 3$, can be represented using the following equation:

$$y_{u,v} = \sigma \left(b + \sum_{i=-2}^2 \sum_{j=-2}^2 \sum_{k=0}^3 w_{i,j} x_{u+i,v+j,k} \right), \quad (3.1)$$

where $y_{u,v}$ represents the output value at position (u, v) , σ is an activation function, b is the bias term, $w_{i,j}$ is the weight of each cell in the convolution kernel, and $x_{u+i,v+j,k}$ refers to the input values.

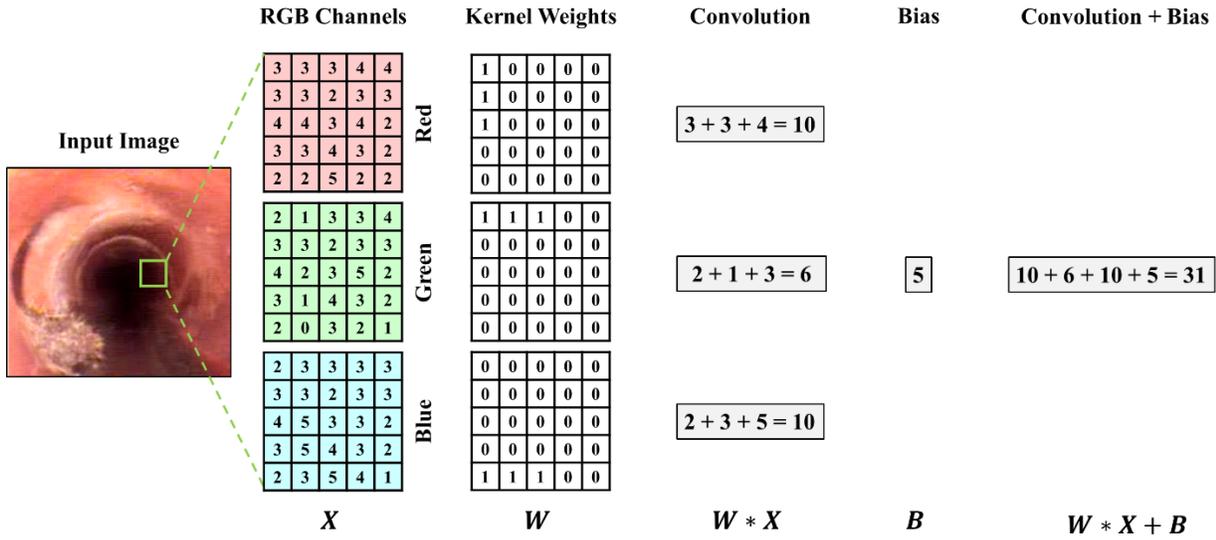


Figure 3.4 Example of a convolution operation on an RGB image

In the proposed architecture, input images are of size $256 \times 256 \times 3$ (length, width, and number of color channels). Each input image is convolved with 32 different kernels, each having a size of $5 \times 5 \times 3$ (length, width, and number of color channels). In order to ensure that the input and output have the same dimensions, the images have to be padded with zeros. Each output channel after the first convolution has a size of 256×256 , and there are 32 such channels (i.e., one channel for each kernel). The resulting output of size $256 \times 256 \times 32$ is first passed through an activation function and then subsampled to $128 \times 128 \times 3$ using a max pooling operation.

The output of the max pooling operation becomes the input to the second convolution stage. Here the input of size $128 \times 128 \times 32$ is convolved with 64 different kernels, each of size $5 \times 5 \times 32$, resulting in 64 output channels each of size 128×128 . After adding a bias to each of the 64 channels, the

result is passed through an activation function. The resulting output of size $128 \times 128 \times 64$ is subsampled to $64 \times 64 \times 64$ using max pooling.

The number of kernels in the first and second convolution layers was determined experimentally. CNN architectures with 32, 64, 96, and 128 kernels in each of the two convolution layers were evaluated for classification accuracy. The architecture with 32 kernels in the first layer and 64 kernels in the second layer yielded the highest accuracy on the validation sets.

3.2.2 Activation Functions (ReLU and ELU)

In artificial neural networks, the sigmoid function (hyperbolic tangent function) is typically used to add nonlinearity to the output channels. Introducing nonlinearity is essential, otherwise the CNN would be computing linear combinations of linear functions and would hence be unable to model non-linear functions. Nair and Hinton (Nair and Hinton 2010) demonstrated that the rectified linear unit (ReLU) nonlinear activation function caused CNNs to converge faster than those using sigmoid functions. The ReLU function is defined as the identity line (i.e., $y = x$), for all positive arguments, and zero otherwise. Recently, Clevert et al. (Clevert et al. 2015) demonstrated that the exponential linear units (ELU) activation function outperforms the ReLU function on networks with more than five channels – leading to shorter convergence times and better generalization performance (Clevert et al. 2015). ELU networks have also demonstrated better performance than ReLUs on classification benchmarks such as the CIFAR-10, CIFAR-100 and ImageNet (Clevert et al. 2015). As a result, in this study, we use the ELU activation function, which is represented by the following equation:

$$f(x) = \begin{cases} x, & x > 0 \\ \alpha (\exp(x) - 1), & x \leq 0 \end{cases}, \quad (3.2)$$

where x refers to the net output after convolution, and α is a hyperparameter.

3.2.3 Max Pooling

The convolution and activation operations are followed by a spatial pooling step. Spatial pooling (also known as subsampling or down sampling) reduces the dimensions of a feature map while

retaining the most important information. The max pooling operation outputs the maximum value over different regions of the input (see Figure 3.5). The amount of reduction in dimensionality depends on the size of the window and the stride size (i.e., the number of pixels by which the window is shifted). Max pooling is frequently used in convolutional neural networks to progressively reduce the number of features and the computational complexity of the network. Pooling operations generalize the results after the convolutional stages, reducing the sensitivity of the output to shifts and distortions (LeCun and Bengio 1995). In the proposed CNN architecture one max pooling stage following each convolution is used to reduce the dimensions by a factor of two. The max pooling operation after the first convolution stage reduces the size of each output channel from 256×256 to 128×128 . The max pooling operation after the second convolution stage further brings down the size of each output channel from 128×128 to 64×64 .

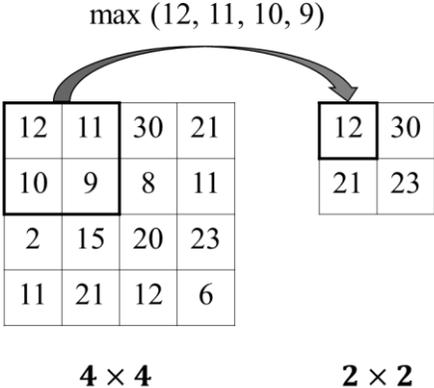


Figure 3.5 Max pooling using a 2×2 window and a stride size of one

3.2.4 Fully-Connected Layers

Fully-connected (FC) layers are commonly used in neural network architectures and are arranged such that neurons between two adjacent fully-connected layers are fully pairwise connected, while neurons within a single layer share no connections (see Figure 3.6). The outputs from neurons in one FC layer undergo a weighted sum and are passed through an activation function. In CNN architectures, FC layers are used at the end of the network after feature extraction and consolidation have been performed by the convolutional and pooling layers. They are used to create final non-

linear combinations of features and for predicting the output (final step in the neural network). Increasing the number of neurons in each FC layer, or increasing the number of FC layers, leads to an increase in the number of parameters (or weights) in the network. For example, two FC layers with three neurons in each layer create nine additional parameters (weights) in the network, assuming that no-biases are added (see Figure 3.6).

The number of FC layers in a CNN architecture is a hyper-parameter, which must be selected based on the classification performance on the validation set. For the proposed binary CNN architecture sample tests were performed using one, two and three FC layers, with 1024 neurons in each layer. On average, there was no decrease in classification accuracy (number of correctly classified images divided by total number of images) when switching from three FC layers to two FC layers. The reduction in one FC layer (from three to two layers) lead to a decrease of around one million parameters in the CNN, resulting in a more streamlined implementation with a reduced likelihood of overfitting. However, switching from two FC layers to one FC layer, resulted in a four percent (average) decrease in classification accuracy. As a result, the final architecture uses two FC layers with 1024 neurons in each layer.

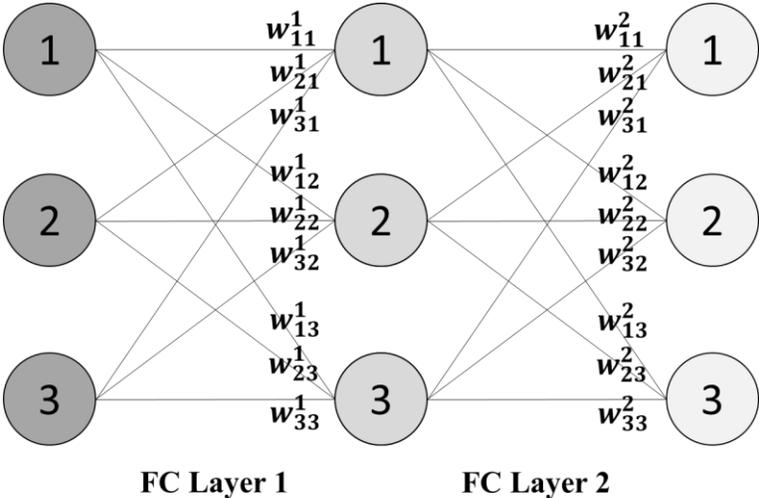


Figure 3.6 Example of two fully-connected (FC) layers with three neurons in each layer

3.2.5 Output Layer

The last layer of the CNN architecture is used to transform the outputs into classification scores. The softmax function, or normalized exponential function is the most commonly used method and it outputs probability scores that add up to 1. The softmax function take as input the predicted class scores, or labels, and outputs a probability score:

$$P(y_i | x_i) = \frac{e^{f_{y_i}}}{\sum_j e^{f_{y_i}}}, \quad (3.3)$$

where, y_i represents the correct label of image x_i , and f_{y_i} represents the predicted score, which is a vector. The CNNs used in this study are binary, i.e. they have two output channels (Defect and No Defect). Since the probability scores for the two outputs must have a sum of one, a score greater than 0.5 is used to indicate the class prediction. For instance, if an input image results in the output channel Defect having a probability score of 0.55, the image would be considered to belong to the class Defect.

3.2.6 Mini-batch Gradient Descent

In the CNN architecture, all of the weights are initialized with random values, and as a result, the predicted output scores and actual output scores do not usually coincide. Training the network involves feeding the network with large datasets of training images, and iteratively adjusting the weights of neurons through a process called backpropagation. Backpropagation uses the output error (i.e., deviation of the predicted outputs from the actual outputs) to adjust the weights of neurons, in a sequential manner, starting from the neurons in the final layer (i.e., output layer) and ending with the neurons in input layer.

The amount of adjustment to the weights is computed using an algorithm known as gradient descent. Gradient descent uses the derivatives of successive layers in the CNN to compute the adjustment in weights, so as to minimize the output error. Conventional gradient descent computes the aggregated output error of the entire batch of training images and is hence computationally expensive. Instead a method known as mini-batch gradient descent is typically used, which splits the training set into smaller batches. The output errors are computed for these smaller batches of

images, rather than for the entire set of training images. Based on the size of our network and GPU memory (12 GB), we selected a mini-batch size of 50 images for training, validation, and testing.

3.3 Techniques to Prevent Overfitting

In the proposed CNN architecture, the number of parameters (i.e., weights in the convolution kernels) is in the order of millions, whereas the number of training samples is in the order of thousands. Since the number of parameters far exceeds the training samples, we risk overfitting the training data. Overfitting occurs when the CNN memorizes the training dataset, and results in high classification performance on the training images, but low classification accuracies on the validation and testing images. The following procedures are implemented in this study to prevent overfitting.

3.3.1 Data Augmentation

Data augmentation is a common method used to avoid overfitting (Chatfield et al. 2014; Krizhevsky et al. 2012; Radford et al. 2015). In data augmentation, the number of training images is considerably increased by applying label preserving transformations on images (Krizhevsky et al. 2012). In this study, random flips, brightness changes, contrast changes, and motion blur were applied sequentially to each input image (see Figure 3.7). The data augmentations boosted the size of the training dataset by a factor of 1000, to over 12 million images.

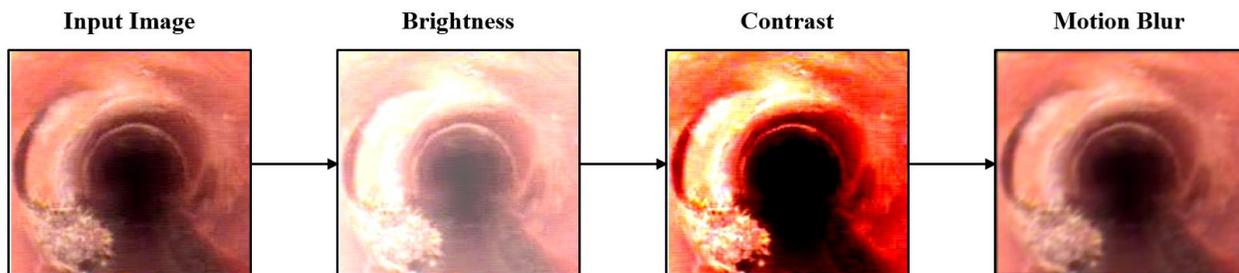


Figure 3.7 Sample data augmentation through changes in brightness, contrast, and motion blur

3.3.2 Dropout

In neural network architectures, regularization or having an even distribution of weights is essential to avoid the classifier from overfitting the data. L1 regularization, L2 regularization, and max-norm constraints are the commonly used forms of regularization (Srivastava et al. 2014). Srivastava et al. (2014) developed an effective regularization method called Dropout, which has been shown to reduce overfitting and significantly outperforming other methods. Dropout, which is implemented only during the training stage, makes neurons active with a certain probability. As the network is trained, neurons (and their weights) get randomly deactivated, forcing the network to adapt. As a result, the learned weights get more evenly distributed, leading to better generalization of the predictive capabilities.

In the proposed CNN architecture, a variety of dropout rates were evaluated for training. Dropout rates greater than 0.5 resulted in a significant amount of overfitting, while dropout rates lower than 0.5 resulted in a reduction in validation accuracies. As a result, a dropout rate of 0.5 was used for training implying that neurons are dropped from the network with a probability of 0.5. A dropout rate of 1.0 was used for the validation and testing sets, indicating that none of the neurons were dropped during validation and testing.

3.4 Experimental Results and Discussion

The application of the CNN is demonstrated in classification three of the most commonly encountered wastewater pipeline defects, namely root intrusions, deposits, and cracks. Root intrusions and deposits usually result in a reduction of the cross-sectional area of wastewater pipeline, reducing the flow capacity, and potentially leading to sanitary wastewater pipeline overflows (SSOs). Cracks are classified as structural defects, which if not remediated can lead to fractures and ultimately wastewater pipeline collapses. Wastewater pipeline collapses have been shown to cause sinkholes, leading to the loss of life and damage to property.

3.4.1 Preparation of Training, Validation and Testing Data Sets

A total of 12,000 images were used for training, validation and testing the CNN classifiers (see Table 3.1). The images were collected from over 200 CCTV inspections of 8-inch and 10-inch

diameter vitrified clay pipes (VCPs), prestressed concrete cylinder pipes (PCCPs), and ductile iron pipes (DIPs). The images could be grouped into eight distinct categories: (1) root intrusions, (2) deposits, (3) cracks, (4) infiltration, (5) debris, (6) connections, (7) material change, and (8) general photographs (undamaged pipe sections). Figure 3.8 shows samples of images used for the development of the CNN classifiers.

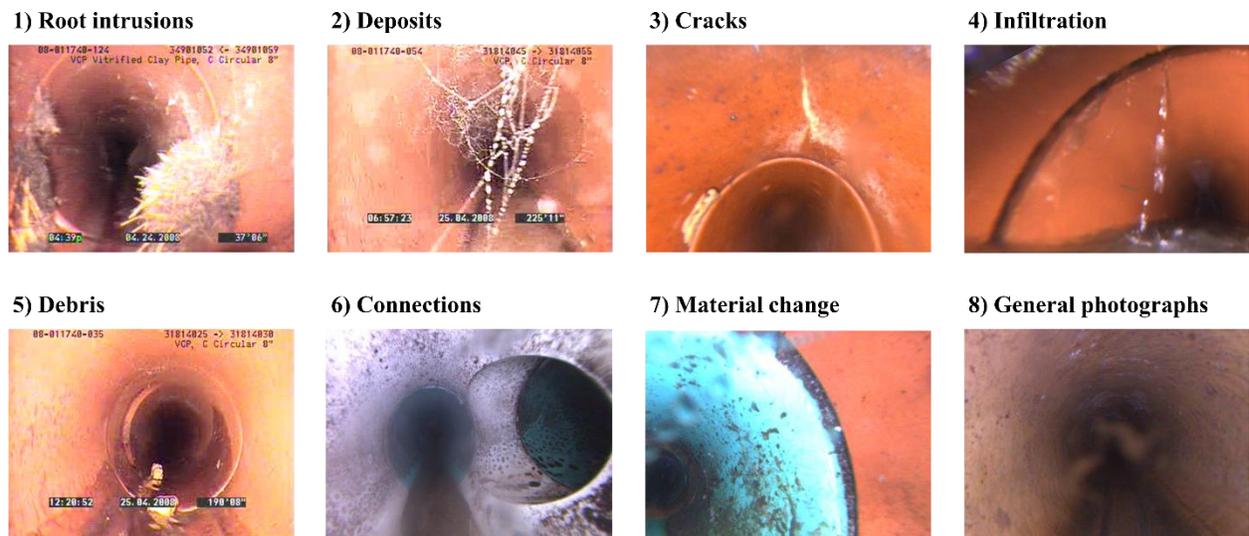


Figure 3.8 Sample images used for training and testing

Ten thousand of the 12,000 images were captured by the RedZone® Solo autonomous CCTV crawler and had resolutions between from 1440×720 and 320×256 pixels. The images were collected from CCTV wastewater pipeline inspections in the state of Georgia, US. The remaining 2,000 images originated from CCTV wastewater pipeline inspections in the state of California, US, and were captured using an unidentified pan, tilt, and zoom camera that produced images at a resolution of 320×256.

CNNs can be theoretically trained using images of any size, while higher resolution images generally provide a greater amount of information than lower resolution images. However, the increase in computational complexity from using higher resolution images results in a significant increase in processing time. As a result, CNNs are usually trained using images with resolutions ranging between 128×128 to 256×256 pixels. In this study, all images were scaled down to

256×256 pixels. The images were then normalized by subtracting the mean pixel intensity for each channel and dividing by the standard deviation of pixel intensity. As a result, the normalized images had a zero mean and unit standard deviation. Normalization ensures that all images have a similar range of pixel intensities, so that the gradients computed during backpropagation do not exhibit disproportionately large variations. Normalization, therefore, improves the speed at which training converges. The images were normalized using the following equation:

$$x_n = \frac{x - \bar{x}}{s}, \quad (3.4)$$

where x_n is the normalized pixel intensity, x is the original pixel intensity, \bar{x} and s refer to the average and standard deviations of pixel intensities of the original image, respectively.

The set of 12,000 images was partitioned into training, validation and testing sets. Seven thousand five hundred images were used for training, 2500 for validation, and 2000 for testing (see Table 3.1). The training, validation and testing sets were created using a five-fold cross validation scheme, such that five triples of training, validation, and testing sets were randomly selected and tested. This approach ensures a reduction in biases while reporting the classification performance. The training, validation and testing sets were equally split into positive and negative image datasets (see Table 3.2). Positive datasets comprised of images containing a particular type of defect, whereas negative datasets comprised of images where that particular type of defect was absent. For instance, in developing a CNN to classify root intrusion defects, images of root intrusions would be assigned positive labels, whereas images of deposits, cracks, infiltration, debris, connections, material change, and general photographs would be assigned negative labels.

Table 3.1 Number of images in each category used for training, validation, and testing

Defect Category	Training	Validation	Testing
Root intrusions	2500	500	500
Deposits	1500	300	300
Cracks	1000	200	200
Infiltration	500	300	200
Debris	500	300	200
Connections	500	300	200
Material Change	500	300	200
General Photographs	500	300	200
Total	7500	2500	2000

Table 3.2 Number of positive and negative samples used

Defect Category	Training		Validation		Testing	
	Positive	Negative	Positive	Negative	Positive	Negative
Root intrusions	2500	2500	500	500	500	500
Deposits	1500	1500	300	300	300	300
Cracks	1000	1000	200	200	200	200

3.4.2 Classification Accuracy, Precision, and Recall

The CNN was developed using Python and the TensorFlow API, which is an open source software library for computation using data flow graphs (Abadi et al. 2015). Training, validation and testing were implemented using a UNIX server, and accelerated using an NVIDIA Titan X 12GB DDR5 GPU. Training converged between 50 and 70 epochs (iterations) and took approximately 500 seconds for each CNN. The criteria for convergence was achieving the maximum accuracy on the validation sets. Multiple training and validation iterations were performed to optimize the hyper-parameters such as number of layers, number of neurons in each layer, convolution kernel size, convolution stride length, etc. The entire process was repeated five times, one for each pair of training, validation, and testing sets. Figure 3.9 shows the training and validation accuracies for the five folds of training and validation sets, for each defect category. The classification accuracies were calculated using Equation 1.1.

The training accuracies were nearly equal to 100% for each of the five-folds, across each defect category. The average validation accuracies averaged over the five-folds were 92.8%, 90.72%, and 90% for the root intrusions, deposits and cracks, respectively. The maximum validation accuracies over the 5-folds were 93.2%, 91%, and 90.4% for root intrusions, deposits and cracks, respectively. The average validation accuracy for root intrusions, deposits, and cracks, taken together was 91.62%. The difference between training and validation accuracies of approximately 8.48% indicates that overfitting exists in the network. Overfitting could be a result of the large number of parameters (i.e., greater than one million parameters) that are trained in the network (see Table 3.3). The average validation accuracy for root intrusions was the greatest followed by deposits and finally cracks.

Table 3.3 Total number of parameters optimized during training

Layer	Size	Number of Parameters
Convolution (first)	5×5×3 (kernel dimensions)	$(5 \times 5 \times 3 + 1) \times 32 = 2432$
Convolution (second)	5×5×32 (kernel dimensions)	$(5 \times 5 \times 32 + 1) \times 64 = 51,264$
Fully-connected (first)	1024 (input)×1024 (output)	$(1024 + 1) \times 1024 = 1,049,600$
Fully-connected (second)	1024 (input)×2 (output)	$(1024 + 1) \times 2 = 2050$
		Total = 1,105,346

Five batches of 2000 images (one for each of the five-folds), which was previously unseen by the CNN, were used for testing the trained and validated CNN. The confusion matrix of the classifiers averaged across five-folds of testing sets is shown in Table 3.4.

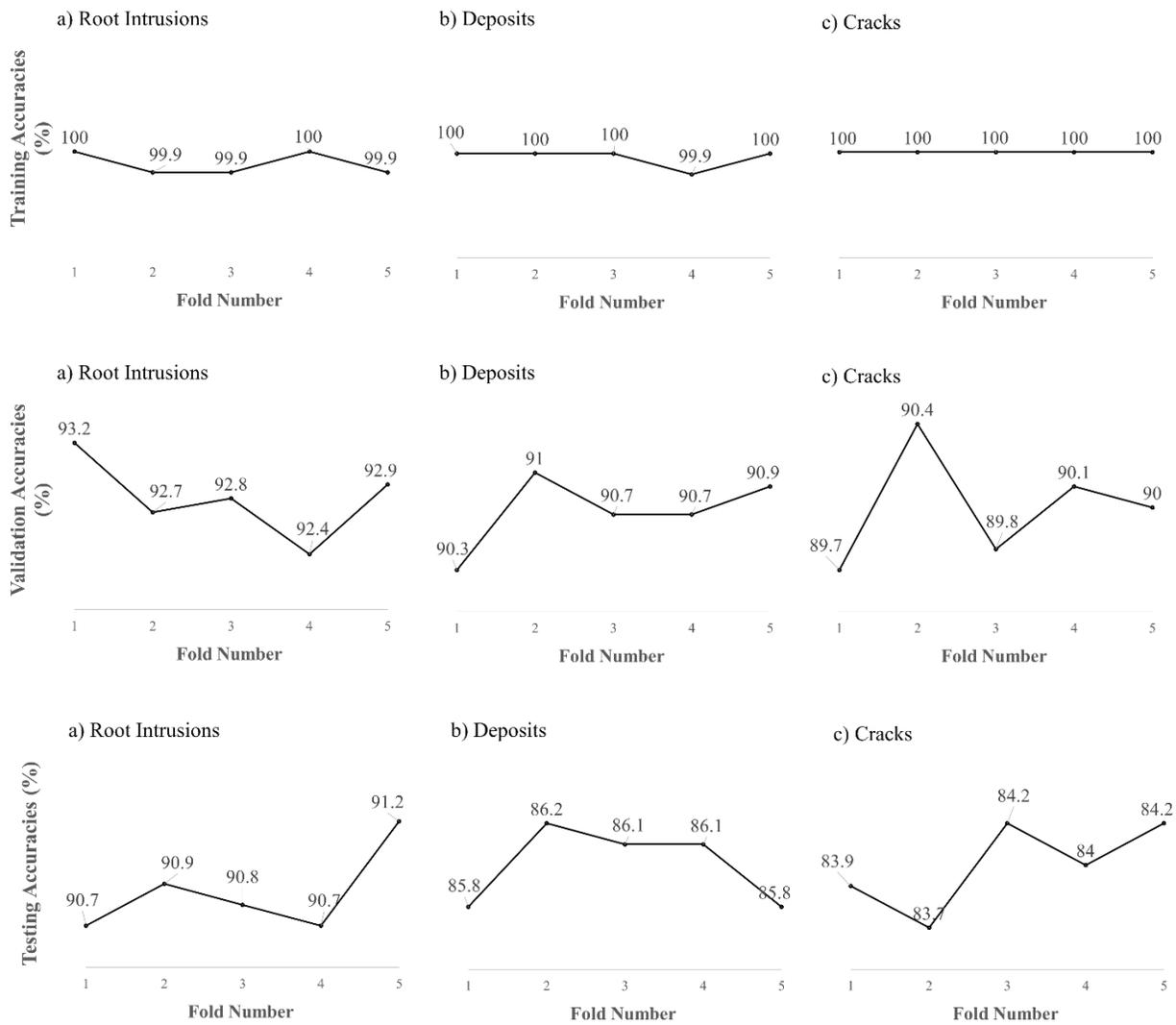


Figure 3.9 Training, validation, and testing accuracies on the 5 folds of root intrusion, deposit, and crack defect images

Table 3.4 Confusion matrix of testing sets (averaged across 5-folds)

Defect Category	Positive Images	Negative Images	TP	TN	FP	FN	Accuracy	Precision	Recall
Root	500	500	471	438	62	29	0.909	0.883	0.942
Intrusions									
Deposits	300	300	261	255	45	39	0.860	0.853	0.870
Cracks	200	200	174	162	38	26	0.840	0.821	0.870
Total	1000	1000	906	855	145	94	0.862	0.877	0.906

Two measures of classification performance, precision and recall were calculated using the equations 1.2 and 1.3 respectively. The average testing accuracy, precision, and recall averaged over root intrusions, deposits, and cracks, and aggregated over the five-folds of 2000 testing images, were 0.862, 0.877, and 0.906, respectively. The average testing accuracies on root intrusions, deposits, and cracks were 0.909, 0.860, and 0.840, respectively. In comparison, the method proposed by Halfawy et al (2014a), which was based on HOG features and SVM classification, achieved an accuracy, precision, and recall of 0.86, 0.86, and 0.86, respectively, in classifying images of roots, when tested on a dataset of 100 images. The method proposed by Halfawy et al. (2014b), which is based on edge detection and morphological segmentation, achieved a marginally higher crack detection accuracy, precision, and recall of 0.85, 0.83, and 0.88, respectively. However, their method was tested on a set of 100 images, which is significantly smaller than the sample size used for testing in this study (i.e., 400 images).

The classification performance, measured by accuracy, precision and recall over the testing sets, was greatest for root intrusions, followed by deposits and cracks. The classifiers generated a larger number of false positives (145) than false negatives (94), indicating a conservative classification approach. The higher classification performance for root intrusions compared to cracks and deposits could be attributed to the larger number of samples used to train the root intrusion classifier. Another reason for the higher classification performance of root intrusions could be the distinct appearance in terms of shape, size, color, and texture of roots in comparison to deposits and cracks.

A potential cause for the inaccuracies in defect classification (across all categories of defects) is that the classification was based on static two-dimensional (2D) images, which lead to a loss of

depth information. Without depth information images that have similar silhouettes appear almost identical, leading to misclassification by the CNN. For instance, in static 2D images, the footprint of cracks, infiltration, and fine roots appear very similar, making it difficult to distinguish between these classes of defects (see Figure 3.10). The use of temporal relationships between successive image frames could provide the necessary information to avoid the aforementioned misclassification errors. CNN classifiers that utilize temporal information from sequences of images rather than single static images could thus improve the classification performance (Chen and Jahanshahi 2017).



Figure 3.10 Examples of similarities in appearance between different categories of images

3.5 Chapter Summary

This chapter described a framework for automated defect classification in CCTV inspections of wastewater pipelines using multiple binary CNNs, each trained to classify a specific class of defect. By passing CCTV wastewater pipeline images through an ensemble of binary CNNs, defects belong to multiple categories can be classified in the same image. The binary CNN used in this study consisted of two layers of convolution, activation (using the ELU function), max-pooling, and fully-connected operations, followed by one output layer.

Many of the previously proposed automated defect classification approaches used feature extraction and morphological methods, leading to poor generalization capabilities. The proposed system significantly improves upon the generalization capabilities of previously proposed automated defect classification approaches, by utilizing deep neural networks trained on a set of

12,000 images from over 200 wastewater pipeline inspections. A data augmentation scheme involving changes in brightness, contrast and motion blur was applied to increase the size of the training sets, to reduce overfitting in the network. The average values of accuracy (0.862), precision (0.877), and recall (0.906) on the testing sets show that the proposed CNN can be used for automated classification of root intrusions, deposits and cracks in CCTV wastewater pipeline videos. Furthermore, the proposed system was tested on images obtained from a diverse set of pipes (i.e., geographic location, diameter, and materials) indicating that the trained CNNs can be used to classify wastewater pipeline CCTV images to a reasonably high accuracy, even when the images exhibit variations due to differences in pipe materials, diameters, and illumination conditions.

At the current stage, the proposed CNNs can be used to classify root intrusions, deposits, and cracks, which are broad classes of wastewater pipeline defects. However, the CNNs do not support localization of defects. According to NASSCO, which sets the standard of sewer inspection reporting in the United States of America, it is mandatory for an inspection report to contain the longitudinal (i.e., distance from the manhole or point of entry) and circumferential position (i.e., position relative to the pipe's cross-section) of defects (NASSCO 2018). Most CCTV crawlers have equipment to measure the distance travelled by the robot, which is typically displayed at one of the corners of the recorded video. Hence, the longitudinal position of a defect can be obtained relatively easily by using a text recognition algorithm on the images and reading the distance travelled (Jahanshahi 2011; Halfawy and Hengmeechai 2014c; Dang et al. 2018). Identifying the circumferential location of defects is significantly more complicated and has been addressed by few studies in this domain. The systems proposed by Xu et al. (1998), Moselhi and Shehab (1999), Guo et al. (2009), Halfawy and Hengmeechai (2014a), Halfawy and Hengmeechai (2014b), Su and Yang (2014), Moradi and Zayed (2017), Hawari et al. (2018), and Kumar et al. (2018), perform defect classification, i.e., given an input image, the system attempts to recognize the type of defect shown in the image. The models used in these studies cannot be directly extended to perform defect localization (i.e., identifying the position and extent of the defects relative to images). Note: Defect localization is a necessary first step in determining the circumferential location of defects. To perform defect localization, these models would have to be applied to images using a sliding-window approach. However, recent deep-learning-based object detection models have been shown

to achieve significantly higher accuracies and processing speeds than sliding window approaches, when tested on benchmark datasets (Girschick et al. 2014; Girschick 2015; Ren et al. 2016). Chapter 4, which has also been published as Kumar et al. (2019) addresses the problem of defect localization.

CHAPTER 4. CNN-BASED AUTOMATED DEFECT DETECTION

[A version of this chapter has been published in the ASCE Journal of Computing in Civil Engineering]²

In computer vision, the process of simultaneously classifying and localizing defects is called defect detection. Figure 4.1 shows an example to illustrate the difference between defect classification and defect detection. To address the limitations of previous studies (i.e., the lack of defect localization capability), Cheng and Wang (2018) used the Faster Region-based Convolutional Neural Network (Faster R-CNN) method for detecting (i.e., classifying and localizing) four different concrete sewer pipe defect types (i.e., root intrusions, cracks, water infiltration, and deposits) in CCTV inspection images. Their study optimized the hyper-parameters (i.e., the size of convolution kernels, stride size, number of parameters, learning rate, etc.) of the Faster R-CNN model by comparing the mean average precision (mAP) of the model across multiple experiments. Cheng and Wang (2018), however, have validated their method only on a limited dataset of images and have not cross-validated the results. Without cross-validation, the reported results contain sampling biases and may not be indicative of the actual performance of the model. Furthermore, their method is validated using specific images of defects and not on actual CCTV videos.

² Kumar, S. S., Wang, M., Abraham, D. M., Jahanshahi, M. R., Cheng, J. C., and Iseley, D. T. (2019). Deep Learning Based Automated Detection of Sewer Defects in Closed Circuit Television Videos, ASCE Journal of Computing in Civil Engineering, 34(1), [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000866](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000866).

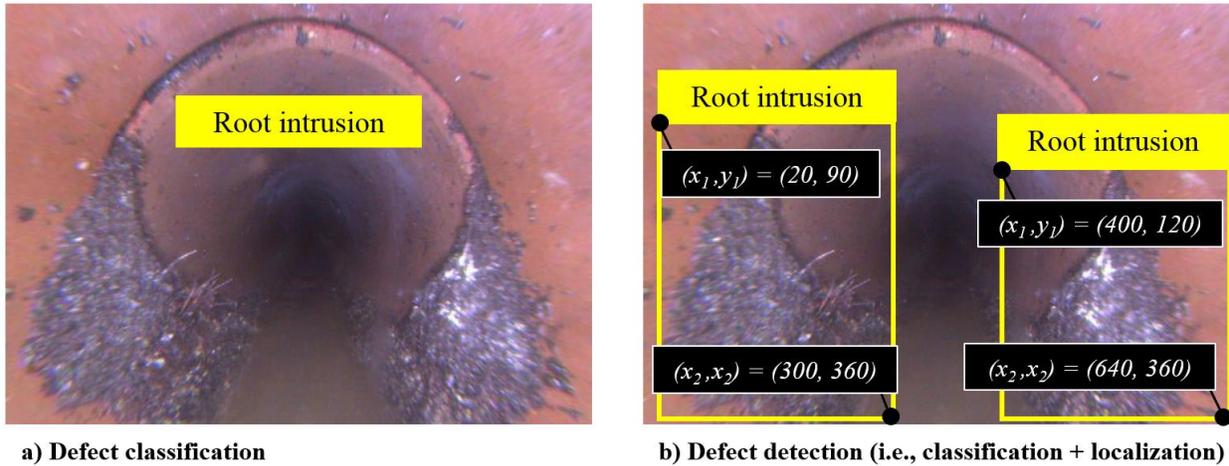


Figure 4.1 a) Example to demonstrate defect classification, b) Example to demonstrate defect detection

Algorithms for automated interpretation of CCTV videos could be installed on desktop computers or servers, facilitating defect detection in videos that have been previously recorded. In such situations, the automated defect detection algorithms could either be used to assist the human inspectors in identifying defects, or for quality checking purposes. Algorithms for automated interpretation of CCTV videos could also be hosted as embedded applications on inspection robots, facilitating real-time defect detection. In such situations, defect detection could be combined with autonomous navigation systems of robots such as the RedZone® Robotics Solo platform. The requirements of the algorithms in terms of computational complexity differ in each situation. For instance, the algorithms hosted on a desktop computer or server have a greater processing power compared to those using the on-board processors on inspection robots. As a result, the algorithms used on a desktop computer or server for offline review of videos may not facilitate real-time defect detection on inspection robots. To determine the most suitable algorithm for each application (i.e., on-board or offline), a comparison of the accuracy and detection speed of various deep learning-based object detection methods must be performed.

This chapter extends the discussion in Cheng and Wang (2018) and Kumar et al. (2018) by describing the development of a prototype system for the automated classification and localization of defects in sewer CCTV videos. The contributions of this study are as follows:

- (1) The speed and classification performance of three state-of-the-art deep learning-based object detection models (i.e., Single Shot Detector (SSD), You Only Look Once version 3 (YOLOv3), and Faster R-CNN) are evaluated within the context of detecting root intrusions and deposits in CCTV sewer inspection videos. Since a diverse set of 3,800 un-augmented images are used for training and testing, the results presented in this study are a better indicator of the performance of these models in detecting sewer defects. Furthermore, sampling bias is minimized by performing 5-fold cross-validation. The results of these experiments will assist developers in selecting appropriate models for real-time and offline defect detection.
- (2) To determine the viability of defect detection in practice, a prototype tool is developed and validated on sample CCTV sewer inspection videos of 20 cm (8-inch) diameter VCP sewers.

4.1 Evaluation of CNN-Based Object Detection Models

Object detection is an extension of image classification, which includes the identification of the location of detected objects in an image, in addition to identifying which category the object belongs to. Deep-learning based object detection gained recognition, when Girshick et al. (2014) developed the region-based convolutional neural network (R-CNN) method which used a method called selective search to produce bounding box region proposals (i.e., rectangular boxes to localize objects). The R-CNN applied a CNN classifier to each bounding box region proposal in an image, to classify the object in that region. However, The R-CNN method used 2000 region proposals and 2000 CNN classifications per images, resulting in the R-CNN method being computationally expensive. The Fast R-CNN by Girshick (2015), improved upon the speed of the R-CNN, by using a single CNN for all region proposals, rather than one CNN for each region proposal. The Faster R-CNN by Ren et al. (2016) improved upon the speed of the Fast R-CNN method by using a trained region proposal network (i.e., a CNN trained to detect ROIs in an image) which is much faster than the selective search region proposal method used in R-CNN and Fast R-CNN. Currently, deep learning-based object detection models fall into three categories: (1) sliding window approach where an image is divided into small sub-windows, which might have significant overlap, and each sub-window is analyzed for the existence of an object (e.g., Atha and Jahanshahi 2017, and Chen and Jahanshahi 2018), (2) two-stage object detection frameworks that improve upon the speed of sliding window approaches by combining a CNN classifier with a region proposal such as the RCNN, Fast R-CNN, and Faster R-CNN, and (3) single stage detectors

such as the SSD by Liu et al. (2015) and YOLO detector by Redmon et al. (2015). Sliding window approaches use CNNs or other neural network architectures over multiple regions of an image and aggregate the classifications. As a result, these methods have a high accuracy of object detection; however, they are typically too slow to be used for object detection in videos and are thus not considered in this study. Two-stage object detection frameworks are known to be highly accurate at detecting objects; however, they have a computational overhead compared to single stage detectors due to the additional region proposal step. Owing to their trade-offs between speed and accuracy, each method finds application in different tasks. The three object detection models which are evaluated in this study are the: (1) SSD, which is known to be a computationally inexpensive object detection model (Liu et al. 2015), (2) YOLOv3 by Redmon and Farhadi (2018), which is computationally more expensive than the SSD, and (3) Faster R-CNN, which is the most computationally expensive of the three models.

To determine the correctness of a detection, a metric called the intersection over union (IOU) is used in this study. IOU measures the degree of overlap between a ground-truth image (i.e., the bounding box manually created during the preparation of the training data) and the bounding box predicted by the model. IOU is calculated by taking the ratio of the area of intersection to the area of union between a predicted bounding box and ground truth bounding box (see Figure 4.2). Next, a threshold value is selected as the minimum IOU, for a detection to be considered correct. For example, a threshold IOU of 0.2 implies that detections which have an IOU less than 0.2 would be considered incorrect.

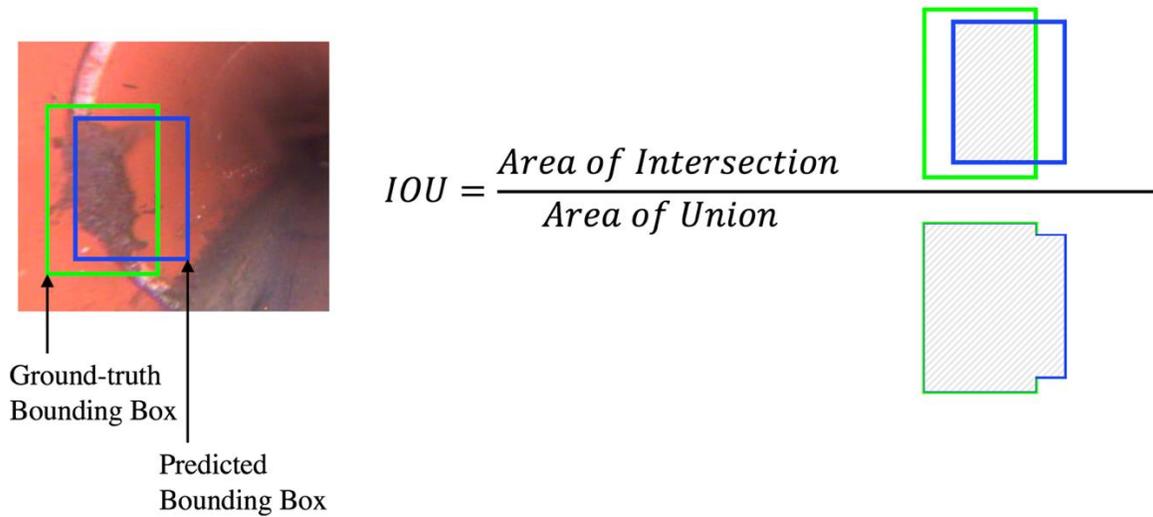


Figure 4.2 Illustration of intersection over union

Training an object detection model consists of three steps: (1) performing the forward pass, (2) computing the loss, and (3) updating the weights of the network. During the forward pass, a training image (i.e., an image in which the defects have been manually identified) is passed through the model, and the resulting output is calculated, usually as a one-dimensional array of outputs. The output is then compared with the ground-truth output to compute the loss. The goal of training is to minimize this loss (i.e., the difference between predicted outputs and ground-truth outputs). The loss is minimized through an algorithm known as backpropagation, which uses the computed loss to update the weights of the model. By training a model with a variety of images, the model ‘learns’ to predict the correct output for previously unseen images.

4.1.1 Single Shot Multibox Detector (SSD)

This model is so named because the task of object classification and localization are performed in a single forward pass of the network. SSD’s improvement in speed comes from the elimination of bounding box proposals and the subsequent feature resampling stage (Liu et al. 2015). The SSD model is based on a feed-forward convolutional neural network. Similar to the Faster R-CNN and unlike YOLO, the SSD model uses the concept of default anchor boxes, which are used to indicate the position of an object. The SSD generates 8,732 default bounding box detections per object class, across six feature maps, for each input image (refer to Figure 3, A + B + C + D + E + F).

For instance, if there are three object classes (e.g., defect 1, defect 2, and background), the total number of detections made by the model, per image is 26,196 (i.e., $8,732 \times 3$). The SSD may produce several overlapping bounding boxes for each object detected in an image. Since only one bounding box per object is desired, a technique called non-maximum suppression, which computes predicted class confidences and uses an intersection over union (IOU) threshold is applied to discard the extra bounding box predictions per detected object. The original SSD implementation by Liu et al. (2016) consisted of adding convolutional feature layers to the end of a truncated VGG-16 (originally proposed by Simonyan and Zisserman (2014)). However, any convolutional neural network architecture can be used as a base network. The model can be made faster with a slight loss in accuracy by replacing the VGG-16 base network with MobileNets (developed by Howard et al. (2017)) and replacing all forward convolutions with depth-wise separable convolutions. MobileNets is significantly faster than VGG-16 since it uses depth wise separable convolutions which requires fewer computations than regular convolutions (Howard et al. 2017). In this study, the SSD is implemented with MobileNets as the base network rather than VGG-16 (see Figure 4.3).

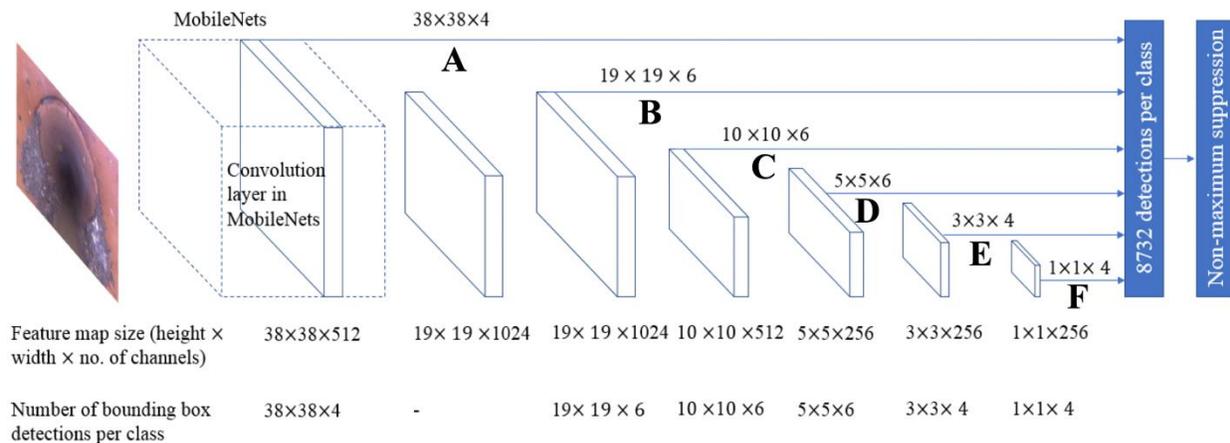


Figure 4.3 Conceptual architecture of the SSD

4.1.2 YOLOv3

You only look once (YOLO) is a unified convolutional network which treats object detection as a regression problem through end to end training. YOLO predicts both the bounding box location and class probabilities simultaneously for each object on a given image (Redmon et al. 2015). The

original YOLOv1 has limitations such as the requirement of fixed input size and poor performance for detecting small objects. YOLOv3 has a significant improvement in terms of detection speed and accuracy. Instead of predicting a fixed number of bounding boxes as YOLOv1, YOLOv3 uses anchor boxes to predict bounding boxes and applies K-means clustering to generate appropriate dimensions for those anchor boxes, such that the network can learn the box locations more efficiently (Redmon et al. 2018). As shown in Figure 4.4, during training, the model divides the image into $S \times S$ grid cells and predicts B bounding boxes for each cell. For each bounding box, there are five predicted values including x, y, w, h and a confidence value. Here, (x, y) represents box centroid coordinates relative to the bounds of the grid cell, while (w, h) represents the box width and height relative to the image, respectively. The confidence value indicates the certainty of the prediction that the bounding box contains an object and how accurately the box fits the object. The confidence is calculated using equation (4.1).

$$confidence = Pr(Object) * IOU_{pred}^{truth} \quad (4.1)$$

where $Pr(Object)$ is equal to 0 if there is no object in the grid cell; otherwise, it is equal to 1. IOU_{pred}^{truth} represents the intersection over union (IOU) between the predicted bounding box and the ground truth bounding box. The ground truth bounding box refers to the bounding box created manually by the human annotator during data preparation. In addition, for each grid cell containing an object, the model predicts probability scores for all the classes, represented by $Pr(Class_i|Object)$. Therefore, in the testing period, each box predicts a confidence score for each class by equation (4.2).

$$Pr(Class_i|Object) * Pr(Object) * IOU_{pred}^{truth} = Pr(Class_i) * IOU_{pred}^{truth} \quad (4.2)$$

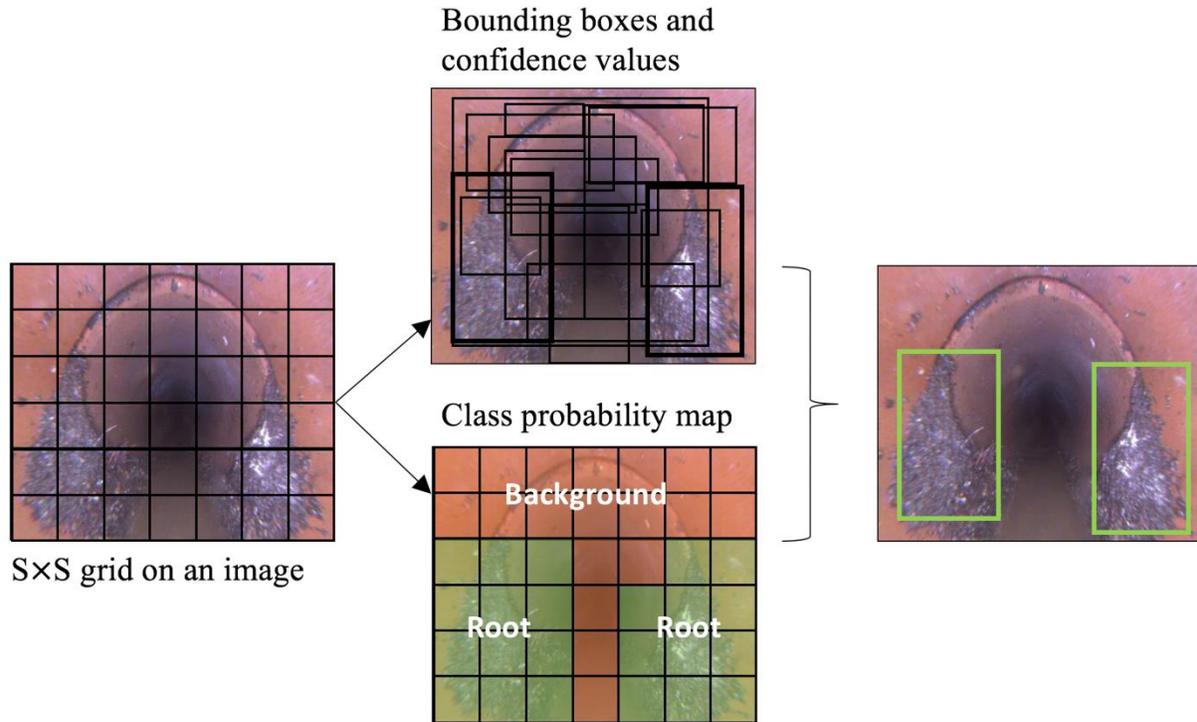


Figure 4.4 An example of how YOLOv3 detects multiple objects in an image

4.1.3 Faster R-CNN

The Faster R-CNN model represents an improvement over the Fast R-CNN object detection model. Unlike the Fast R-CNN model which uses the selective search algorithm to generate bounding boxes, the Faster R-CNN model uses a region proposal network (RPN) (see Figure 4.5). The forward pass in a Faster R-CNN model consists of three main steps. First, a pre-trained convolutional neural network is utilized on the raw images to generate a feature map. The original implementation of the Faster R-CNN method used the VGG12 convolutional neural network to generate the feature map, but recent implementations use the ResNet. Next, a RPN, which is a single fully convolutional network, is applied to the feature map. The RPN identifies regions of interest (ROIs), i.e., areas where an object might be present, and generates bounding box coordinates to indicate the location of these objects. The RPN may generate several bounding boxes for a single object. Hence, an intermediate step called non-maximum suppression is applied such that bounding boxes with a high degree of overlap are discarded. At the end of the RPN step,

the network has produced several object predictions (i.e., regions in the image where there may be an object). The next task is to determine which class each object belongs to (i.e., whether the object is a root intrusion, deposit, or neither). To do so, a region based convolutional neural network (RCNN) is used. The regions in the image corresponding to each bounding box (i.e., for each ROI) are cropped, reshaped, down sampled, and flattened, in that order, and passed through two fully connected convolutional layers. The output of the fully connected layers are a class probability score (i.e., the probability that the ROI represents a particular class) and a bounding box adjustment factor, which attempts to better fit the bounding box to the object.

Since the Faster R-CNN contains two trainable layers (i.e., the RPN and an RCNN), the training objective consists of four losses. These losses are: (1) the bounding box localization loss in the RPN step, (2) the classification loss in the RPN step, (3) the bounding box localization loss in the RCNN step, and (4) the object classification loss in the RCNN step.

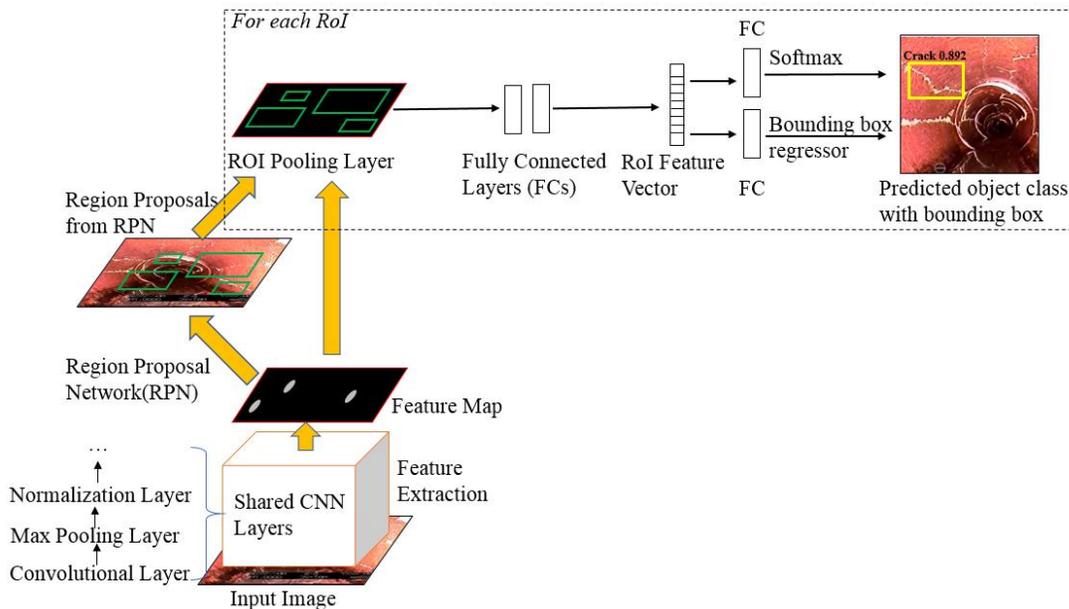


Figure 4.5 Faster R-CNN conceptual architecture (Wang and Cheng, 2018)

4.2 Experimental Results and Discussion

In this study, the automated defect detection systems were evaluated in the context of detecting root intrusions and deposits. Roots enter pipes through loose joints and openings in the pipe wall

due to damage (Stål 2007; Ridgers 2007). Root intrusions can expand existing openings in wastewater pipelines, allowing the surrounding soil to enter through the defect, further weakening the pipe, and ultimately causing breakage and collapse of the wastewater pipeline structure (Schrock 1994). Deposits are typically caused by fats, oils, and grease (FOG), and combined with root intrusions lead to a reduction in hydraulic capacity of the pipe (Marlow et al. 2011). To avoid potential wastewater pipeline overflows arising from such blockages, it is essential to routinely clear the pipes of root intrusions and deposits. Hence, the location and severity of root intrusions and deposits must be known in order to plan maintenance operations on a pipe. In this study, the automated defect detection model characterizes these defects and identifies their location. The severity of defects is however not measured and could be a topic for further investigation.

4.2.1 Preparation of Training and Evaluation Data Sets

A total of 3,800 images were used for training and validation of the defect detection models. The images were recorded using front facing CCTV cameras and had resolutions between 720×576 and 1507×720. The images were recorded from over 200 CCTV inspections of 203 mm (8-inch), 254 mm (10-inch), and 305 mm (12-inch) VCPs, PCCPs, and polyvinyl chloride (PVC) pipes. The sewer pipes were in the states of Virginia and Ohio in the United States of America. The images used in the study depicted eight categories of sewer features including root intrusions, deposits, cracks, infiltration, debris, connections, material change, and general photographs (i.e., undamaged pipe sections). Figure 4.6 shows examples of the images that were used for training, validation, and testing. One thousand one hundred (1100) images contained root intrusions only, 1100 images contained deposits only, 1100 images contained features other than root intrusions or deposits, and 500 images contained root intrusions and deposits simultaneously (see Table 4.1).

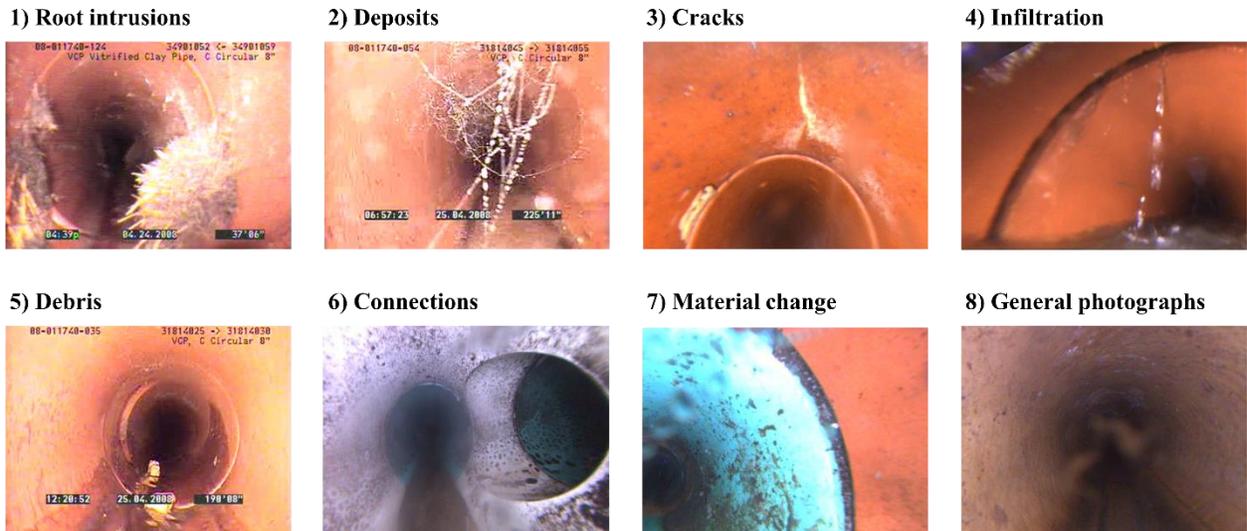


Figure 4.6 Examples of images used for training and testing (Kumar et al. 2018)

To train and validate the defect detection models, the defects in the images should first be manually identified. The manually annotated images serve as ground truth images that are used by the defect detection models to learn to detect defects. The annotated images are also used to evaluate the accuracy of the models by comparing their similarity with the detections automatically generated by the object detection models. LabelImg, an open source image annotation tool, was used during this study (Tzutalin 2015). Annotating an image consists of identifying all instances of features in that image and drawing separate rectangular bounding boxes around each feature. The bounding boxes serve to identify the position and extent of each feature in a rectangular coordinate frame. In this study, the annotations were saved as Extensible Markup Language (XML) files in the Pascal Visual Object Classes (VOC) format (see Figure 4.7).

Table 4.1 Number of Images used for training and validation

Image Category	Training	Validation	Testing	Total
Root intrusions only	880	110	110	1100
Deposits only	880	110	110	1100
Do not contain root intrusions or deposits	880	110	110	1100
Root intrusions and deposits	400	50	50	500
Total	3040	380	380	3800

CCTV images have a high variability, which can make the determination of the type of feature in an image highly subjective. Furthermore, due to the large number of images that were used in this study, the annotations were performed by two individuals. To improve consistency in the annotations approximately 50 random images were initially selected and annotated by both individuals. To improve inter-coder reliability, the 50 annotated images were then compared to highlight inconsistencies in the annotations created by the two individuals.

4.2.2 Training and Evaluation of the Defect Detection Models

In machine learning studies, it is common practice to use between 70 to 90 percent of the data for training and the remainder for testing (Halfawy and Hengmeechai 2014a; Soukup and Huber-Mork 2014; Cha and Choi 2017; Chen and Jahanshahi 2018). In this study, eighty percent of the data (i.e. 3,040 images) were used as the training set, 10 percent (i.e., 380 images) were used as the validation set, and 10 percent (i.e., 380 images) were used for testing. Since the deep neural network architectures evaluated in this study contain over a million parameters, increasing the number of training images would decrease the amount of overfitting and likely improve the classification accuracy on the testing set.

The training and validation sets were created using a 5-fold cross-validation technique, i.e., five sets of training and validation sets were created by randomly partitioning the dataset. Training data are used to compute the gradients for the backpropagation algorithm, whereas the validation data are used to determine the optimal number of training iterations. Since 5-fold cross validation was used, five sets of training and validation tests were conducted, and the classification performance measured in mAP was averaged over the five sets. Based on the average scores, the training

iteration corresponding to the highest mAP on the validation set was chosen as the optimal training iteration. Finally, the model corresponding to the best validation mAP is evaluated upon the testing dataset.

Training, validation, and testing were performed using a desktop computer comprising of an Intel CPU and an NVidia Training, validation, and testing were performed using a desktop computer comprising of an Intel CPU and an NVidia P4000 Quadro CUDA supported graphics card with 8GB memory. To compare training times, each model was trained up to 150,000 iterations. At intervals of 10,000 iterations, each model was saved, and its mAP measured on the validation set. The Faster R-CNN model took the longest time to train (i.e., the time taken to complete 150,000 training iterations) at approximately 12 hours, whereas the YOLO and SSD models took approximately 10 hours and 2 hours to train, respectively. The optimal number of training iterations was determined by plotting each model's mAP (i.e., averaged over 5-folds) in detecting defects in the validation dataset at an IOU threshold of 0.2, and selecting the iteration corresponding to the highest mAP (see Figure 4.8). In Figure 4.8, the mAP on the validation set first increases up to a certain number of iterations (i.e., 50,000 iterations for YOLOv3, 70,000 iterations for Faster R-CNN, and 70,000 iterations for SSD) and then tends to decrease. This decrease in mAP is due to overfitting, i.e., when the model learns the details and noise in the training data such that the performance on the validation set is negatively affected. The mAP at IOU thresholds of 0.3, 0.4, and 0.5 were also calculated and are reported in Table 4.2.

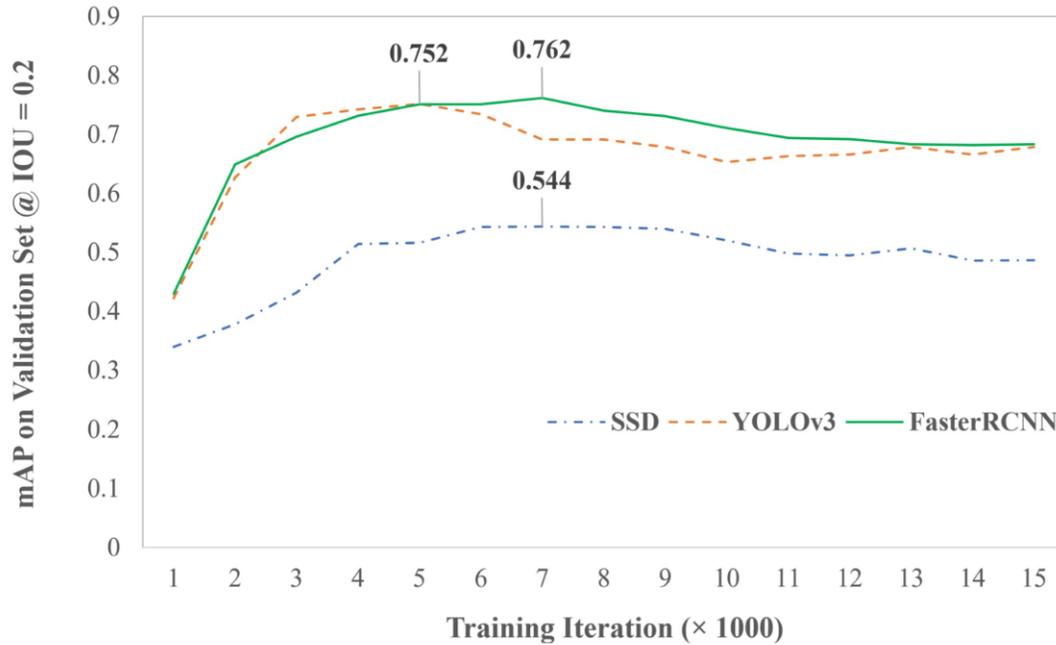


Figure 4.7 Peak mAP curve used to find optimal training iterations for the three models

4.2.3 Performance of the Models

The highest average validation accuracy for the SSD, YOLOv3 and Faster R-CNN were 0.544, 0.745, and 0.762, respectively. The models corresponding to the highest validation accuracy are then tested on the testing dataset. Table 4.2 summarizes the performance for the two categories of defects on the testing dataset. Four threshold IOU values (i.e., 0.2, 0.3, 0.4, and 0.5) are experimented with. An IOU threshold of 0.5 indicates that only predicted bounding boxes which have an IOU greater than or equal to 0.5 will be considered as correct detections. It is common practice to use an IOU threshold of 0.5 for object detection. For instance, the PASCAL VOC Project, which is a benchmark for testing object detection models uses an IOU threshold of 0.5 (Everingham et al. 2010). Sewer defects, however, are often spread out and have discontinuities in their boundaries. Furthermore, it is preferable for the detectors to capture a larger number of defects with a lower localization accuracy, rather than capturing fewer instances of defects with a higher localization accuracy, since missed defects could lead to unforeseen failures. Upon visual observation of the detections produced at various IOU thresholds, it was found that an IOU threshold of 0.2 is appropriate for detecting sewer defects.

Table 4.2 Summary of defect detection average precision values for the different IOU thresholds on the testing set

	Average Precision @ IOU = 0.5		Average Precision @ IOU = 0.4		Average Precision @ IOU = 0.3		Average Precision @ IOU = 0.2	
	Deposit	Root	Deposit	Root	Deposit	Root	Deposit	Root
SSD	0.393	0.496	0.419	0.489	0.498	0.510	0.515	0.545
YOLOv3	0.445	0.589	0.577	0.696	0.646	0.719	0.682	0.708
Faster R-CNN	0.515	0.605	0.624	0.722	0.637	0.728	0.662	0.775

The testing mAP at an IOU threshold of 0.2 was 0.530, 0.695, and 0.718 for the SSD, YOLOv3, and Faster R-CNN models, respectively. Figure 4.9 shows examples of predicted bounding boxes generated by each model at an IOU threshold of 0.2.

In general, the Faster R-CNN model yielded the highest average precision among the three models. The average time taken to evaluate a single image by the SSD, YOLOv3, and Faster R-CNN was approximately 33 ms, 57 ms, and 110 ms, respectively, using the same computing system and an image of size 1500×720 .

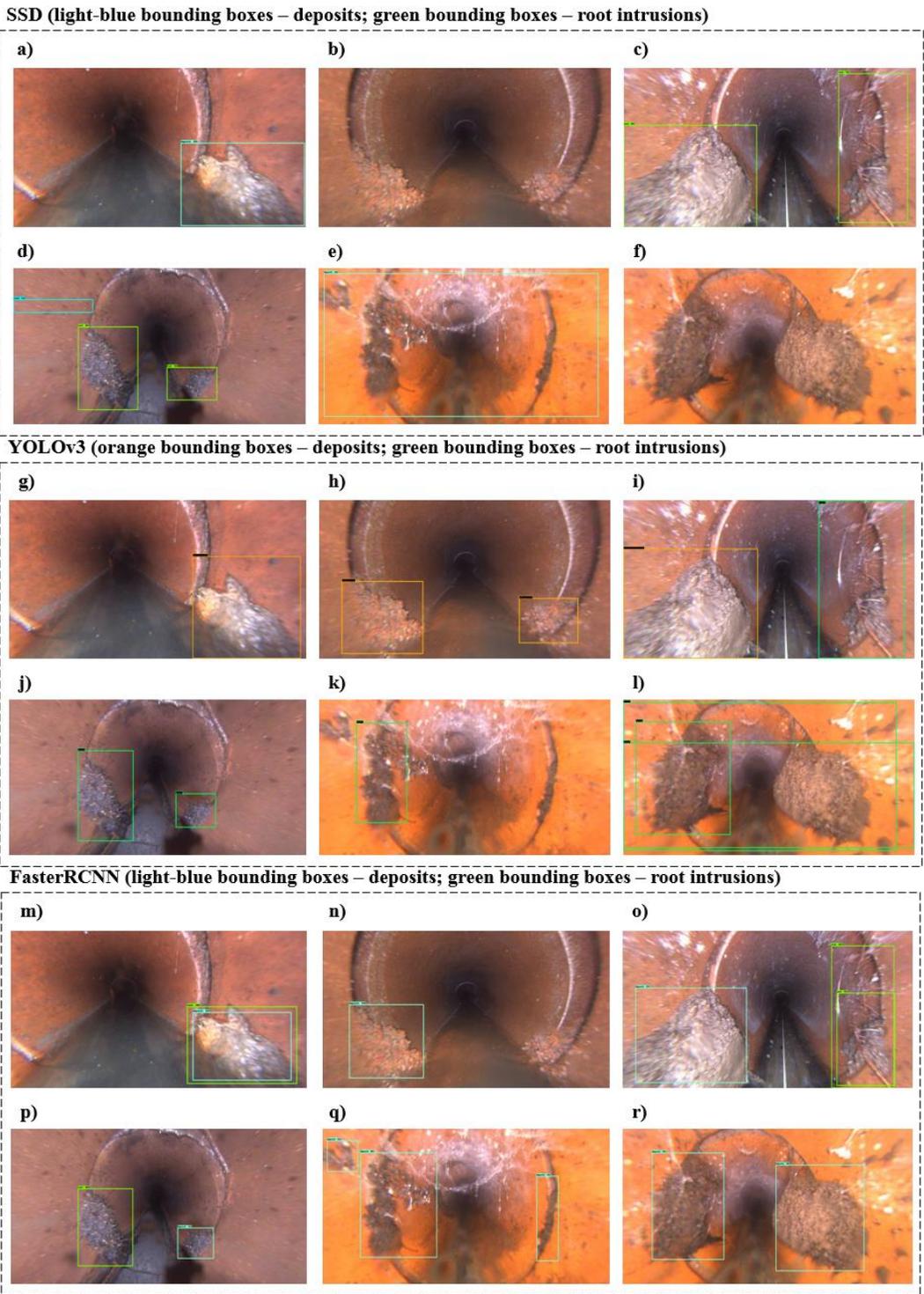


Figure 4.8 Comparison of the bounding box detections by the SSD (a – f), YOLOv3 (g – l), and Faster R-CNN (m – r) models

4.2.4 Discussion of Experimental Results

The SSD was found to be significantly faster than YOLOv3 and Faster R-CNN; however, the classification performance (i.e., as measured by the mAP) was considerably lower at all IOU thresholds. The three models tend to miss certain defects, with this tendency most pronounced in the SSD, as shown in Figure 4.9b, Figure 4.9e, and Figure 4.9f. The Faster R-CNN model had only a slight advantage over YOLOv3 in terms of defect detection accuracy. YOLOv3 however, was found to be almost twice as fast as the Faster R-CNN in evaluating images. Due to its high detection accuracy and speed, the YOLOv3 model could be suitable for deployment on the on-board electronics of autonomous sewer inspection robots. The slower but more accurate Faster R-CNN model could be used for the off-site review of inspection videos.

A potential cause for the inaccuracies in defect classification (i.e., across all categories of defects) is that the classification was based on static two-dimensional (2D) images leading to a loss of depth information. Without depth information, images that have similar silhouettes appear almost identical, leading to misclassification by the CNN. For instance, in static 2D images, the footprint of cracks, infiltration, and fine roots appear very similar, making it difficult to distinguish between these classes of defects (see Figure 4.10). The use of temporal relationships between successive image frames could provide the necessary information to avoid the aforementioned misclassification errors. CNN classifiers that utilize temporal information from sequences of images rather than single static images could thus improve the classification performance.

Inconsistencies in the ground truth annotations (i.e., the manually annotated images) are another cause for incorrect classifications. This inconsistency was typically observed in images of root intrusions and deposits. Due to similarities in the appearance of certain types of deposits and roots, the human defect coders tended to interchange the labels. Furthermore, defects that spanned the entire circumference of the pipe cross-section were often labelled inconsistently by the coders. For instance, the defect in Figure 4.9a could be annotated in multiple ways (i.e., either using three bounding boxes as shown in Figure 4.9b and Figure 4.9c, or using a single bounding box as shown in Figure 4.9d). Since the ground-truth dataset contained biases, all three models trained on this dataset exhibited biases as well. The training data would have been less biased had the images been annotated by experienced CCTV coders.

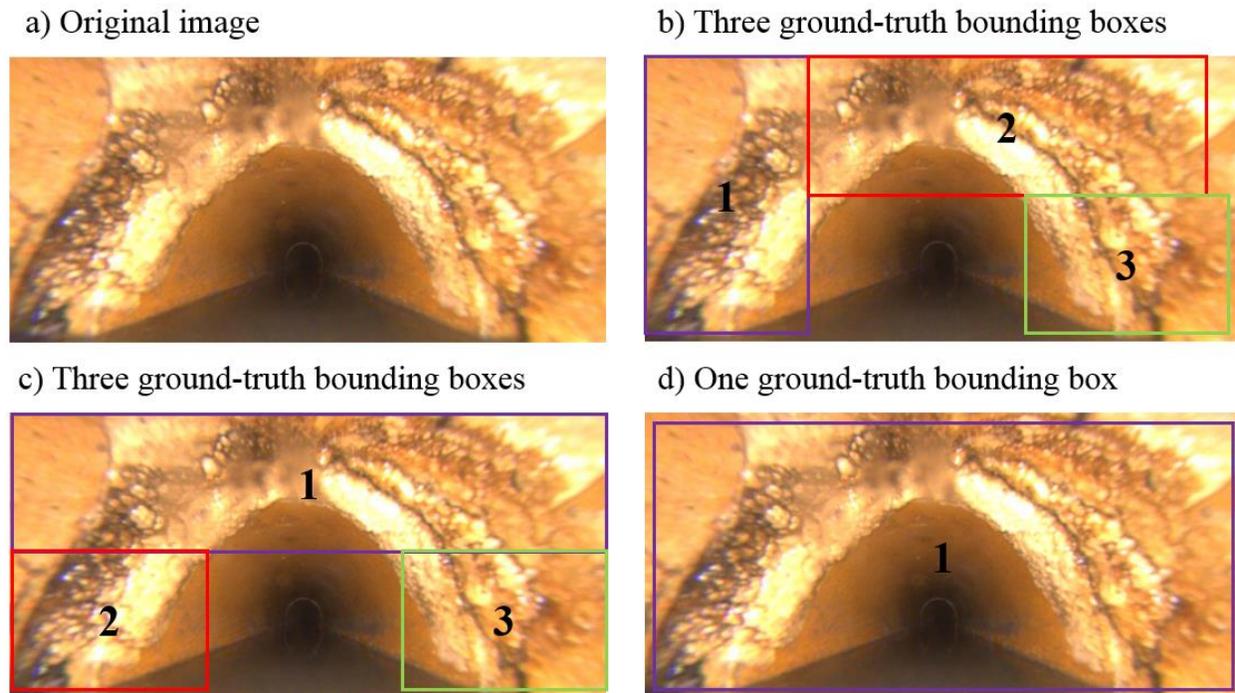


Figure 4.9 Example showing multiple ways to annotating a defect that spans the circumference of the pipe

4.3 Demonstration Example

To demonstrate the viability of the proposed approach in practice, a prototype automated defect detection tool was created using the Faster R-CNN model as the detection engine. The prototype tool takes CCTV videos as input and identifies regions where root intrusions or deposits are present, in real-time. Rather than output two class labels (i.e., root intrusion or deposit), the tool outputs one class label and calls it ‘defect’. The simplification to one class label allows for a reduction in computational complexity, allowing the model to run faster thereby facilitating future deployment on a sewer inspection robot. The identification of defects in this manner would be useful in developing autonomous inspection robots. Autonomous inspection robots could use such a defect detection tool to locate areas in the pipe that contain defects. Once the defect regions are identified, the autonomous robots could then zoom in on those regions to provide a closer inspection. The framework for producing the detections on the video is shown in Figure 4.10. Images are sampled from the video at 30 frames per second using OpenCV (Bradski 2000). Each extracted image is then passed through a Faster R-CNN model to detect defects.

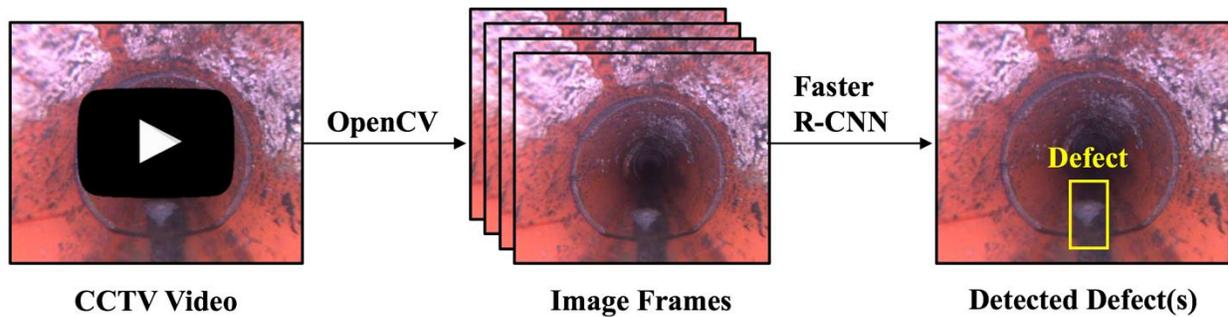


Figure 4.10 Framework for producing defect detections on an input CCTV video

To demonstrate the accuracy of the prototype system, three CCTV inspection videos of 20 cm (8-inch) VCP sewer laterals are considered for evaluation. The three videos cumulatively represent approximately 45 minutes of CCTV inspections and televise approximately 335 meters (1100 feet) of sewer laterals. Out of a known total of 56 instances of root intrusions and deposits defects (i.e., 26 roots and 30 deposits), the automated defect detection tool was able to identify 51 (i.e., 25 roots and 26 deposits) instances of defects (i.e., 91 percent of the defects) while generating 7 false positive detections. Figure 4.11 shows examples of the detections produced by the prototype tool.

The CCTV videos used in the demonstration are sampled at 30 frames per second. The criteria for a defect to be considered ‘detected’ is that a detection should have been produced in at least 1 frame. Similarly, a detection on a non-defect region is considered a false positive if the detection is produced in at least 1 frame. However, this may not be the best approach, and could lead to many false positives. An alternate approach would be to consider different thresholds (e.g., a defect should be detected in 20% of the frames) and evaluate the accuracy, precision, and recall of the automated system to determine the optimal threshold value. The measurement of true positive and false positive detections was performed by a human observer, i.e., through visual inspection of the videos. This approach to measuring true positive and false positive detections was undertaken to simulate the real-world scenario where a CCTV operator observes the video in real time. A sample video demonstrating automated defect detection in videos can be found at (Kumar 2018).



Figure 4.11 Example images showing the defects detected by the automated defect detection tool applied to a CCTV video in real-time

4.4 Limitations

A limitation of this method is that it applies the Faster R-CNN method to every image frame extracted from a CCTV video regardless of whether the image contains a defect or not. However, only a small fraction of the images extracted from a video contain defects. As a result, this method performs unnecessary computations on images that do not contain defects. Furthermore, the method was evaluated for the detection of operational defects only. Chapter 5 of this study, which is partly published as Kumar and Abraham (2019) involves using a two-step framework to improve the speed of detecting defects in CCTV videos and also addresses the detection of structural defects.

CHAPTER 5. TWO-STEP DEFECT DETECTION FRAMEWORK

[A version of this chapter was published in the proceedings of the 2019 ASCE International Conference on Computing in Civil Engineering]³

In Chapter 4, we demonstrated a prototype sewer defect detection system using the Faster R-CNN model for detecting root intrusions and deposits in sewer CCTV videos. Root intrusions and deposits were selected for detection because: (1) they have an adverse impact on the hydraulic capacity of a pipe, and (2) they have the potential to progress into structural defects or exacerbate existing structural defects. The system was first trained and tested using 3,800 images of defects, which were extracted from over 200 CCTV inspections of 8-in, 10-in, and 12-in VCP and Concrete sewers located in the US states of Florida and Ohio. The system was then evaluated on CCTV videos televising 335 meters of sewer laterals. The Faster R-CNN-based system could detect 51 out of 56 instances (i.e., 91 percent of the defects) of root intrusions and deposits while generating 7 false positives.

A limitation of this method is that it applies the Faster R-CNN model to every image frame extracted from a CCTV video regardless of whether the image contains a defect or not, although only a fraction of the entire duration of a video depicts defects. As a result, the previous method performs unnecessary computations on image frames that do not contain defects. As a result, the previous method that we proposed could be slow to process CCTV videos and may not be able to perform real-time detection, when executed on computers without powerful computing abilities (e.g., onboard electronics of CCTV cameras).

This chapter extends the discussion in Chapter 4 by proposing a framework for real-time defect detection. The proposed framework uses a pre-processing step to determine whether an image contains a defect or not and then applies the Faster R-CNN model only on image frames that contain defects. The pre-processing step helps improve the speed and accuracy of defect detection and avoids unnecessary computations on images that do not contain defects. Furthermore, this

³ Kumar, S. S. and Abraham, D. M. (2019). A Deep Learning Based Automated Structural Defect Detection System for Wastewater Pipelines, ASCE International Conference on Computing in Civil Engineering (i3CE 2019), Atlanta, Georgia, USA

report extends the automated system to encompass the detection of root intrusions, fractures, and lateral connections.

5.1 Description of the Framework

Prior methods for automated defect classification in sewers have also used two-step methods. For instance, Sinha and Fieguth (2006a, 2006b) used a morphological method to segment defects (i.e., separate defect pixels from non-defect pixels) and then applied multiple crack detection filters to detect the presence of cracks in images captured by the Sewer Scanning and Evaluation Technology (SSET). SSET cameras produce cylindrical unrolled images of the pipe wall and are thus more uniform than those captured by front-facing CCTV cameras. However, front-facing CCTV is the primary technology used in inspection of sewers in the US and is hence the focus of this project (Halfawy and Hengmeechai 2014). Due to the high variability (i.e., size of defects, angle at which image is captured, illumination, etc.) in CCTV images, morphological and feature classification approaches are not effective in identifying defects. Deep learning-based methods, however, result in a significant improvement in accuracy over feature classification methods, on images which have high variability (LeCun et al. 2015). As a result, the two-step method proposed in this chapter leverages deep learning.

The first step involves using an anomaly identification system based on the ResNet34 CNN to determine whether images contain anomalies or not. The anomaly identification system is described in Section 5.2. This step does not provide the location information about the anomalies. In the second step, images that have been classified as anomalies are processed by a Faster R-CNN detector to localize the defects. The overall framework is illustrated in Figure 2.3. The ResNet34-based anomaly identification system consists of significantly fewer computations than the Faster R-CNN-based defect localization system and can thus process images significantly faster. Based on datasets of videos which we have collected, generally 20 percent of the entire duration of a CCTV sewer video depicts defects with the remaining 80 percent not depicting defects. The rationale behind adopting the pre-processing step is that it allows the Faster R-CNN based system to be used only on image frames depicting defects, thereby improving the overall speed of defect detection, when applied to videos.

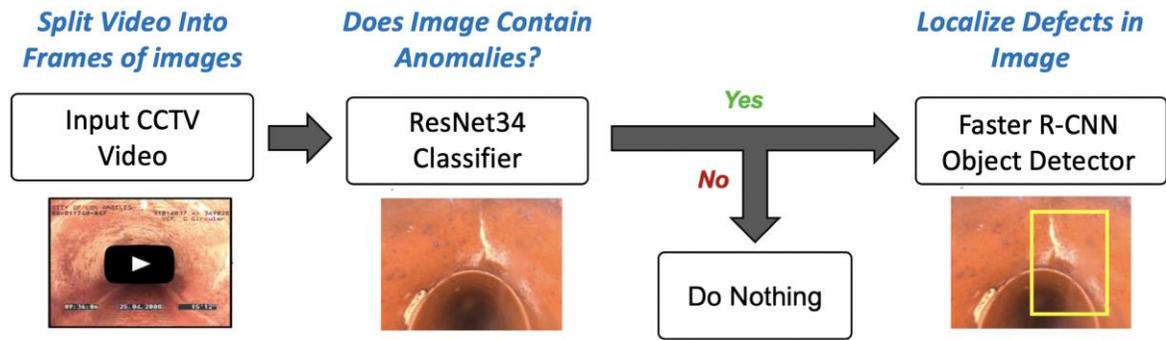


Figure 5.1 Illustration of the two-stage detection framework

5.2 Development of Anomaly Identification System

Anomaly identification seeks to identify which image frames in a sewer CCTV video contain anomalies or regions of interest. We define anomalies as cracks, fractures, broken pipes, joint offsets, root intrusions, deposits, and lateral connections. Sewer anomaly identification has been previously addressed through feature classification and morphological approaches by Moselhi and Shehab (2000), Sinha and Fieguth (2006a, 2006c), and Halfawy and Hengmeechai (2014). However, since deep learning-based image classification methods have been demonstrated to achieve significantly higher classification accuracies and generalization capabilities than feature classification approaches, we use CNNs for anomaly identification.

The anomaly identification system was developed based on the ResNet34 CNN, since this model has been demonstrated to achieve one of the highest image classification accuracies on benchmark datasets (He et al. 2015). The anomaly identification system takes color images as input and produces one out of two outputs: (1) ‘Anomaly’ if the system identifies an anomaly in an image and (2) ‘No Anomaly’ if the system is not able to identify an anomaly in an image. The ResNet34 CNN was trained using a dataset consisting of 12,000 images of anomalies and 12,000 images without anomalies. A separate dataset consisting of 3,000 images of sewer anomalies and 3,000 images without anomalies was used to test the performance of the system. Table 5.1 describes the dataset used for training and testing the automated anomaly identification system. The images originated from over 2,000 different sewer pipe inspections of 8-inch and 10-inch diameter VCPs in Florida, Georgia, and Ohio. All images were collected by front-facing CCTV cameras and had

a native resolution of 640×480. The images were provided by Hazen and Sawyer, and Hydromax USA.

Table 5.1 Description of the dataset used for training and testing the automated anomaly identification system

	Number of Images							
	Crack	Fracture	Broken	Joint Offset	Root	Deposit	Lateral	No Anomaly
Training	2000	2000	1000	1000	2000	2000	2000	12000
Testing	500	500	250	250	500	500	500	3000
Total	2500	2500	1250	1250	2500	2500	2500	15000

Deep neural networks such as ResNet34 are susceptible to overfitting and data augmentation is a common method to reduce overfitting. In data augmentation, the number of training images is considerably increased by applying label preserving transformations on images (Krizhevsky et al. 2012). In this study, the training dataset was augmented through horizontal flips (i.e., taking mirror images across a vertical axis) and image rotations. The augmentations boosted the number of images in the training dataset by a factor of 16, resulting in 192,000 images of anomalies and 192,000 images without anomalies. The trained ResNet34 CNN was evaluated on 6,000 images (i.e., 3,000 images of anomalies and 3,000 images without anomalies). The system yielded an accuracy of 98.4% on this dataset. After training and validation, the trained ResNet34 CNN was incorporated as the first step in the two-step framework.

5.3 Results of Testing on CCTV Videos

The two-step framework was tested on 10 videos of 8-inch diameter VCP which collectively represent 2200 feet of sewer mains. The framework is tested in the detection of cracks/fractures, root intrusions, and lateral connections in pipes. Cracks/fractures are selected for detection since they are important indicators of the structural health of sewers. Root intrusions are selected for detection because they have an adverse impact on the hydraulic capacity of a pipe and have the potential to progress into structural defects or exacerbate existing structural defects. Lateral connections are selected for detection because sewer defects (such as root intrusions and infiltration) are often concentrated around these locations. Furthermore, it is mandatory to report

the location of lateral connections in PACP reports. Note: At the current stage due to the high variability in panning, tilting, and zooming of the CCTV cameras, the system cannot distinguish between cracks and fractures. As a result, we have trained the detector to identify cracks and fractures as a single category, hereafter referred to as crack/fracture. The dataset used for testing the automated system consisted of 10 videos (see Table 5.2).

Table 5.2 Description of the video files used for testing the automated system

Video Id	Meta Data About the Pipes						
	Video Resolution	Location	Diameter (inches)	Pipe Material	Year Surveyed	Length (feet)	Presence of Grease
1	720×480	Alabama	8	VCP	2015	53	Absent
2	720×480	Alabama	8	VCP	2015	198	Present
3	720×480	Alabama	8	VCP	2015	154	Present
4	720×480	Alabama	8	VCP	2015	412	Present
5	720×480	Alabama	8	VCP	2015	390	Absent
6	640×480	Ohio	8	VCP	2016	252	Absent
7	640×480	Ohio	8	VCP	2016	205	Present
8	640×480	Ohio	8	VCP	2016	145	Present
9	640×480	Ohio	8	VCP	2016	174	Present
10	640×480	Ohio	8	VCP	2016	217	Absent

Note: All of the videos were recorded using pan, tilt, and zoom CCTV cameras

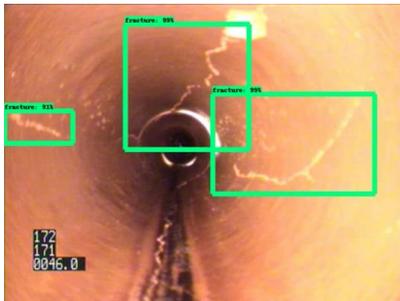
The results of the tests are shown in Table 5.3. The system correctly detected 112 out of 124 (i.e., 90.3%) instances of cracks/fractures, 88 out of 98 (i.e., 89.8%) instances of root intrusions, and 54 out of 59 (i.e., 92%) instances of lateral connections. Figure 5.2 shows examples of correct detections. The automated system also generated 45 false positive detections of cracks/fractures, 29 false positive detections of root intrusions, and 1 false positive detection of lateral connections. The system tends to misclassify pipe joints as cracks/fractures due to high visual similarities in their silhouettes (see Figure 5.3). The system also tended to misclassify fine roots as cracks/fractures due to high visual similarities in the two defects. A sample video of the automated system applied to CCTV videos can be accessed from: <https://youtu.be/JFiVZd489Fg>.

Table 5.3 Defect detection results

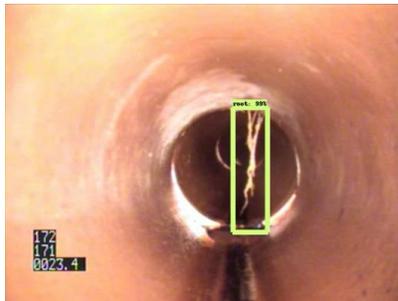
Video Id	Number of Defects			True Positive Detections			False Positive Detections		
	F	R	C	F	R	C	F	R	C
1	3	0	1	3	0	1	1	0	0
2	6	0	5	6	0	5	3	1	0
3	16	0	4	14	0	4	5	2	0
4	42	56	12	38	52	10	16	12	0
5	18	13	11	15	11	9	5	5	0
6	12	10	8	11	8	7	5	3	0
7	4	5	5	4	5	5	1	1	1
8	3	0	4	3	0	4	2	1	0
9	4	0	3	4	0	3	2	1	0
10	16	14	6	14	12	6	5	3	0
Total	124	98	59	112	88	54	45	29	1

Note: F – Fracture; R – Root Intrusion; C – Connection

a) Crack/Fracture



b) Root Intrusion

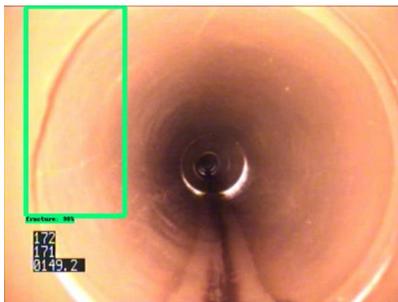


c) Lateral Connection



Figure 5.2 Examples of correct detections by the system

a) Joint Detected as Fracture



b) Root Detected as Fracture

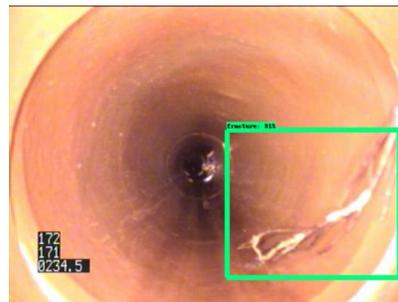


Figure 5.3 Examples of incorrect detections by the system

Since the automated system correctly detected approximately 90% of cracks/fractures and root intrusions, as well as 92% of the lateral connections, we believe that the tool can be used in practice to support off-site review of CCTV videos. The tool in its current form can significantly shorten the duration of CCTV video required to be watched by inspectors, thereby improving the speed of coding defects.

The tool has the potential to support automated PACP coding in the future if the number of false positives be significantly reduced. An iterative procedure of training and evaluation could assist in reducing the number of false positives. We propose the development of an iterative procedure to aggregate the correct and incorrect detections from tests on CCTV videos, in order to improve the training dataset. The procedure will provide a feedback mechanism in order to iteratively improve the accuracy of the system by updating the training dataset. Figure 5.4 shows a schematic diagram of the proposed procedure. Using the proposed procedure, incorrect detections (e.g., joints that are detected as fractures) could be flagged, extracted, and used for re-training the automated system. Hence, with continued use of the automated system, the percentage of incorrect detections can be significantly reduced.

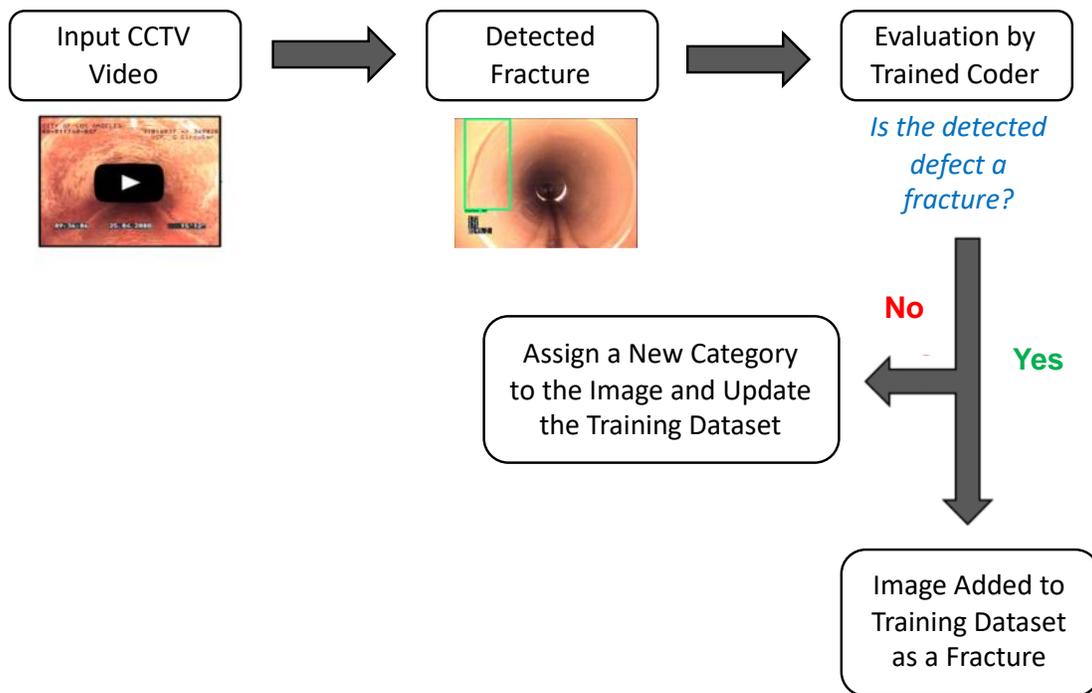


Figure 5.4 Schematic diagram of the iterative procedure to aggregate incorrect detections

5.4 Development of Supplementary Tool for Training Image Preparation

Deep learning-based automated defect detection algorithms typically require tens of thousands of images for training and testing. In this study, the research team curated a dataset of 30,000 sewer CCTV images for training and testing. As described in Section 5.2, the 30,000 images included 15,000 images of defects and 15,000 images of healthy pipe sections (i.e., images without any defects) and originated from over 4,000 sewer pipe inspections of 8-inch and 10-inch diameter vitrified clay pipes. During the initial stages of this research study, many of these images were tediously extracted from CCTV sewer inspection videos. However, we later developed a software tool called *ImgXtract* to streamline the process of image extraction from videos. This section describes the challenges faced in training image collection and provides an overview of *ImgXtract*.

Municipalities, contractors, and consultants typically use sewer inspection software such as *ITPipes*, *PipeTech*, and *WinCan*, to manage information from inspections. These software programs typically organize the information from inspections into the following databases:

- a. *Inspection videos*, which contain the inspection video files from a cohort of pipes,
- b. *Defect snapshots*, which contain images of defects that have been identified in the videos, and
- c. *Inspection reports*, which typically consist of Microsoft Excel or Microsoft Access databases that provide metadata about each inspection (e.g., location of the pipeline, pipe material, types of defects identified in a pipe, time(s) in the video when defects were identified, etc.).

We had planned to use images from the defect snapshots databases for development of the automated system. However, the following challenges were encountered when using these images for training the CNNs:

1. The defect snapshots generated by the software programs were typically overlaid with the defect labels (see Figure 5.5a). Upon using these images for training, it was discovered that the CNNs had learned to classify the labels, rather than learn discriminatory features about the defects. As a result, the CNNs tended to overfit the training data and were not able to generalize to images without the labels. The research team experimented with masking the labels,

however, the masks obscured important portions of the images and hindered the classification performance.

2. The snapshots were not always accurate representations of the defects (Figure 5.5b). Several images contained in the defect snapshots databases did not clearly show the defects and were thus unsuitable for use as training images.
3. The defect snapshots databases did not contain images of healthy pipe sections (i.e., images which do not contain any defects) which were needed to train the CNNs.

a) Original image



b) Image with mask



Figure 5.5 a) Original image with label; b) Image after addition of the mask to cover the label in the original image.

Note: the label indicates ‘Crack Longitudinal’ although the crack is not clearly visible in the image. Hence, this image would not be suitable for training the automated system.

The database of inspection videos contained the raw footage from pipe inspections and could be used to extract training images. However, the process of extracting images from videos requires significant manual effort. Initial experiments showed that the speed of image extraction was typically 15 images of unique defects per hour, and that 50 images of healthy pipe sections per hour could be extracted from the videos per hour.

ImgXtract was developed to improve the speed of image extraction from videos. Using ImgXtract, the speed of image extraction was determined to be approximately 300 images of unique defects

per hour and 500 images of healthy pipe sections per hour. The next section describes the functionality of the tool.

5.4.1 Description of ImgXtract

ImgXtract is a graphical image annotation tool written in Python and leverages the OpenCV library. The tool requires a database of sewer CCTV inspection videos, currently supporting videos with .mpg, .mp4, and .avi extensions, and the associated PACP database that contains the meta-data related to the inspection videos, currently supporting files with .csv extension. The tool operates in two modes: (1) defect image extraction and (2) healthy pipe section image extraction. Upon running the tool, the user is presented with a GUI to select the mode. Once the mode is selected, the user is requested to provide the path (i.e., the directory) where the video files are stored as well as the path to the PACP database.

In the defect image extraction mode, the tool searches the PACP database for the time(s) when defects have been identified in inspection videos and correlates this information with the video files. The tool then displays only the sections in the video where defects have been identified. To obtain the most representative image of a defect, the tool displays 9 images and allows the user to select the best image by entering a number from 1 to 9 into the computer's keyboard (see Figure 5.6). We believe that image selection using keyboard input is significantly faster than mouse clicks, since the necessity to hover the mouse pointer is avoided. However, in future versions of this tool, support for mouse clicks will also be provided.

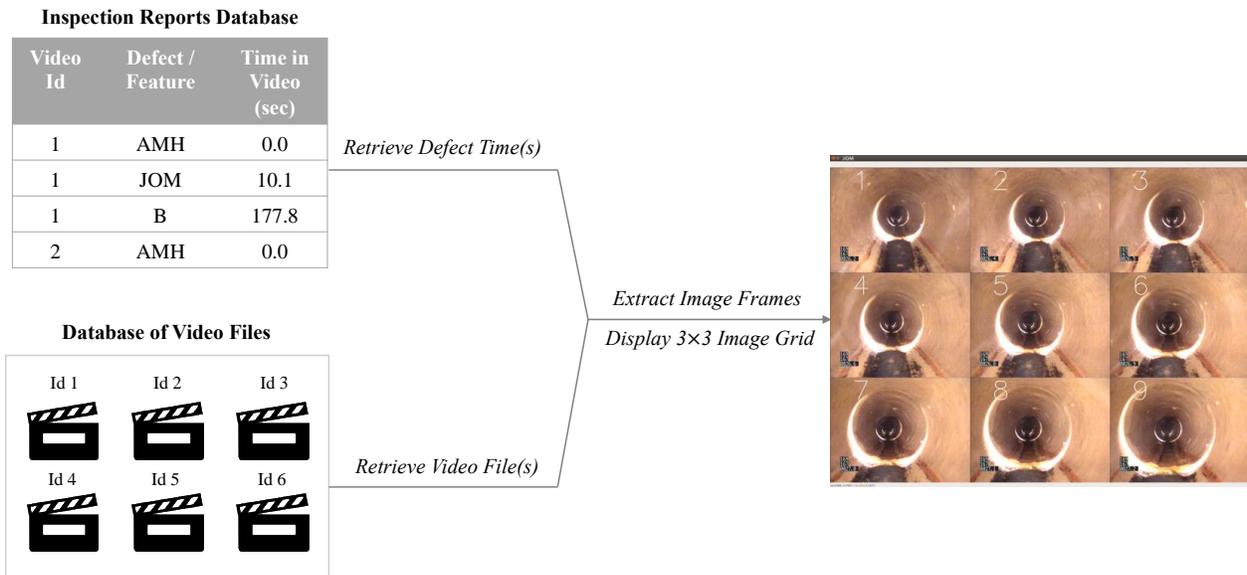
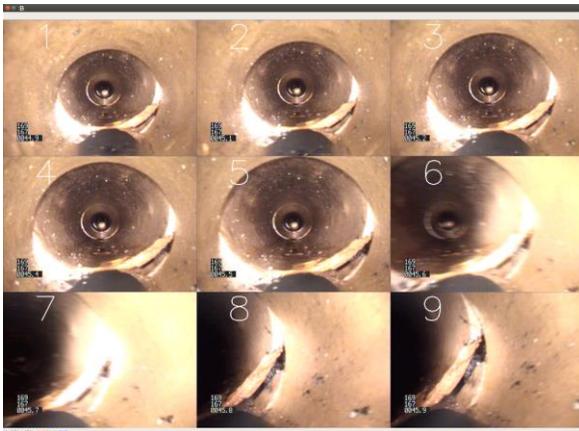


Figure 5.6 Schematic diagram of ImgXtract

The tool also provides options to traverse forward or backward in the video, to aid in the selection of the best image. Upon receiving the user input, the tool saves the selected image in a folder based on the type of defect (see Figure 5.7). The tool uses the NASSCO PACP convention of naming defects. In the healthy pipe section image extraction mode, the tool extracts image frames at 10 second intervals and displays the images as a grid of 9 images. The tool ensures that images containing defects are not displayed to the user. Once an image is selected, the image is saved to the 'NoDefect' folder.

a) Grid of 9 images displayed by the tool



b) Directory structure used to store defect images



Figure 5.7 a) ImgXtract displays a grid of nine images to the user; b) The tool automatically stores images in folders based on the defect type.

Note: the grid of nine images allows the user to select the most suitable image for training the automated system. The user selects the best image by using the keyboard to enter a number from '1' to '9'. For example, if the user enters the number '8', then the eighth image in the grid is saved to the folder.

CHAPTER 6. CNN INTERPRETATION TECHNIQUES

[A version of this chapter was published in the proceedings of the 2020 ASCE Construction Research Congress]⁴

Although CNNs outperform conventional feature engineering methods on image classification tasks, the large number of parameters and complex interconnections in these models leads to a lack of interpretability and CNNs being used as ‘black boxes’. The lack of interpretation capabilities results in the loss of generalization capability, i.e., the algorithms may produce unexpected results when exposed to edge cases. This issue is prevalent in CNNs used for automated sewer defect identification since the training and testing images differ significantly. For example, municipalities typically maintain large numbers (i.e., several thousands) of labeled sewer defect images as an outcome of their sewer inspection programs; however, these images typically contain the defect labels imprinted on the images. Cropping the labels out or obscuring them is manually cumbersome and typically results in loss of information from the images. Hence, CNNs trained with these labeled images may exhibit unusual behavior, such as being sensitive to defect labels and other markers in the images and may result in significantly lower classification accuracies when tested on unlabeled images.

The focus of this chapter is on leveraging CNN interpretation techniques to improve the generalization capability of automated defect identification models used in sewer CCTV video interpretation. We define generalization capability of automated defect identification models as the ability to generalize to images that differ significantly from the images used for training. The contributions of this study are as follows. (a) A CNN interpretation technique called class activation mapping (CAM) is used as a diagnostic tool to interpret the generalization capability of CNNs in sewer defect identification. (b) We demonstrate that the insights gained from CAM enables the development of more accurate sewer defect classification models.

⁴ Kumar, S. S. and Abraham, D. M. (2020). Leveraging Visualization Techniques to Develop Improved Deep Neural Network Architectures for Sewer Defect Identification, ASCE Construction Research Congress (CRC 2020), Tempe, Arizona, USA.

6.1 CNN Feature Visualization

Various approaches for visualizing the learned features in CNNs have been developed in literature as a response to their lack of interpretability. In this section, we briefly survey some of these approaches and related work.

A common visualization technique involves showing the activations of the network during the forward pass. For untrained CNNs, these activations usually start out looking relatively dense. However, as training progresses the activations appear sparser and more localized. A drawback of this technique is that the layer activations are complicated and require significant expertise to interpret. Furthermore, given the large number of layer activations, this technique tends to be manually cumbersome. Another CNN interpretation strategy involves visualizing the learned weights. These weights are usually most interpretable in the first few convolutional layers that take raw pixel data as inputs. These weights are useful to visualize because well-trained CNNs usually display smooth filters with few noisy patterns. Noisy patterns can be indicative of CNNs that have not been trained sufficiently enough or exhibit low regularization strength. This technique suffers from the same issue as the previous method in that the weights require significant effort and expertise to interpret. Furthermore, the insights that these techniques provide, such as under-training and overfitting, can also be detected by comparing the training, validation, and testing accuracies of CNNs.

CAM is a technique which was developed by Zhou et al. (2016) that can be applied to CNN architectures that use a global average pooling layer (e.g., AlexNet, ResNet, VGGNet, etc.). CAM takes a trained CNN and a sample image as input and finds the discriminative regions in the sample image which results in a particular output. CAM takes into consideration the weights in the fully connected layers of a CNN, determines the features which contribute to a particular classification (e.g., root), and project these features back onto the input images in the form of heatmaps (see Figure 6.1). For a CNN that is trained to classify images as either belonging to the root, fracture, or lateral category, CAM can be applied to a sample image to determine the regions in the image that lead it to be classified as a root or classified as a fracture (see Figure 6.1). Note: The red regions in Figure 6.1 represent pixels in the image that the CNN gives maximum importance to.

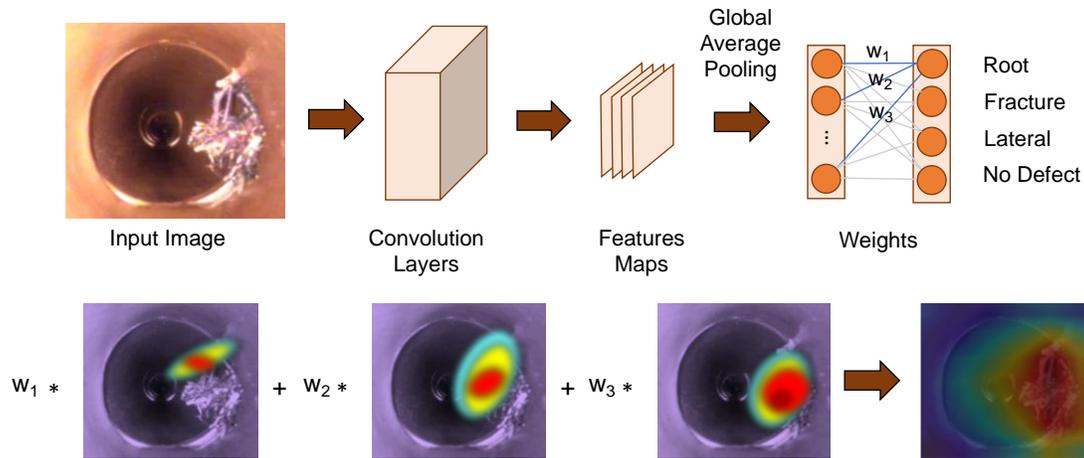


Figure 6.1 CAM applied to an example CCTV image.

The results of CAM can be visualized as a heatmap and thus provide an intuitive explanation of what the CNN learns to recognize. The limitation of this method is that it can only be applied to CNNs with a global average pooling layer and is hence not directly applicable to object detection architectures such as Faster R-CNN, SSD, and YOLO. However, since object detection architectures also use CNNs as feature extractors, this method can provide an indirect assessment of object detection methods. In this study, we consider CAM as a CNN interpretation technique due to its highly intuitive visualizations.

6.2 Experiments and Discussions

The datasets used for training and testing were created to facilitate the measurement of the generalization capability of CNNs. The training dataset consisted of images that had previously been manually coded by municipalities. Imprinted on these images were the defect labels (see Figure 6.2). Municipalities typically maintain large volumes of such images in their databases as a result of their sewer inspection programs. The test dataset, however, consisted of images that were manually extracted from un-coded CCTV videos. Since these images were manually extracted from the videos, they did not contain defect labels (see Figure 6.2). Note: The images used in the training datasets have labels imprinted (on the bottom left of the image), whereas the images used for testing do not have labels imprinted on the images. Experiments were then conducted to evaluate whether the CNNs trained using the images with labels would be able to

generalize to images without the labels imprinted on them. Such an experimental setup simulates the real-world scenario where large volumes of archived images are readily available for training automated defect identification algorithms. However, the trained models are intended to be applied to images that may differ significantly from the images used for training.

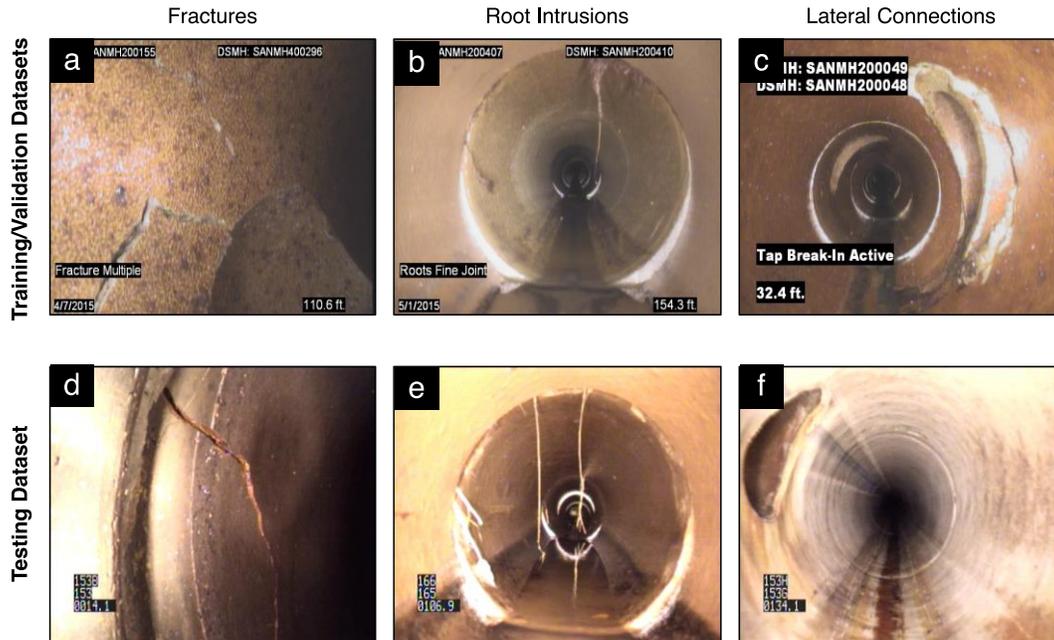


Figure 6.2 Examples of images used for training and testing

Three categories of sewer defects/features, i.e., fractures, root intrusions, and lateral connections were considered for automated identification. Fractures are considered as structural defects and are crucial for evaluating the health and remaining service life of pipes, whereas root intrusions can expand existing openings in sewers, further weakening the pipe, and ultimately causing breakage and collapse of the pipe. Lateral connections are regions where drainage and plumbing from homes and offices connect to the main line. These defects were considered in this study because of their typically high frequency of occurrence in sewer pipes. These are regions of stress concentration and are often regions of infiltration. In addition to images of defects/features, images of healthy pipe sections were also included in the training and test datasets.

Table 6.1 lists the number of images from each defect/feature category that were used for training, validation, and testing. In total, 14,400 images were used for training and testing. The training images originated from 8-inch diameter vitrified clay pipes in Ohio, whereas the images used for testing originated from 8-inch diameter vitrified clay pipes in South Carolina. The number of images of healthy pipe sections was three times that of either defect category. A larger number of healthy pipe section images was incorporated into the datasets to account for the observation made by Meijer et al. (2019) that images of healthy pipe sections far outnumber images of defects, leading to biased CNNs if fewer defect images are incorporated into the training datasets. These images had already been classified by trained human inspectors and we consider these human-labeled images as ground-truths.

Table 6.1 Number of images in each category used for training, validation, and testing

Defect Category	Training	Validation	Testing
Fractures	1600	400	400
Root Intrusions	1600	400	400
Lateral Connection	1600	400	400
Healthy Pipe Sections	4800	1200	1200
Total	9600	2400	2400

In this chapter, a single CNN architecture (i.e., the ResNet34) was used for training and testing. ResNet34, developed by He et al. (2015), was selected since it has been shown to achieve state-of-the-art image classification accuracies on benchmark datasets, while being relatively fast (ResNet34 has an error rate of 21% compared to VGG-Net, which has an error rate of 24% on the ImageNet dataset. Our tests also showed that ResNet34 could process images at a rate of 40 frames per second on a computer with a Nvidia GTX1070Ti graphics card). However, multiple data augmentation hyperparameters were incorporated to improve the classification accuracy of the trained model. Data augmentation is a common method used to improve the classification accuracy of CNNs and involves applying label preserving transformations on the training images. By implementing data augmentations on the training images, CNNs are exposed to a larger set of images with geometric variations

First, four ResNet34 models were trained with various combinations of data augmentations, including horizontal flips, vertical flips, and centre crops as (see Table 6.2). Each model was trained until the validation accuracy reached a maximum value and decreased upon further training. This method of using the validation accuracy as an indicator of the best model is typical practice in training sewer defect identification models (Halfawy and Hengmeechai 2014, Cheng and Wang 2018, Kumar et al. 2018, Meijer et al. 2019). Model 3, i.e., the model which was trained with images augmented by horizontal and vertical flips was found to be the best model, with a validation accuracy of 94.8%. The precision (i.e., ratio of true positives to the sum of true positives and false positives) was 93.6% and the recall (i.e., ratio of true positives to the sum of true positives and false negatives) was 96.16%. Note: Since human-identified labels were considered as ground-truths for training and testing, an accuracy of 100% would indicate that the model’s classification accuracy is equivalent to that of a trained human inspector.

Table 6.2 Data augmentation options considered

Model No.	Data Augmentations	Model Accuracy Validation Set
1	No data augmentations	90.3%
2	Horizontal flips	93.6%
3	Horizontal flips, Vertical flips	94.8%
4	Center cropping, Horizontal flips, Vertical flips	92.6%

Note: Accuracy is defined as the ratio of correctly classified images to the total number of images

CAM was then used to interpret what Model 3 had learned. The CAM outputs on sample validation images of fractures, root intrusions, and lateral connections are shown in Figure 6.3. It can be seen that the CNN was sensitive to the presence of image labels, i.e., the labels were used to bolster the predictions. The sensitivity of the CNN to the defect pixels was low as shown in Figure 3d and Figure 3e. We hypothesize that the CNNs learned to recognize the labels rather than defect pixels because of the distinct and localized nature of the labels. That is, the labels were always located in approximately the same region of the image. The defect pixels on the other hand, were not confined to a single location. In other words, the labels represent ‘low hanging fruit’, and CNNs are able to learn to recognize the labels much faster than learning to recognize the defect pixels. However, unlike images of fractures and root intrusions, lateral connections are usually confined to a few locations since they are constructed based on standardized designs. As seen in Figure 6.3f the CNN

was able to prioritize the pixels of the lateral connection instead of the image labels. From the CAM outputs it is evident that the CNN would not generalize well to images of fractures and root intrusions in the test set. It was found that the accuracy on the test set was approximately 15% lower than the accuracy on the validation set (i.e., the validation and test accuracies were 94.8% and 79.9%, respectively, see Table 6.3). The precision and recall on the test set were 90.2% and 67%, respectively, indicating that a significant number of false negatives (i.e., missed defects). The missed defects were presumably due to the absence of labels imprinted on the test images. Hence, it can be concluded that the CNN learned to ‘cheat’ during training, by leveraging statistical similarities in the images rather than learning the distinguishing features of the defects.

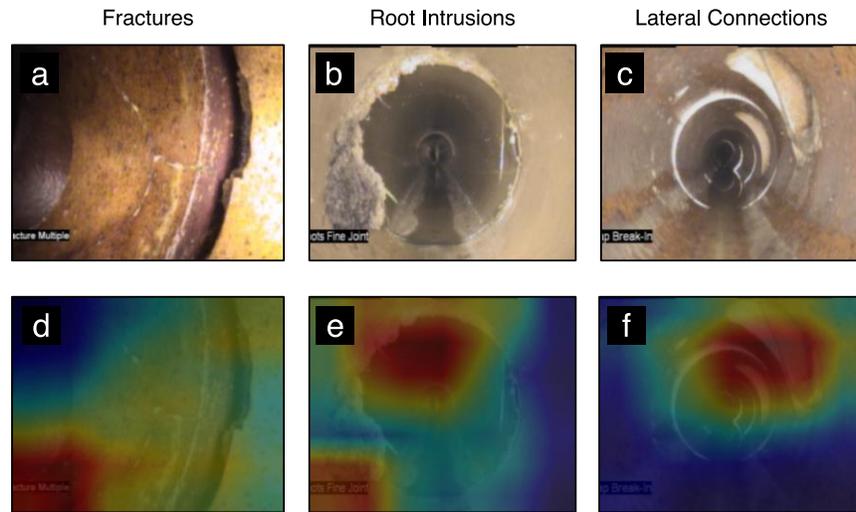


Figure 6.3 CAM outputs for model 3 on validation images.

In order to mitigate this behaviour of learning the labels, we implemented a data augmentation technique which included inducing random rotations of less than or equal to 90 degrees in the images. We believe that by randomly rotating the images, the labels would not always be confined to the same areas and would increase the difficulty of learning the labels. Figure 6.4 compares the CAM outputs of model 3 trained with and without the incorporation of rotations. The heatmaps produced by CAM show that augmenting the training data with rotations, results in the model giving a higher importance to defect pixels rather than on the labels. However, an interesting observation is that the validation accuracy of this model decreased by 1.5% (see Table 6.3). We believe that the decrease in validation accuracy is because the model trained with rotations did not

learn to recognize the defect labels and rather focused on the defect pixels alone. Hence, the model was not able to ‘cheat’ on the test resulting in a lower classification accuracy. However, the accuracy on the test set of this model was 90.5%, which is comparable to the same model’s validation accuracy (93.3% as shown in Table 6.3) and significantly higher than the test accuracy of the model without rotations (79.9% as shown in Table 6.3). Furthermore, the recall on the test set, of the model with rotations was 90.0%, which is significantly higher than that of the model without rotations (67.0%) indicating significantly fewer false negative detections (i.e., missed defects). The higher recall indicates that the model was able to correctly classify defects in the absence of imprinted labels. The significantly higher accuracy on the test set and insignificant decrease from the validation accuracy indicates a higher generalization capability of the model with rotations.

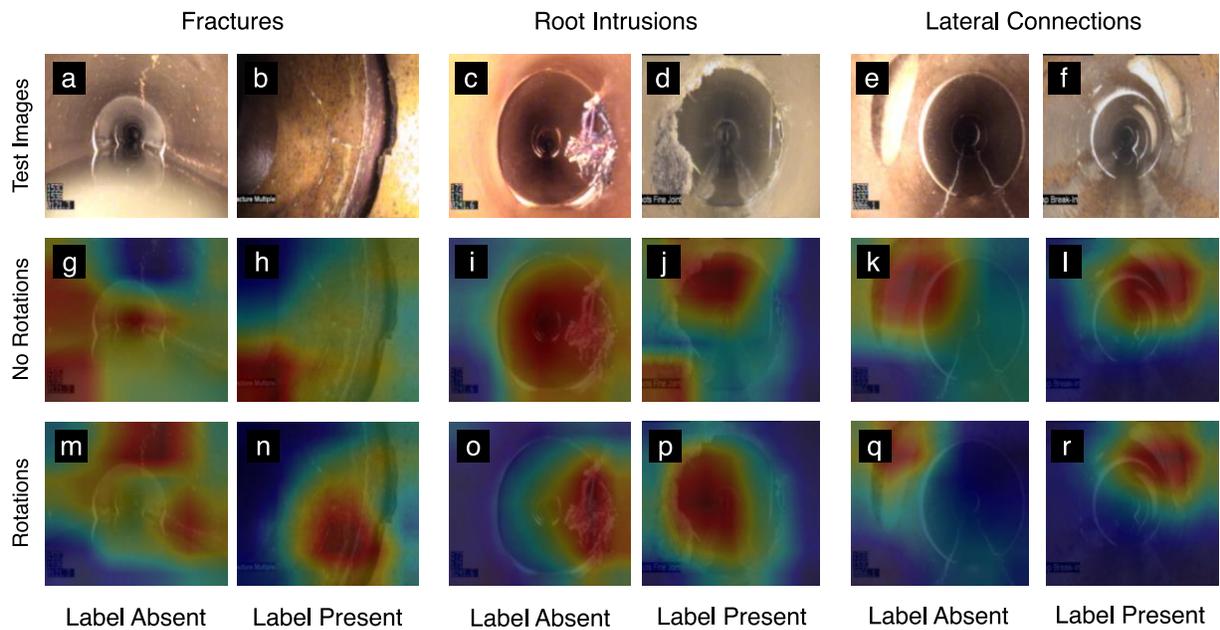


Figure 6.4 Example CAM of model 3 with and without the incorporation of rotations as a data augmentation technique

Table 6.3 Data augmentation options considered

Data Augmentations	Model Accuracy					
	Validation Set			Test Set		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Horizontal flips, Vertical flips	94.8%	93.6%	96.2%	79.9%	90.2%	67.0%
Horizontal flips, Vertical flips, Rotations	93.3 %	93.2%	93.4%	90.5%	90.9%	90.0%

6.3 Chapter Summary

Deep learning-based methods such as CNNs are increasingly being used in lieu of feature engineering techniques in automated sewer CCTV inspection studies. However, the large number of parameters and complex interconnections in CNNs contributes to their lack of interpretability, which may result in a loss of generalization capability (i.e., CNNs may produce unexpected results when exposed to edge cases). This issue is significant in CNNs used for automated sewer defect identification, since the images used for training typically differ from the images used for testing. For instance, municipalities possess several thousands of labeled sewer defect images which are suitable for training deep learning models. However, these images typically contain defect labels and other text data imprinted on them. CNNs trained with these pre-labeled images tend to exhibit unusual behavior, such as being sensitive to defect labels and other markers in the images, leading to lower classification accuracies when tested with unlabeled images. In this study, CAM, which is a CNN interpretation technique, is used to guide the development of sewer defect identification models that have a higher generalization capability.

Our experimental setup consists of training CNNs using 12,000 images pre-labeled images (i.e., images with labels imprinted on them) and evaluating the accuracies of the trained models using a test set consisting of 2,400 unlabeled images. An analysis of the heatmaps produced by CAM indicated that CNNs trained using the pre-labeled images were highly sensitive to the image labels. We believe that the CNNs learned to recognize the labels rather than defect pixels because of the distinct and localized nature of the labels, (i.e., the labels were always located in approximately the same region of the image). In order to decrease this tendency of learning the labels, we implemented a data augmentation technique which included inducing random rotations in the

training images. We hypothesized that by incorporating random rotations in the images, the labels would not be confined to the same locations in the images, resulting in an increased difficulty for the CNNs to learn the labels. CAM showed us that the CNNs trained using images augmented with rotations gave much higher importance to the defect pixels and disregarded the image labels. The model trained with rotated images also achieved a significantly higher classification accuracy on the test set.

Hence, this study represents a shift towards using CNN interpretation techniques to develop sewer defect identification models with better generalization capabilities. A limitation of this study is that the CNN interpretation technique used (i.e., CAM) cannot be directly applied to object detection models, due to their lack of a global average pooling layer. However, various methods such as image occlusion sensitivity exist for interpreting object detection models. Our future research will investigate how CNN interpretation techniques can be combined with CNN pruning to develop robust and computationally efficient architectures for automated sewer defect identification.

CHAPTER 7. VISION-BASED ORIENTATION ESTIMATION OF CCTV ROBOTS TO SUPPORT DEFECT LOCALIZATION AND AUTONOMOUS NAVIGATION

In the US, closed-circuit television (CCTV) is the primary method used by municipalities for inspecting non man-entry sewers (Halfawy and Hengmeechai 2014c). This method of inspection involves televising the inner surfaces of pipes using a CCTV camera equipped robot crawler. Trained operators review the video-feed transmitted by the CCTV cameras in order to identify defects in the pipe. The process of reviewing inspection videos to identify defects is called defect coding and most municipalities in the US adopt the NASSCO PACP convention of coding defects. According to the NASSCO PACP convention, defects must be classified according to their type (e.g., crack, fracture, hole, root, etc.), and their locations in the pipe must be identified. Due to the reliance on manual interpretation of videos, sewer CCTV inspections have the propensity to be slow and inconsistent. Dirksen et al. (2013) conducted a study, where sewer CCTV images from 60 sewers were shown to six trained operators, and their results compared. The study found that on average the trained operators failed to report approximately 25% of the defects. Automated sewer CCTV inspections have the potential to address the limitations of manual inspections and can improve the consistency, accuracy, and speed of sewer condition assessment. Hence multiple research studies have sought to develop methodologies to automate various components of the sewer CCTV inspection process. The next section summarizes the contributions of these studies.

7.1 Related Studies

Research on automated sewer CCTV inspections has addressed the following two themes: (1) automated defect interpretation, which attempts to replicate the manual process of identifying defects in videos and (2) autonomous robot navigation, which attempts to automate the control and navigation of inspection robots in sewers.

7.1.1 Automated Defect Interpretation

Automated defect interpretation involves using computer vision techniques to process inspection videos and perform the following three tasks: 1) defect classification, i.e., assigning defect labels

to defects (e.g., root medium joint, fracture longitudinal, joint offset medium, etc.), 2) longitudinal localization, i.e., reporting the distance along the pipe where a defect is located, and 3) circumferential localization, i.e., reporting the position of a defect relative to the cross-section of a pipe (see Figure 7.1).

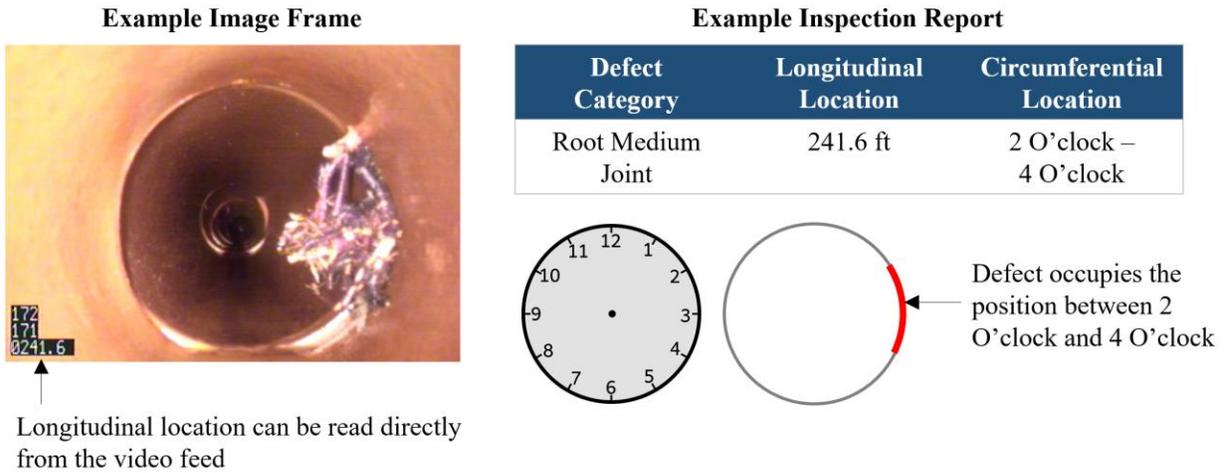


Figure 7.1 Example to illustrate the attributes of defects that are identified during sewer CCTV inspections

Prior research work on automated defect interpretation, has addressed defect classification and longitudinal localization (refer to Section 2.1.1 for details about the methodologies, contributions, and limitations of these research studies). Circumferential localization of defects, however, is significantly more challenging and has been addressed by few studies in this domain. Research in the area of defect circumferential localization, is currently limited to the identification of the height, width, and position (i.e., in pixel coordinates) of defects in images. Cheng and Wang (2018) used the Faster R-CNN model to detect cracks, roots, and deposits in CCTV images whereas Kumar et al. (2019) developed a two-step CNN framework to detect fractures, roots, and lateral connections in CCTV videos. The approaches presented by Cheng and Wang (2018) and Kumar et al. (2019) process CCTV images and produce bounding boxes to identify the height, width, and position of defects in images (refer to Chapter 4 for details about the methodologies, contributions, and limitations of these two studies). However, the bounding boxes alone are not enough to deduce the circumferential location of defects, since the images are not referenced to regions in the pipe. For

instance, if the camera is oriented straight down the pipe, then the resulting image frame is a forward view frame, such as shown in Figure 7.2a. If the camera is oriented left off the center or right off the center, then the resulting image frames resemble Figure 7.2b and Figure 7.2c, respectively. Estimating the orientation of a camera, aids in referencing the images to regions of the pipe, which facilitates circumferential localization. Since CCTV robots do not possess special equipment to determine the camera orientation, a vision-based technique to estimate camera orientation would enable circumferential localization of defects.

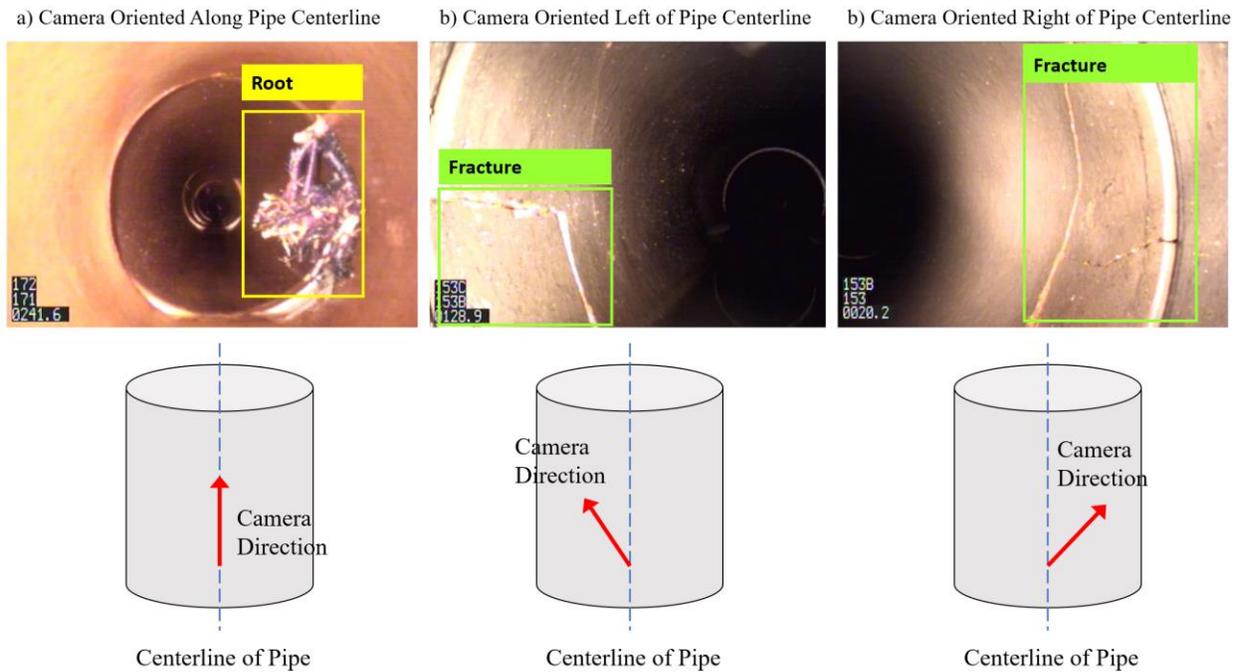


Figure 7.2 Example to illustrate how camera orientation affects the appearance of images

7.1.2 Autonomous Navigation of Sewer Robots

Autonomous navigation of sewer CCTV inspection crawlers is an area of research which seeks to develop algorithms that can enable CCTV inspection robots to traverse sewer pipes without human control. For an inspection robot to navigate autonomously in a sewer pipe, the robot must be able to determine its position and orientation relative to the pipe. For instance, if the robot is aligned parallel to the centerline of the pipe as shown in Figure 2a, it should continue traveling forward. However, if the robot is oriented left off the centerline (see Figure 2b) or right off the centerline (see Figure 2c), it should steer right and left, respectively.

A considerable body of research exists on the development of autonomous navigation systems for sewer robots. However, most studies use depth cameras, lasers, LiDAR, or orientation sensors to determine the location and orientation of robots in pipes—sensors which are not typically found on sewer CCTV inspection robots. Sewer CCTV robots generally contain a single RGB camera, as opposed to an array of sensing devices. Ahrary et al. (2007) developed an autonomous mobile robot system called KANTARO, which was demonstrated to successfully navigate through dry concrete pipes of diameters ranging between 8 inches and 12 inches. However, their study used multiple sensing devices such as RGB cameras, 2D lasers, and infrared sensors to determine the location of robots in pipes—whereas typical CCTV inspection robot only contain a single RGB camera. Hence, the method proposed by Ahrary et al. (2007) cannot be directly applied to CCTV inspection robots. Nassiraei et al. (2010) developed a robot localization system to enable autonomous sewer inspections of pipes with diameters ranging from 8 inches to 12 inches. Their method uses a robot that is equipped with passive arms on either side (i.e., left and right). During inspections, these arms brush against the walls of the pipe, and the angle made by the arms is used to determine the robot's orientation. However, adapting this method for sewer CCTV inspections would entail retrofitting of existing robots with passive arms, which could have various practical limitations. Lee et al. (2011) developed an autonomous navigation system for sewer inspection robots using a pathfinding concept called landmark recognition. Their study used a custom robot that was equipped with line lasers to identify the locations of landmarks, such as elbow joints and branches in a pipe. Additionally, a 3D orientation sensor was used to determine the robot's orientation with respect to the landmarks and to calculate the inspection trajectory. Meeks (2016) attempted to develop an autonomous robot for the inspection of storm sewers. Their robot contained an RGB camera and LiDAR scanner to visually inspect sewers, and a GPS device to estimate the robot's position and orientation. However, since underground pipes are GPS-denied environments Meeks (2016) evaluated the robot's autonomous navigation capabilities on open pipes, rather than in underground sewers. To enable autonomous navigation in underground sewers, their study recommended the development of methods to determine a robot's position without the use of GPS. Alejo et al. (2019) and Alejo et al. (2020) developed a framework to determine the position of robots in sewers, in the absence of GPS. Their framework utilized robot odometry and GIS information about the location of sewer manholes to determine the position of robots in pipes. However, the methods proposed by Alejo et al. (2019) and Alejo et al. (2020) utilize 7 RGBD

cameras, in order to create a 3D map of the sewer, whereas most municipalities in the US use CCTV robots with a single RGB camera. Hence, their method cannot be directly applied to CCTV inspection robots either.

In order to facilitate autonomous inspections using sewer CCTV robots, there is a need for methods that use information from a single camera sensor to determine the (1) position and (2) orientation of robots in pipes. Halfawy and Hengmeechai (2014c) partially addressed this need by developing a vision-based method to determine the longitudinal position of sewer CCTV robots. This study extends upon work proposed by Halfawy and Hengmeechai (2014c) by developing a vision-based method for estimating the orientation of sewer CCTV robots in pipes.

7.1.3 Contributions of this Study

The proposed method takes a CCTV video as input and estimates the orientation of the sewer CCTV camera in every image frame of the video. The estimated camera orientation, which is an output of this method, could enable automated circumferential localization of sewer defects, which is an unaddressed problem in literature. Additionally, the proposed methodology for vision-based estimation of camera orientation could also benefit autonomous sewer navigation, by enabling inspection robots to estimate their trajectories and take corrective actions while traversing pipes. Although the proposed method is developed for CCTV inspections that use robot crawlers, the approach is adaptable to other inspection platforms such as unmanned aerial vehicles (UAVs) and to pipes of different sizes and materials. Hence, an anticipated outcome of this study is the development of autonomous inspection platforms that leverage UAVs or other inspection technologies for rapid condition assessment of sewer pipe networks.

7.2 Methodology

The proposed framework uses visual markers in sewer images to infer the orientation of a CCTV camera in a pipe. Specifically, the position of the vanishing point of a pipe is used to characterize the orientation of a camera relative to the center line of a pipe. This study defines the vanishing point of a pipe as the point at which the sewer walls appear to converge in images (see Figure 7.3). In order to detect vanishing points in images, the study uses a deep learning-based object detection

model called the Single Shot Detector (SSD). The SSD was selected for vanishing point detection because of its relatively low computational complexity (in comparison to other deep learning-based object detection models such as the Faster R-CNN), which enables the video-feed to be processed in real-time, i.e., a processing speed of 30 image frames per second or higher, facilitating autonomous navigation (refer to Section 7.2.2.2 for further details about the processing speed). The camera orientation is then estimated based on the position of the detected vanishing points in images.

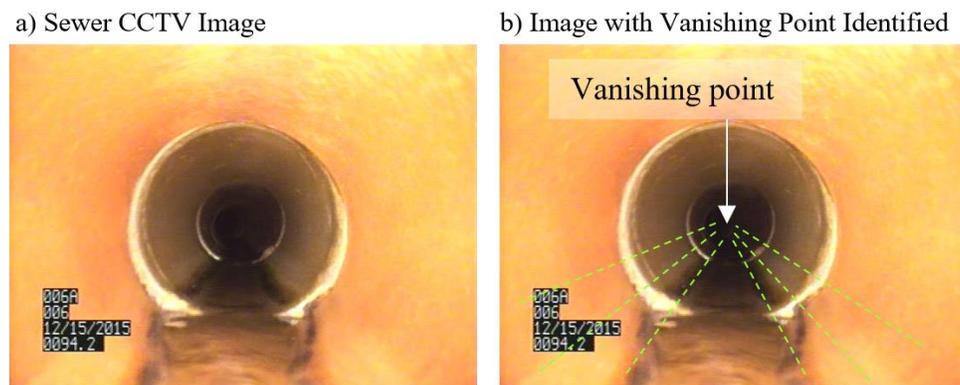
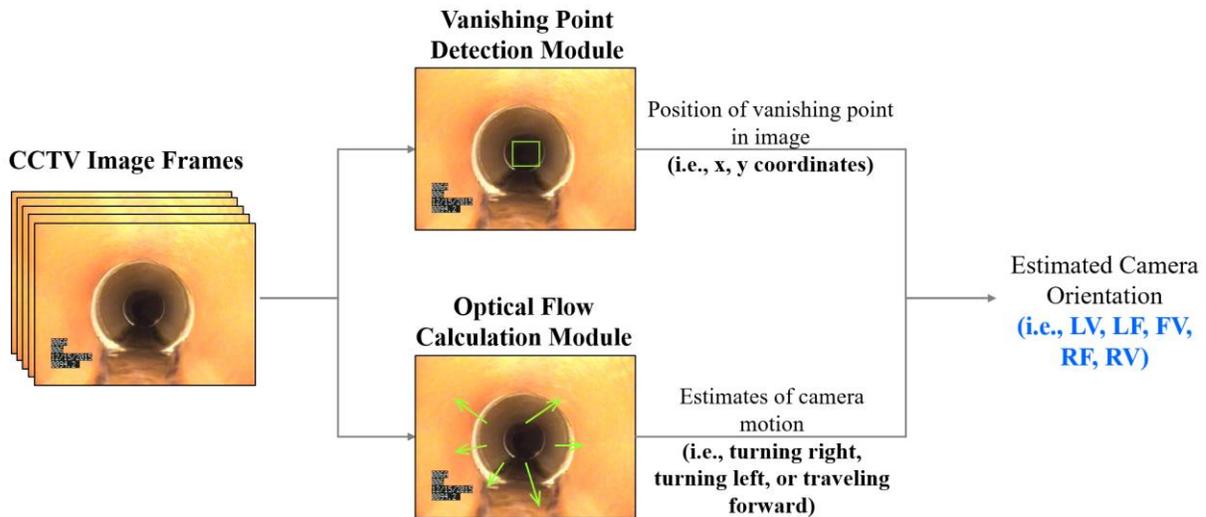


Figure 7.3 Illustration of sewer pipe vanishing point

However, due to large variations in images due to variable camera orientations, illumination conditions, motion blur, gas buildup, etc., the vanishing point may either not be visible in images or the SSD model fails to detect its presence. Hence, a technique called optical flow is used to infer the orientation of the camera when vanishing points are not detected. The conceptual framework consists of two modules: (1) Vanishing Point Detection Module and (2) Optical Flow Calculation Module (see Figure 7.4).



LV – Left-Wall View; LF – Left-Wall/Forward View; FV – Forward View, RF – Right-Wall/Forward View; RV – Right-Wall View

Figure 7.4 Conceptual framework for vision-based estimation of camera orientation

7.2.1 Image Frame Categorization

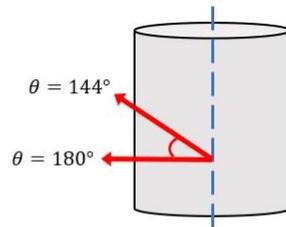
A major challenge in estimating the camera orientation is the lack of knowledge about the intrinsic parameters (i.e., focal length, lens distortion, and principal point) of the sewer CCTV cameras. The intrinsic parameters affect the field of view seen through the lens of the camera and cause distortions in the scale of the images. Without information about these intrinsic parameters, it is impossible to measure the angle made by the camera with the pipe centerline. Additionally, the intrinsic parameters of sewer CCTV cameras vary greatly based on the manufacturer, type of camera, and model year. In order to circumvent this problem, the approach presented in this study focuses on classifying the orientation of a camera into distinct classes, rather on measuring the angle made by the camera. This approach allows us to develop a framework that can be broadly used across multiple camera types. A similar approach has been proposed in a study by Halfawy and Henmeechai (2014), where the orientation of cameras is classified into two classes: (1) forward view, where the camera direction is parallel to the centerline of the pipe and (2) wall view, where the camera orientation is perpendicular to the centerline of the pipe. However, CCTV inspections typically consist of images captured at various camera orientations, which cannot be accounted for by the method proposed by Halfawy and Hengmeechai (2014c).

In order to account for the variations in camera orientation, this study groups camera orientations into the following five classes:

- (1) Left-wall view (LV), which represents a situation where the camera is oriented towards the left wall of the pipe (see Figure 7.4a). The vanishing point of the pipe is not visible in image frames captured at this orientation.
- (2) Left wall/forward view (LF), which represents a situation where the camera is oriented between the left wall and pipe centerline (see Figure 7.4b). The vanishing point as well as the left wall are partially visible in image frames captured at this orientation.
- (3) Forward view (FV), which represents a situation where the camera is oriented along the pipe centerline (see Figure 7.4c). The vanishing point is clearly visible and approximately in the center of these image frames.
- (4) Right-wall/forward view (RF), which represents a situation where the camera is oriented between the right wall and pipe centerline (see Figure 7.4d). The vanishing point as well as the right wall are partially visible in these image frames.
- (5) Right-wall view (RV), which represents a situation where the camera is oriented towards the right wall (see Figure 7.4e). The vanishing point of the pipe is not visible in image frames captured at this orientation.

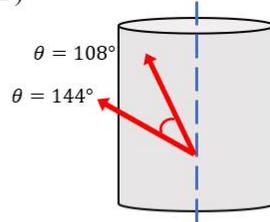
The images corresponding to different camera orientation classes bear visual similarities. For example, LV images are similar to LF images, in that the left wall of the pipe is typically visible in both sets of images. Images corresponding to LF, FV, and RF classes are similar in that the vanishing point is typically visible in all three images. RV images and RF images are similar in that the right wall of the pipe is typically visible in both sets of images. To account for these similarities, this study introduces the concept of adjacent classes and defines three sets of adjacent classes: {LV, LF}, {LF, FV, RF}, and {RV, RF}.

a) Left-wall view (LV)



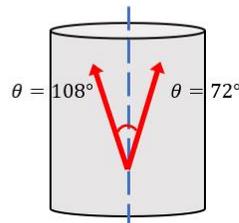
- Camera oriented towards the left wall
- Vanishing point of pipe **not visible**

b) Left-wall/forward view (LF)



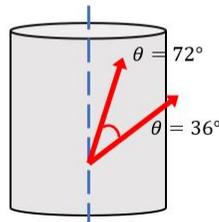
- Camera oriented between left wall and pipe centerline
- Vanishing point of pipe **partially visible**

c) Forward view (FV)



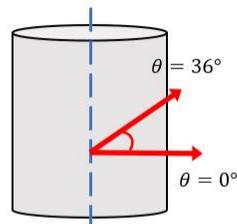
- Camera oriented along the pipe centerline
- Vanishing point of pipe **clearly visible**

d) Right-wall/forward view (RF)



- Camera oriented between right wall and pipe centerline
- Vanishing point of pipe **partially visible**

e) Right-wall view (RV)



- Camera oriented towards the right wall
- Vanishing point of pipe **not visible**

θ – Angle (measured counterclockwise) between the camera and the pipe centerline's normal

Figure 7.4 Five camera orientations addressed by the methodology

The next sections describe the Vanishing Point Detection Module and the Optical Flow Calculation Module.

7.2.2 Vanishing Point Detection Module

This module identifies the position of vanishing points in images and uses an SSD MobileNets model that is trained using 1000 annotated sewer images to facilitate the detection of vanishing points in images.

SSD MobileNets

The architecture of SSD MobileNets model used for vanishing point detection is similar to the model described in Section 4.1.1 and performs object classification and localization in a single forward pass of the network. However, unlike the model described in Section 4.1.1, the output of the model in this section consists of two object classes (i.e., vanishing point and background). Since the SSD produces 8,732 detections per object class, the total number of detections per image is 17,464 (i.e., $8,732 \times 2$). The SSD may produce several overlapping bounding boxes for each vanishing point detected in an image. Since only one bounding box per vanishing point is desired, a technique called non-maximum suppression, is applied to discard the extra bounding boxes. Similar to the SSD model described in Section 4.1.1, the SSD for vanishing point detection is implemented with MobileNets as the base network to reduce computational complexity (see Figure 7.5).

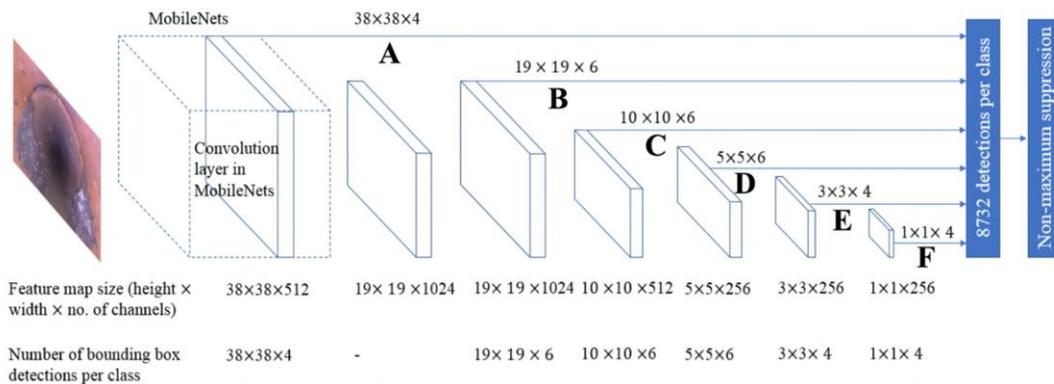


Figure 7.5 Architecture of the SSD MobileNets model used for vanishing point detection

Model Training and Inference

The SSD MobileNets model was trained using 1000 annotated sewer images. The images used for training were extracted from CCTV inspections of over 100 vitrified clay sewer pipe segments. The pipe diameters were 8 inches and 10 inches and originate from Ohio and Florida. The images were manually annotated, such that bounding boxes were drawn over the vanishing points. In order to reliably be able to detect vanishing points in CCTV videos, it was important to train the model with representative inspection images, i.e., images that exhibited large variations in camera orientations, illuminating conditions, and motion blur were included in the training and testing data sets. Among the 1000 images, 80 percent of the images were used in the training set, 10 percent in validation set, and 10 percent in the testing set. Images in the training and validation sets were augmented by applying combinations of image rotations, contrast changes, and random cropping. The augmentations serve to artificially enhance the training set with greater variations in the images. The model was trained for approximately two hours using an Nvidia GTX1070Ti 8GB graphics card to accelerate the training process. The mAP on the validation and testing set were found to be 0.90 and 0.87, respectively, using an IOU threshold of 0.5. The mAP on the testing set was approximately 0.03 lower than the mAP on the validation set, indicating low overfitting of the model.

The trained model was then applied to CCTV videos with native resolutions equal to 640×480 pixels. The inference was performed using a desktop computer with an Nvidia GTX1070Ti 8GB graphics card and yielded a video processing frame rate of 44 frames per second. The frame rate could further be improved by incorporating techniques such as neural network pruning. However, since our model achieved faster than real time performance (i.e., a frame rate greater than 30 frames per second), we did not implement pruning at the current stage.

Camera Orientation Estimation from Vanishing Point Position

The trained SSD model takes CCTV images as input and outputs the position of the vanishing point, represented by four bounding box coordinates (i.e., the x and y coordinates of the top left and bottom right vertices) (see Figure 7.6). Note: the images have a size of 640 pixels (horizontal) by 480 pixels (vertical). The coordinates of the center of the vanishing point are then calculated by averaging the coordinates of the two vertices. It was observed that the position of the center of

the vanishing point is highly dependent on the orientation of the camera. For instance, when a camera is oriented left off the centerline, the vanishing point center appears towards the right of the image, and when the camera is oriented right off the centerline, the vanishing point center appears towards the left of the image (see Figure 7.7). Hence, the methodology presented in this study, uses the position of the vanishing point center to determine the orientation of a camera.

Based on an analysis of 1000 image frames that we manually annotated, the following observations were made: LF orientations resulted in the vanishing point center's x coordinate lying between 0 and 209 pixels, FV orientations resulted in the vanishing point center's x coordinate lying between 210 and 430 pixels, and RF orientations resulted in the vanishing point center's x coordinate lying between 431 and 640 pixels. Vanishing points were typically not visible in LV or RV frames.

Thus in order to estimate the camera orientation in CCTV videos, our method first detects vanishing points using the SSD model, and then uses the following rule to classify the camera orientation: If the vanishing point's x coordinate lies between: (1) 0 and 209 pixels, the camera orientation is classified as LF; (2) 210 and 430 pixels, the camera orientation is classified as FV; and (3) 431 and 640 pixels, the camera orientation is classified as RV. However, vanishing points are not always visible or reliably detected in CCTV images and the Optical Flow Calculation Module is used to estimate the camera orientation in the absence of vanishing points.

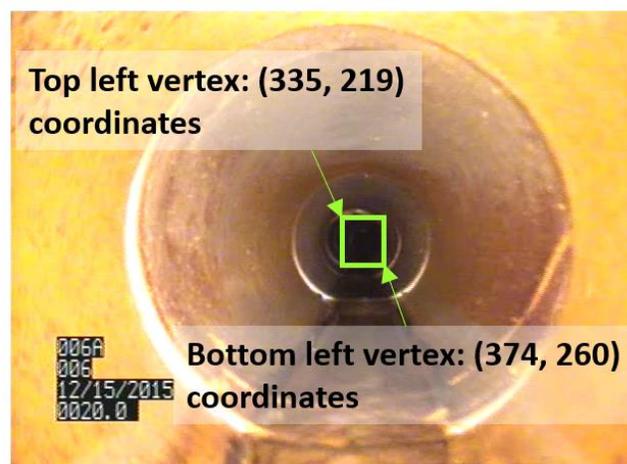
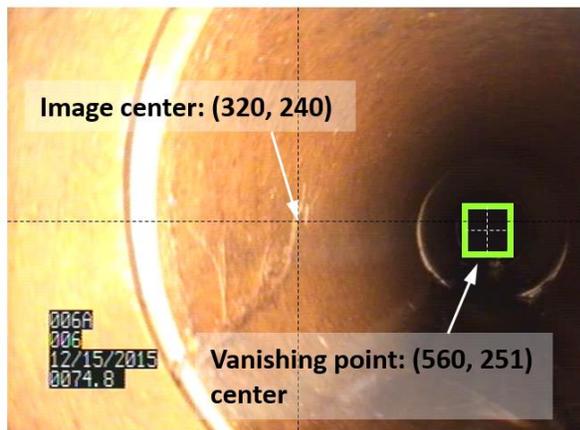


Figure 7.6 Example bounding box coordinates output by the SSD

a) Camera oriented left off centerline



b) Camera oriented right off centerline

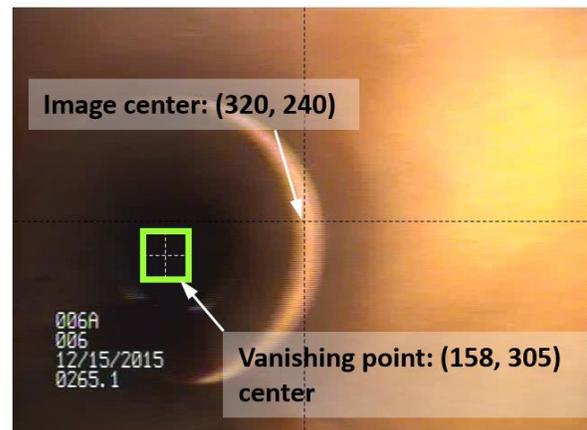


Figure 7.7 Example to illustrate how the vanishing point center varies with camera orientation

7.2.3 Optical Flow Calculation Module

Optical flow is a computer vision technique, which when applied to a video, can be used to estimate the direction of motion of the camera, in order to determine whether a camera travels forward, turns left, turns right, or rotates on one of its axes. Hence, optical flow can be used to estimate the orientation of a camera in the absence of vanishing point detections. The approach presented in this study uses optical flow to determine and quantify the motion of a camera in a video, based on which its orientation is determined.

Background on Optical Flow

Optical flow-based motion estimation calculates the motion of corresponding pixels in successive image frames. The algorithm first identifies keypoints (i.e., visually significant features such as Canny Edges) in the first image frame of a video. Next, a set of matching keypoints, i.e., features which are visually similar to the keypoints in the first image are identified in the second image frame. The algorithm then computes the displacement vectors of the matching keypoints from one image to the next (see Figure 7.8). These displacement vectors, which are also referred to as optical flow vectors, indicate the velocity of motion. Since objects in sewer CCTV videos are predominantly stationary, the optical flow vectors would correspond to the motion of the camera. Thus, the optical flow vectors can be used to estimate whether the camera is stationary, moving forward, or turning towards a wall. Note: the proposed method would not be applicable in videos

with floating objects, vermin, and infiltration gushers, since the assumption of sewer objects being stationary would be violated.

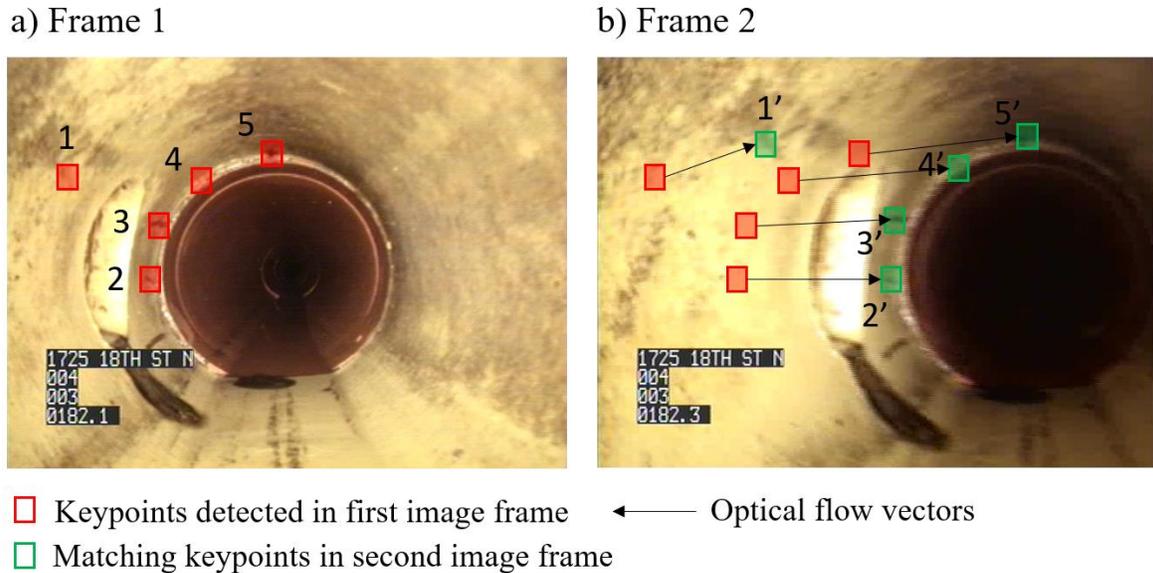


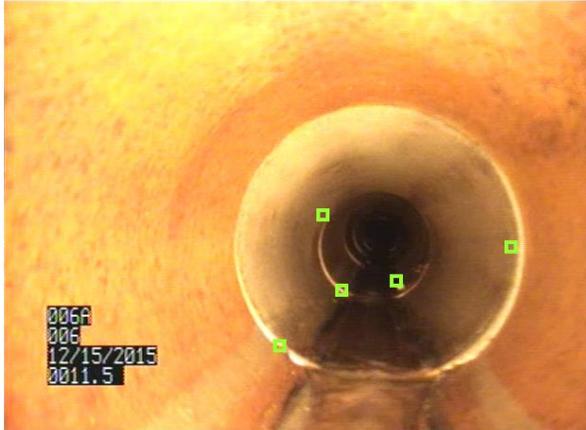
Figure 7.8 Illustration of optical flow vectors computed between two image frames

Several algorithms exist for calculating optical flow, such as the Horn-Schunck and Block Matching Algorithm). Halfawy and Hengmeechai (2014c) demonstrated that the Lucas-Kanade algorithm (Lucas and Kanade 1981) yields the highest accuracy when applied to sewer CCTV videos, since it is not sensitive to image noise. The Lucas-Kanade algorithm uses the following three assumptions to determine flow vectors between two images: (1) brightness of pixels does not change significantly between consecutive image frames, (2) the camera motion is relatively small, and (3) adjacent pixels have nearly similar flow vectors. In general, sewer CCTV videos satisfy these constraints; however, a small fraction of the images violate the brightness constancy assumption, due to reflection of camera light on wet surfaces. Despite this violation, Halfawy and Hengmeechai (2014c) found the Lucas-Kanade technique to outperform other methods. The Lucas-Kanade algorithm was also used successfully by Chen et al. (2018) to determine whether CCTV cameras are in motion or stationary; however, their study does not report the accuracy of using the algorithm. Aligning with the rationale of these previous studies, the approach presented in the current study also uses the Lucas-Kanade algorithm for optical flow calculation.

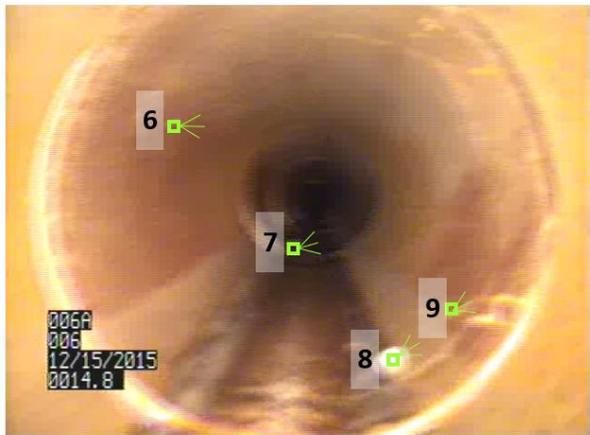
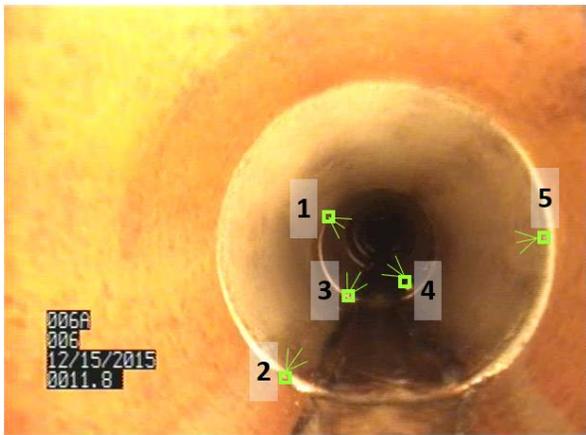
Sewer CCTV Camera Motion Estimation using Optical Flow

The optical flow vectors calculated for a particular image frame represent the displacement in pixels of the keypoints moving from the previous frame to the current frame. The direction of camera motion can be determined from these vectors. The approach in this study uses the optical flow vectors to estimate whether the robot is turning left, turning right, or not turning at all. Based on preliminary tests the following observations were made. When the robot travels forward, the optical flow vectors tend to point radially outward, whereas backward motion results in radially inward vectors (see Figure 7.9a). Right turns result in flow vectors that tend to point left, whereas left turns result in right pointing vectors (see Figure 7.9b). The proposed approach for estimating the camera motion involves identifying optical flow vectors in successive image frames and computing the length and direction of each these vectors. The length of an optical flow vector indicates the distance travelled by a keypoint in successive frames and can be used to determine the magnitude of camera motion. Longer vectors are indicative of greater motion between successive image frames than shorter vectors. For instance, vector number 6 in Figure 7.9b has a magnitude of 28 pixels indicating greater motion than vector number 1 in Figure 7.9a, which has a magnitude of 17 pixels.

a) Flow vectors point radially outward indicating forward motion



b) Flow vectors point left indicating a right turn



Optical Flow Vector Magnitudes:

- 1 – 17 pixels
- 2 – 26 pixels
- 3 – 16 pixels

- 4 – 15 pixels
- 5 – 22 pixels
- 6 – 28 pixels

- 7 – 18 pixels
- 8 – 18 pixels
- 9 – 20 pixels

Figure 7.9 Optical flow vectors corresponding to forward motion and right turn of a CCTV camera

Determining the orientation of a camera using optical flow requires the flow vectors to first be calibrated. Our approach for calibrating the flow vectors is as follows. First, 25 video snippets of CCTV videos where the camera turns towards the left wall and 25 video snippets of CCTV videos where the camera turns towards the right wall were extracted from a dataset of CCTV videos. Each snippet begins with the camera facing forward, i.e., FV and ends with the camera pointing at a wall, i.e., LV or RV. The average duration of the snippets was 1.5 seconds.

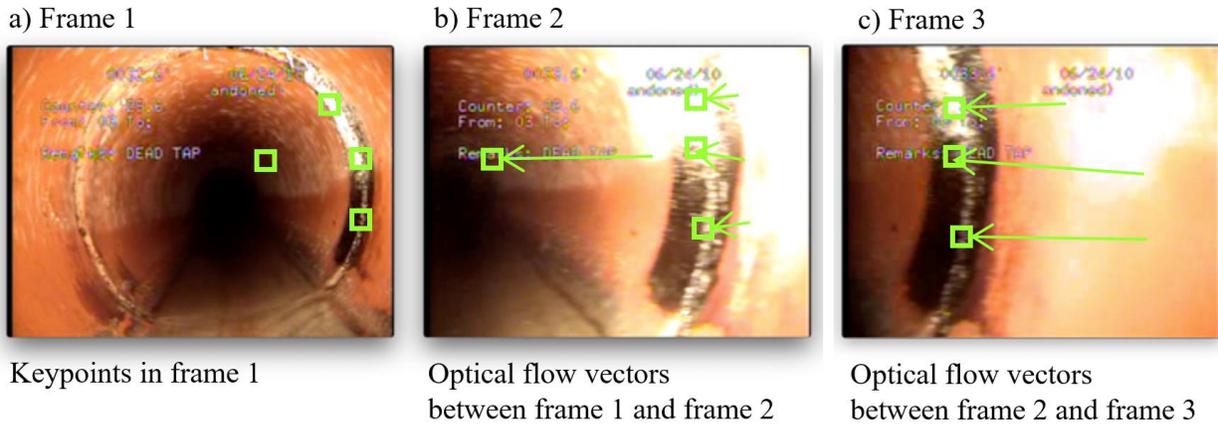


Figure 7.10 Example optical flow vectors computed on image frames that depict a right turn

For each video snippet (among the 50 snippets), image frames are extracted at a frame rate of 30 frames per second. Hence, for a video with a duration of 1.5 seconds, 45 image frames are extracted. Optical flow vectors are then calculated between successive image frames. For 45 image frames, 44 sets of optical flow vectors would be calculated. Figure 7.10 illustrates image frames extracted from a video snippet of a right turn. Note for illustration simplicity, Figure 7.10 only shows three image frames, whereas the actual methodology would extract image frames at 30 frames per second and then calculate optical flow vectors between successive frames. The horizontal components of the optical flow vectors are then computed. For instance, the flow vectors computed for the video snippet in Figure 7.10 have an average magnitude of 267.5 pixels in the left direction. Upon computing the flow vectors for all 50 video snippets it was found that for left turns, the average magnitude of flow vectors was 252.1 pixels with a standard deviation of 56.5 pixels. For right turns, the average magnitude of flow vectors was 237.8 pixels with a standard deviation of 50.3 pixels.

In order to determine the orientation of a camera, our approach computes the magnitudes of optical flow vectors in the horizontal direction and compares this value with the average magnitude for left and right turns. If the computed magnitude is $252.1 \text{ pixels} \pm 56.5 \text{ pixels}$ in the left direction, then the camera is assumed to have made a turn towards the left wall and the resulting view is LV. If the computed magnitude is $128 \text{ pixels} \pm 28.3 \text{ pixels}$ (i.e., half the magnitude of a left turn) in the left direction, our methodology assumes that the resulting view is LF. If the computed

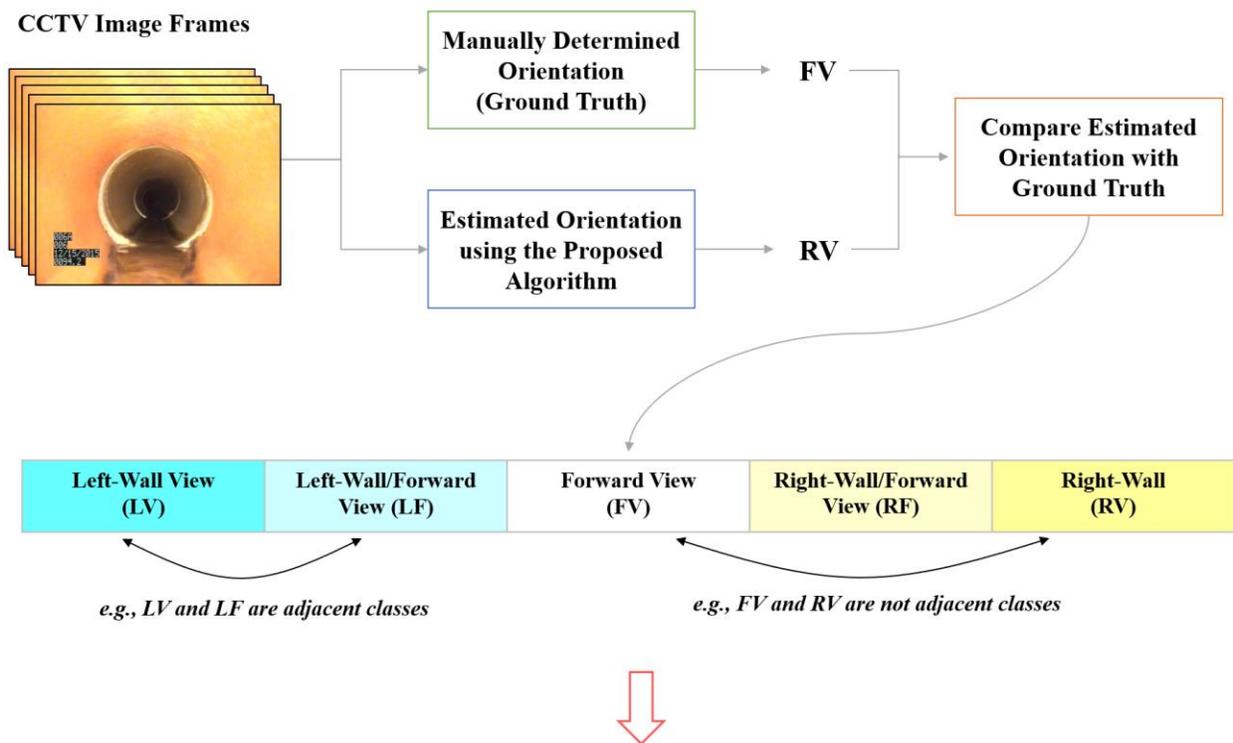
magnitude of flow vectors is $237.8 \text{ pixels} \pm 50.3 \text{ pixels}$, then the camera is assumed to have made a turn towards the right wall and the resulting view is RV. If the computed magnitude is $118.9 \text{ pixels} \pm 25.1 \text{ pixels}$ (i.e., half the magnitude of a right turn) in the right direction, our methodology assumes that the resulting view is LR.

7.3 Experiments and Discussions

A prevalent issue in the area of automated sewer CCTV inspections is the lack of benchmark datasets to evaluate and compare the performance of different methodologies (Haurum et al. 2020). Hence, the datasets used for experimental evaluation are constrained by the availability of data and the time needed for data cleaning, preprocessing, and annotation. The methodology presented in this study was evaluated on a set of 10 sewer CCTV videos of 8-inch diameter vitrified clay sewer pipes located in Florida and Ohio. Note: these videos are different from the videos used for training the vanishing point detection. The average video duration was 15 minutes and the average length of pipe was 280 feet. The objective of the experimental setup was to evaluate the accuracy of camera orientation estimation, when applied to the CCTV videos. Hence, the frames in the videos were first manually annotated to denote the angle made by the robot. Since measuring and annotating the angle of each and every frame in the videos would be too time consuming, a set of 50 candidate frames were selected for annotation from each video (resulting in a total of 500 annotated frames). These 500 frames were then manually categorized as belonging to five different classes (i.e., LV, LF, FV, RF, RV), based on the orientation of the camera according to the convention depicted in Figure 7.4. The total amount of manual effort needed to extract and categorize the 500 image frames was approximately 15 manhours.

The proposed method for estimating camera orientation was then applied to each of the 10 CCTV videos. In order to calculate the accuracy of the algorithm, the orientation estimates generated by the proposed algorithm were compared with the ground truth values (i.e., manually determined orientations). If the orientation estimate produced by the algorithm matched exactly with the ground truth annotation, then it was considered a perfect match. For example, if the ground truth orientation of a frame was FV and the orientation estimate produced by the algorithm was also FV, then the estimate would be considered a perfect match. If the estimated orientation differed from the ground truth, but belonged to an adjacent class, it would be considered as a soft match. For

example, if the ground truth annotation of an image was FV, but the estimated orientation was LF or RF, the output would be considered a soft match, since LF and RF are the classes adjacent to FV. However, if the estimated orientation differed from the ground truth and did not belong to an adjacent class, it would be considered as an incorrect classification. For instance, if the ground truth orientation of a frame was FV but the output of the algorithm was RV or LV, then the estimate would be considered incorrect, since RV and LV are not adjacent to FV. Figure 7.11 provides a conceptual overview of the experimental setup.



- Perfect Match:** If Estimated Orientation **is equal to** Ground Truth Orientation
- Soft Match:** If Estimated Orientation **is adjacent to** Ground Truth Orientation
- Incorrect:** If Estimated Orientation **is not equal to or adjacent to** Ground Truth Orientation

Figure 7.11 Conceptual overview of experimental setup for accuracy evaluation

The soft match category was created to account for edge cases, i.e., images which could be interpreted as belonging to multiple classes. Based on the number of perfect matches and soft matches, two accuracy metrics were calculated using the following two equations:

$$\text{Perfect Match Accuracy} = \frac{\text{No. of perfect matches}}{\text{Total no. of images}} \quad (7.1)$$

$$\text{Soft Match Accuracy} = \frac{\text{No. of perfect matches} + \text{No. of soft matches}}{\text{Total no. of images}} \quad (7.2)$$

The results of orientation estimation applied to the 500 image frames are listed as a confusion matrix in Table 7.1. The proposed method yielded 399 (i.e., 83+74+84+76+82) perfect matches and 41 (i.e., 6+10+3+4+5+5+6+2) soft matches. Hence, the perfect match accuracy of the proposed method on the candidate image frames is 0.798 (i.e., 79.8 percent) and the soft match accuracy is 0.880 (i.e., 88.0 percent).

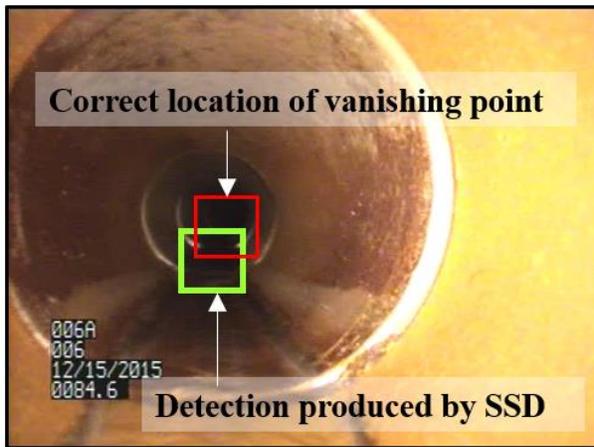
Table 7.1 Confusion matrix for camera orientation estimation

		Output of Camera Orientation Estimation Algorithm				
		LV	FV/LV	FV	FV/RV	RV
Ground-truth Frame Classifications	LV (Total = 100)	83	6	4	4	3
	LF (Total = 100)	10	74	3	7	6
	FV (Total = 100)	3	4	84	5	4
	RF (Total = 100)	6	7	5	76	6
	RV (Total = 100)	3	4	6	5	82

The proposed method yielded 101 (out of 500) incorrect orientation estimates, i.e., 41 soft matches and 60 incorrect classifications. An analysis of these images revealed two sources of errors. The first source of error, which accounted for 8 soft matches and 38 incorrect classifications originates from incorrect vanishing point detections. This occurred when the SSD object detection model identified vanishing points in images, but the bounding box coordinates of the vanishing points were inaccurate (see Figure 7.12a). This error could be reduced by using more computationally expensive models such as YOLO and RetinaNet instead of the SSD. YOLO and RetinaNet have been shown to achieve higher mAP values on the PASCAL VOC and MSCOCO benchmark datasets compared to SSD (Redmon et al. 2016, Lin et al. 2018), and could hence generate more

precise detections of vanishing points. Additionally, the SSD model also tended to erroneously identify a small fraction of lateral connections as sewer mains, resulting in incorrect estimates of the camera orientation (see Figure 7.12b). These incorrect detections stem from the visual similarities between some lateral connections and of sewer mains. Retraining the SSD model to distinguish between lateral connections and sewer mains is a possible solution to this problem and will be explored in future research.

a) Image of Sewer Main



b) Image of Lateral Connection

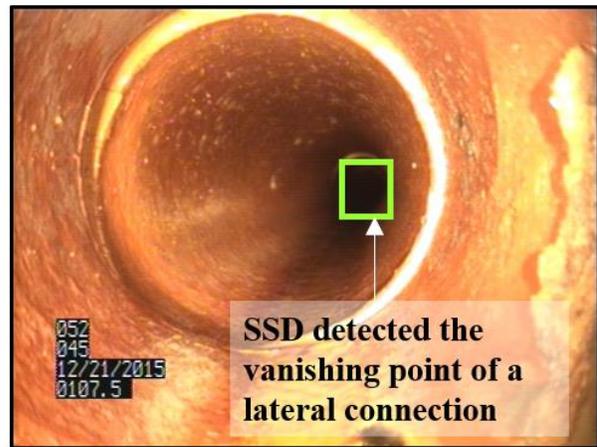


Figure 7.12 Images to illustrate errors originating from incorrect vanishing point detections

The second source of error, which accounted for 7 soft matches and 16 incorrect classifications originates from errors in optical flow calculation. These errors are caused by the optical flow vectors indicating an incorrect direction of motion. An example of such an error is a sequence of image frames where the camera turns right, however, the optical flow vectors indicate that the camera turned left. These errors are caused because distinct keypoints could not be identified in image frames and most notably due to brightness and contrast changes. For example, some images appeared overly bright due to the reflection of camera light on the pipe walls, resulting in a violation of the Lucas-Kanade method's brightness constancy assumption, leading to erroneous optical flow calculation. A potential solution to this problem could be the incorporation of deep learning-based optical flow techniques, such as FlowNet (Dosovitskiy et al. 2015) into the camera orientation estimation framework. Techniques such as FlowNet use CNNs to estimate optical flow

between images, rather than keypoint matching (used by the Lucas-Kanade method), resulting in more accurate motion estimation. However, Dosovitskiy et al. (2015) mention that the preparation of training images for CNN-based optical flow requires considerable manual effort, which is why FlowNet has only been validated using synthetic images (i.e., virtual images generated using computer graphics techniques). Hence, further research is required to adapt CNN-based optical flow techniques for sewer camera orientation estimation.

A major limitation of our proposed approach is that it cannot account for rotations along the forward axis of the camera (see Figure 7.13a). Such rotations typically result in image frames that appear tilted. However, this type of rotation can be identified by analyzing the position of water lines (i.e., flow lines on the bottom of pipes) in images (see Figure 7.13b). Hence, a separate module to detect the position of water lines in sewer images could be incorporated into the proposed approach to account for such rotations.

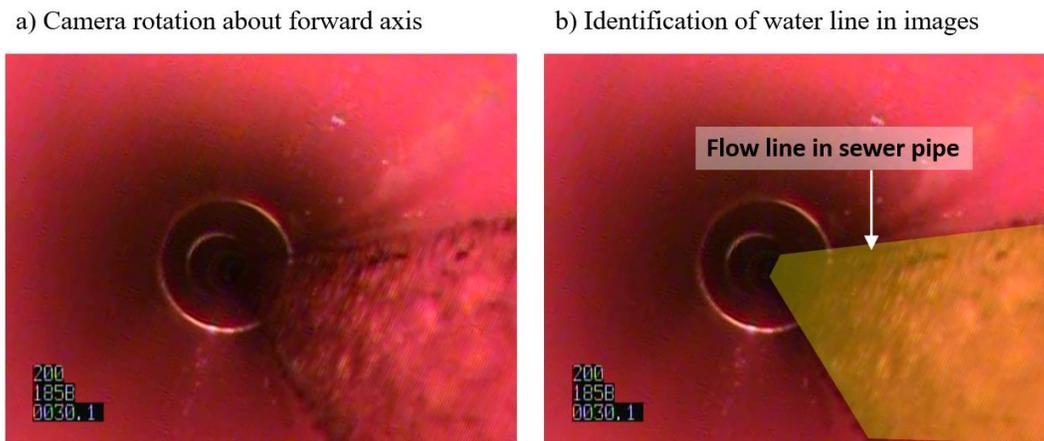


Figure 7.13 Example image to demonstrate camera rotations about forward axis

7.4 Conclusions

This chapter presents the development of a novel vision-method for estimating the orientation of cameras in sewer CCTV inspection videos. The proposed method could become an important component in the development of automated defect interpretation systems by facilitating circumferential localization of defects. Additionally, the method could find application in autonomous sewer robotics, by enabling robots to determine their position and orientation inside

pipes, in the absence of GPS signals. The proposed method leverages visual cues in CCTV images, such as the positions of vanishing points and optical flow vectors, to classify the camera orientation into five classes (i.e., LV, LF, FV, RF, RV). The position of the vanishing point of a pipe, i.e., the point at which the walls appear to converge, is used to characterize the orientation of a sewer camera. In order to detect vanishing points, a deep learning-based object detection model called SSD-MobileNets was used due to its high processing speed. The SSD-MobileNets model was developed using a dataset of 1000 images, using an 80/10/10 training/validation/testing split, and achieved a mAP of 0.87 on the testing set. Since vanishing points are not visible in all image frames (e.g., when the camera is oriented towards a wall) of a CCTV video, a technique called optical flow is used to infer the orientation of a camera when vanishing points are not detected. The Lucas-Kanade algorithm was adopted for optical flow calculation, and the flow vectors were calibrated using 50 video snippets from CCTV videos. Evaluation of the proposed camera orientation estimation method on 500 images (from 10 CCTV videos of 8-inch diameter VCP sewers in Florida and Ohio) yielded a classification accuracy of 79.8%. Two sources of error that accounted for a majority of the incorrect detections were 1) imprecise detection of vanishing points and 2) erroneous optical flow vector calculation due to brightness changes in images. At the current stage, the proposed method is limited to classifying the camera orientation into five classes; however, further research is required to estimate the camera orientation in degrees. Additionally, the proposed method cannot address camera rotations along the forward axis, which lead to tilted-view frames.

Automated CCTV inspections of sewers have the potential to improve the consistency, accuracy, and speed of condition assessment. Prior research on automated CCTV inspections has focused on two areas, automated interpretation of defects and autonomous robot navigation. However, both areas are challenged by the lack of techniques to estimate the orientation of CCTV cameras in sewer pipelines. In the area of automated defect interpretation, this challenge manifests as the inability to determine the circumferential location of defects, which is a crucial component of pipe structural integrity assessment. In the area of autonomous robot navigation, this challenge leads to an inability in determining a robot's orientation in a pipe, which is essential for path planning. Hence, the proposed camera orientation estimation algorithm developed in this study represents advancements in automated defect interpretation and autonomous robot navigation, paving the

way for future inspection systems that combine both technologies. The development of UAV-based sewer pipeline inspection systems is an emerging area of research (Rizzo et al. 2016) and the method proposed in this study could be integrated with such UAV systems, to facilitate rapid inspection and assessment of sewers at a fraction of the cost of current methods. Additionally, using a network of wireless sensors, such inspection technologies could relay information about the condition of pipes in real-time, which would enable asset managers to make quick, pre-emptive maintenance decisions. Furthermore, since the proposed method is vision-based, it may also find application in other GPS-denied environments, such as utility tunnels, boreholes, and underground mines.

CHAPTER 8. A FRAMEWORK FOR MINING SPATIAL CHARACTERISTICS OF SEWER DEFECTS FROM INSPECTION DATABASES

[The content in this chapter is reprinted from a manuscript that is currently under preparation]⁵

Currently, municipalities rely on visual inspections and deterioration modeling to plan maintenance, repair, and rehabilitation activities. Visual inspections seek to assess the condition of pipes by identifying various types of defects (e.g., cracks, fractures, broken walls, etc.) in pipes. Visual inspections also entail assigning numerical grades (typically between 1 and 5) to defects, in order to represent their severity and provide a snapshot about the condition of a pipe at an instance of time. Since budget constraints limit the percentage of pipes that can be inspected each year, asset managers rely on deterioration modeling to estimate the future condition of pipes, based on which maintenance, repair, and rehabilitation decisions are made (Harvey and McBean 2014). Current deterioration modeling approaches represent the condition of pipes using single numeric grades and do not account for information about individual defects in pipes. For instance, current approaches represent the average condition of a pipe as the sum of grades of individual defects. However, the aggregation of defect grades into a single condition grade leads to a loss of spatial information, i.e., information about the density, severity, and co-occurrence characteristics of defects—information which can play a crucial role in calculating a pipe’s likelihood of failure. For instance, the approach of using a single aggregated condition grade overlooks the increased likelihood of failure of a pipe with defect clusters (i.e., areas with multiple defects in proximity). Figure 8.1 illustrates this problem. Under the conventional method of assigning a single grade to pipes, the pipes in Figure 8.1a and Figure 8.1b would both be assigned identical condition grades and hence be deemed to be equally prone to failure.

⁵ Tentative title: Kumar, S. S., Abraham, D. M., Choi, J. (2020). A Framework for Mining Spatial Characteristics of Sewer Defects from Inspection Databases. Tables and figure captions have been modified to maintain the form of the dissertation.

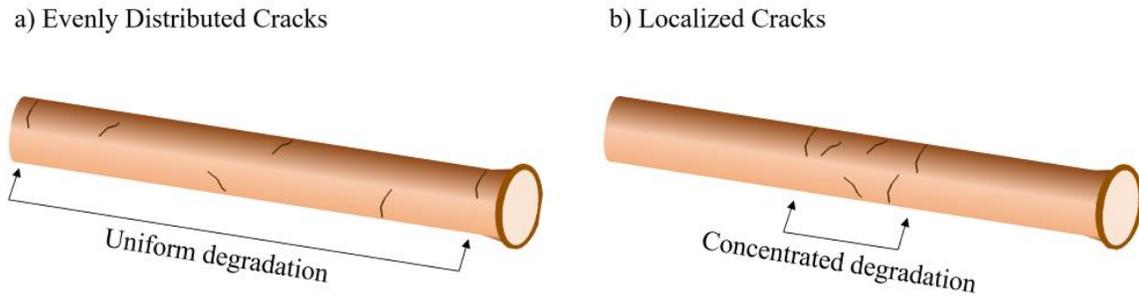


Figure 8.1 Illustration to signify the importance of considering defect locations along a pipe

However, it could be argued that that the pipe in Figure 8.1b has a higher likelihood of failure for the following reasons: (1) defects, which are close to each other could propagate and coalesce into more severe defects; (2) multiple cracks and fractures may lead to soil infiltration leading to the formation of voids over the pipe. Voids over pipes are known to result in sinkholes; and (3) multiple defects in proximity can lead to a localized region of weakness, resulting in an increased likelihood of collapse. Hence, existing deterioration modeling approaches, which rely on aggregated condition scores, do not account for the relationships between defect locations and likelihood of failure. The lack of consideration for spatial information also affects maintenance, repair, and rehabilitation decisions. For instance, if defects are evenly distributed along the length of a pipe, such as the case depicted in Figure 8.1a, it is likely that the entire length of pipe must be repaired or rehabilitated. However, for instances where only a section of the pipe is deteriorated with the rest of the pipe being in good condition such as the case depicted in Figure 8.1b, patch repairs may be an economical alternative to repairing the entire length of pipe. Thus, there is a need to develop techniques that also consider spatial information about defects when assessing sewer pipe condition.

8.1 Related Studies on Sewer Deterioration Modeling

Development of deterioration models to forecast the condition of sewer pipes is an active area of research and consists of two predominant approaches: physical modeling and statistical modeling (Ana and Bauwens 2010; Wilson, Filion, and Moore 2017; Nicolas Caradot et al. 2017). Physical modelling aims to simulate the physical processes of deterioration and the failure mechanisms of buried pipes through an analysis of the loads (physical and environmental) and estimated capacity

of the pipes to resist the loads (Ana and Bauwens 2010). Prior studies have mainly used physical models to estimate the condition of sewer pipes under the influence of corrosion. For example, König (2005) used soil moisture, soil aggressiveness, and resistance of the cement as key factors to estimate the magnitude of external corrosion on sewer concrete pipes in millimeters. Their study used empirical observations to calibrate the effects of soil moisture (i.e., dry, medium, or wet), soil aggressiveness (non, medium, or high corrosive), and cement resistance (good, medium, or poor) on the magnitude of external corrosion. However, the authors concluded that the physical mechanisms that lead to pipe failure are multifaceted and complex, and their approach could only provide a rough estimation of the external corrosion of pipes (without stating formally the accuracy of their prediction). Vollertsen and König (2005) and Wells and Melchers (2014) attempted to predict internal corrosion of pipes using biological and chemical indicators, such as organic matter and presence of gas at the water/gas surface. Both of these studies entailed the collection of a wide range of data from pipes such as the pH of concrete, concrete thickness, hydrogen sulfide concentration, temperature, humidity, etc., to develop physical models to represent pipe deterioration. Because of the intensive data collection requirements, their approach is only suitable for assessing the condition of individual pipes and not economically feasible for analyzing a network of pipelines. Additionally, because of the complexity of modeling the physical deterioration in pipes, physical modeling has low adoption rates among municipalities (Ana and Bauwens 2010; Rokstad and Ugarelli 2015).

To address the limitations of physical models, researchers have proposed statistical models, which attempt to predict the condition of sewer pipes through an analysis of historical sewer condition data. This approach does not entail the collection of new data/samples from pipes and relies entirely on historical data that have already been collected during routine inspections. A wide variety of statistical models have been proposed in literature, based on cohort survival functions (Baur and Herz 2002), Markov chains (Wirahadikusumah et al. 2001; Micevski et al. 2002; Tran et al. 2006; Le Gat 2008; Ana 2009; Rokstad and Ugarelli 2015) and logistic or multiple regression (Ariaratnam et al. 2001; Chughtai and Zayed 2008; Ahmadi et al. 2013; FuchsHanusch et al. 2015). More recently, Mohammadi et al. (2019) proposed a logistic regression model to predict the condition of sanitary sewer pipes of the City of Tampa, Florida. The model used physical factors (i.e., pipe age, material, size, depth, slope, and length) and environmental factors (i.e., soil type

and water table level) to predict pipe condition and calibrated their model based on condition data from 20,282 pipe segments. The authors used 80% of the data for training and 20% for validation and state that their model achieves an 81% accuracy in predicting the condition of pipes in the validation set. Recent approaches have also used machine learning techniques such as neural networks (Tran et al. 2006; Sousa et al. 2014; Sousa et al. 2019), random forests (Vitorino et al. 2014; Harvey and McBean 2014; Laakso et al. 2018), decision trees (Harvey and McBean 2015), and support vector machines (Sousa et al. 2014; Hernández et al. 2018), to predict the condition of sewer pipes. For instance, Harvey and McBean (2015) showed that decision tree models could be used to predict the condition of a pipe (either as ‘good’ or ‘bad’) at an accuracy of 76% using pipe-specific attributes such as pipe age, diameter, and length for pipes based in Guelph, Ontario, Canada. Laakso et al. (2018) applied the random forest algorithm to model the physical condition of sewer pipes in southern Finland with a data set for 6,700 inspected sewer pipes. Their model could predict the condition of sewers with an accuracy of 62% using seven explanatory variables (i.e., pipe location, pipe slope, pipe age, pipe length, pipe installation year, sewage flow, and construction class).

Despite advancements in sewer condition prediction, the approaches proposed in all previously mentioned studies represent the condition of pipes using single numeric grades. That is, the condition grades of individual defects are aggregated into a single numeric score to represent the condition of a pipe. However, this approach leads to a loss of spatial information (i.e., locations, densities, and co-occurrence characteristics) about defects. Our study aims to address this limitation by developing a methodology for assessing sewer deterioration by incorporating spatial information, such as the locations, densities, and co-occurrence characteristics of defects in pipes. A methodology called Defect Cluster Analysis (DCA) is proposed to identify and quantify defect clusters (i.e., areas with multiple defects in close proximity) in pipes. DCA could be used as a diagnostic tool to identify sections of the pipe that are likely to fail or have the propensity to progress into severe defects. When combined with contextual information, DCA could provide additional insights into pipe failure, such as the likelihood of sinkhole formation due to sand infiltration. The DCA approach could be used to periodically determine how the distribution of defects and their clustering progresses with time and could reveal pipeline deterioration patterns. The identification of defect clusters could also inform the choice of rehabilitation option. For

instance, the identification of defect clusters could lead to insights into whether whole length rehabilitation or local patch repairs should be pursued.

8.2 Methodology

This section describes two techniques (i.e., DCA and co-occurrence mining), which have been developed to facilitate the assessment of pipe deterioration based on the relative locations of defects in pipe segments. The first technique is called DCA and identifies pipeline segments that contain multiple defects which are closely located to one another. The second technique is called defect co-occurrence mining and identifies pairs of defects which occur frequently together in pipes.

8.2.1 Defect Cluster Analysis (DCA)

In this study, a defect cluster is defined as a set of defects that are spatially collocated, i.e., consecutive defects which are within a predefined distance from one another. For a pipe segment with n defects $\{D_1, D_2, \dots, D_n\}$, D_i and D_{i+1} belong to defect cluster C_j , if $dist(D_i, D_{i+1}) \leq S$. Where, $dist(D_i, D_{i+1})$ is the longitudinal distance between the defects D_i and D_{i+1} , and S is a threshold that specifies the maximum distance between consecutive defects in a cluster. Figure 3 illustrates this concept using an example of a pipe segment with seven defects. In Figure 8.2, if D_1 to D_7 refer to defects located along a pipe segment, and the distance threshold S is taken to be two (2) meters, then $\{D_1, D_2, D_3\}$, $\{D_4, D_5\}$ and $\{D_6, D_7\}$ are considered as three (3) defect clusters in the pipe segment. If S is assumed to be one (1) meter, then $\{D_1, D_2, D_3\}$ and $\{D_6, D_7\}$ will be considered as defect clusters in the pipe segment. In practice, the selected value of S would depend on pipe material, soil type, pipe age, and the asset manager's preference. Hence, the algorithm is developed such that different values of S can be considered.

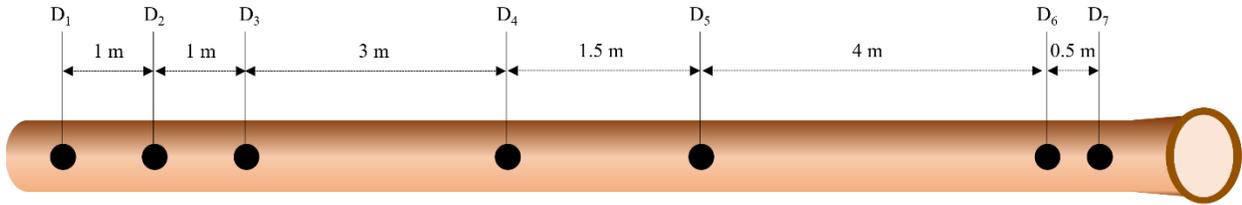


Figure 8.2 Illustration of a pipe segment with seven defects

To quantify the severity of degradation in a cluster, two metrics are defined. The first metric called Cluster Severity is defined as the sum of defect grades in a cluster, whereas cluster severity density is defined as the sum of defect grades in a cluster divided by the length of the cluster. Thus, cluster severity represents the total degradation in a cluster, however, the cluster severity grade represents the degradation per unit length of pipe. Cluster severity density controls for the unit length of cluster. A longer pipe may have more clusters (by virtue of it being longer), however, no statistical relationships between pipe length and cluster severity or cluster severity density were observed.

$$\text{Cluster Severity} = \text{Sum of Defect Grades in a Cluster} \quad (8.1)$$

$$\text{Cluster Severity Density} = \frac{\text{Sum of Defect Grades in a Cluster}}{\text{Length of Cluster}} \quad (8.2)$$

Cluster severity is an aggregate of the amount of deterioration in a cluster, whereas cluster severity density provides insights into how highly localized the defects are. Figure 8.3 illustrates three examples of defect clusters with different cluster severities and cluster severity densities. In Figure 8.3, Cluster 2 has higher cluster severity and cluster severity density values than cluster 1 and hence we consider it to be more deteriorated than Cluster 1. Cluster 2 and Cluster 3 have the same cluster severities, however, the cluster severity density in Cluster 2 is higher. This indicates that the defects in Cluster 2 are more localized than the defects in Cluster 3. The next section describes an algorithm that was developed to identify defect clusters in sewer pipe inspection databases.

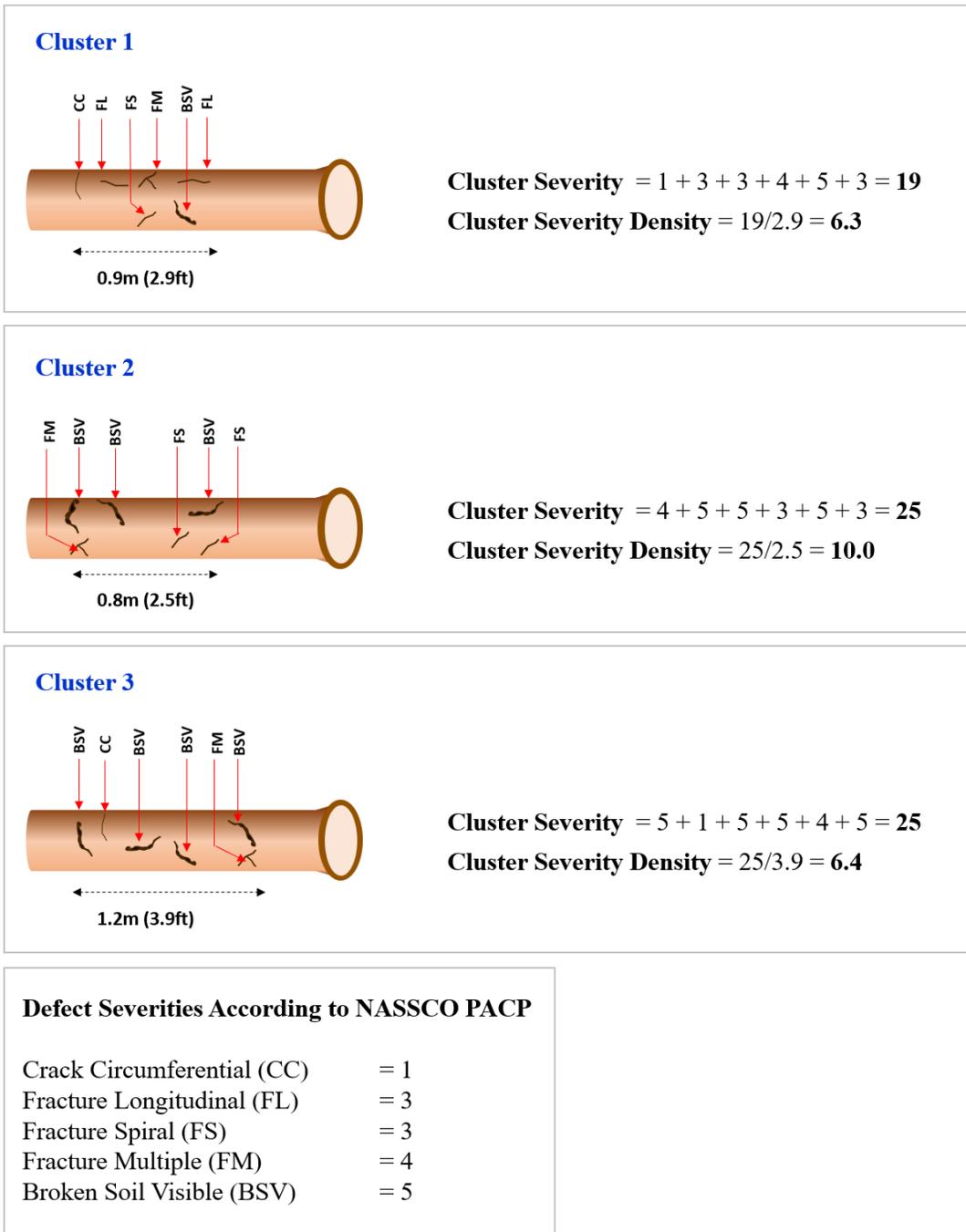


Figure 8.3 Illustration of three example defect clusters

Defect Cluster Identification Algorithm

The defect cluster identification algorithm takes a PACP inspection spreadsheet as input and outputs a list of clusters of various sizes, cluster severities, and cluster severity densities. Table 8.1 illustrates an example of such an inspection spreadsheet of a pipe segment that is approximately 100 meters long. For a pipe segment having less than or equal to n defects, the algorithm has a linear execution time (i.e., $O(n)$). That is, the number of execution steps taken by the algorithm is proportional to the number of observations in the spreadsheet. Figure 8.5 illustrates the execution time of the algorithm on inspection spreadsheets of varying sizes. The execution time was computed using a desktop computer equipped with an Intel Core i7-8700 CPU and using Python 3.6. From Figure 8.5 it can be observed that the execution time is approximately proportional to the number of defects in the spreadsheet. For instance, the time taken to process a spreadsheet containing 10 defects was 12 milliseconds, whereas the time taken to process a spreadsheet containing 81 defects was 77 milliseconds. Thus, when the number of defects increases eight times, the execution time of the algorithm increased approximately by a factor of eight. However, if the algorithm had a quadratic execution time (instead of a linear execution time), the execution time would increase approximately by a factor of 8^2 (i.e., 64), resulting in a slow processing time. Thus, because the algorithm developed in this study has a linear execution time, it is suitable for processing spreadsheets that contain hundreds or even thousands of defects.

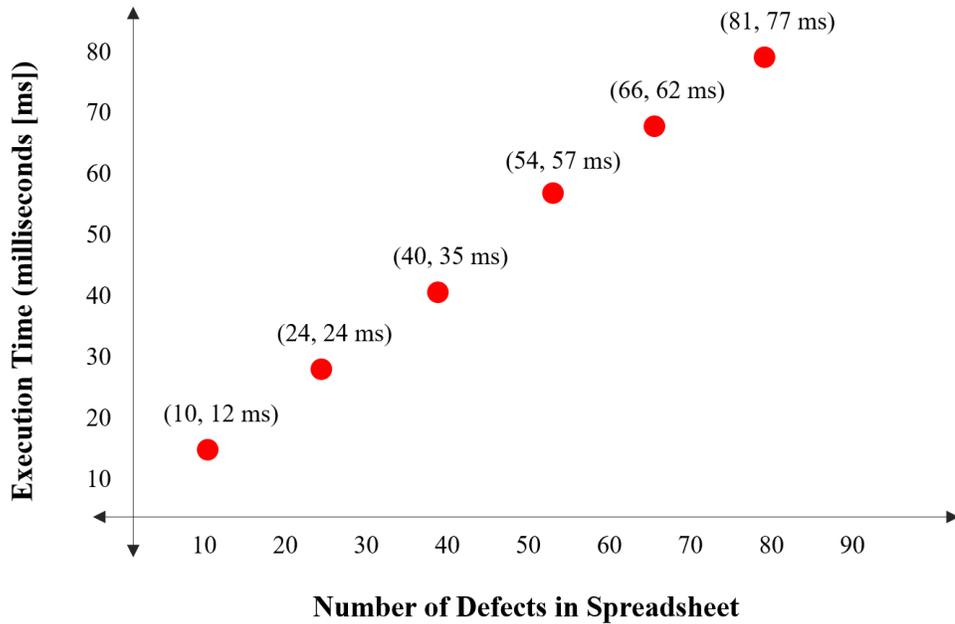


Figure 8.4 Execution time of the algorithm for inspection spreadsheets

Table 8.1 Example PACP spreadsheet table that the Defect Cluster Identification Algorithm takes an input

Defect Index	Defect Code	Distance (m)
0	FC (fracture circumferential)	25
1	CL (crack longitudinal)	26
2	B (broken)	28
3	CS (crack spiral)	52
4	JOM (joint offset medium)	88
5	FL (fracture longitudinal)	89

Let $d[i]$ be the distance of the i^{th} defect, and N be the total number of defects in a single pipe segment. Let $c[j]$ represent the j^{th} defect cluster. S represents the cluster threshold. The algorithm (written in pythonic pseudo code) for finding all defect clusters in the database is as follows:

$j = 0$ // j refers to the index of the cluster

for $i = 0$ to $N - 1$:

if $|d[i] - d[i + 1]| \leq S$: // if the distance between i^{th} and $(i+1)^{th}$ defect is less than the threshold

c[j].append(i) // append ith defect to cluster j

else:

j = j + 1

Table 8.2 shows the execution steps of the algorithm on the sample PACP spreadsheet presented in Table 2. For this example, S is assumed to be 2 meters. At the final step of execution, two defect clusters are identified. The first cluster (i.e., $c[0]$) consists of defects FC, CL, and B whereas the second cluster (i.e., $c[1]$) consists of defects JOM and FL. Note: additional constraints could be incorporated while searching for defect clusters, e.g., different distance thresholds for different classes of defects. The algorithm can be adapted to accommodate the preferences of a municipality/organization.

Table 8.2 Execution steps of the cluster finding algorithm

Step of Execution	Index of Defect Cluster j	Index of Defect i	Algorithm Logic and Execution	Clusters Identified c
1	0	0	Since $ d[i] - d[i + 1] \leq S$, append i to $c[j]$ $i = i + 1$	$c[0] = [0]$
2	0	1	Since $ d[i] - d[i + 1] \leq S$, append i to $c[j]$ $i = i + 1$	$c[0] = [0, 1]$
3	0	2	Since $ d[i] - d[i + 1] > S$, and $ d[i] - d[i - 1] \leq S$, append i to $c[j]$ $i = i + 1$ $j = j + 1$	$c[0] = [0, 1, 2]$
4	1	3	Since $ d[i] - d[i + 1] > S$, and $ d[i] - d[i - 1] > S$ $i = i + 1$	$c[0] = [0, 1, 2]$ $c[1] = []$
5	1	4	Since $ d[i] - d[i + 1] \leq S$, append i to $c[j]$ $i = i + 1$	$c[0] = [0, 1, 2]$ $c[1] = [4]$
6	1	5	Since $ d[i] - d[i + 1] > S$, and $ d[i] - d[i - 1] \leq S$, append i to $c[j]$ $i = i + 1$ $j = j + 1$	$c[0] = [0, 1, 2]$ $c[1] = [4, 5]$

8.2.2 Defect Co-Occurrence Mining

In this study, a technique called association rule mining is applied to databases of sewer inspection records to discover correlations among the occurrences of different types of defects. Association rule mining (also known as market basket analysis) is a method that is typically used for discovering customer purchasing patterns by extracting associations or co-occurrences from transactional databases. For example, discovering that online shoppers are likely to purchase two specific items together can assist in the design of websites and marketing strategies. Association rule mining was introduced by Agrawal et al. (1994) and can be stated as follows: Given two items X and Y , an association rule $X \rightarrow Y$ indicates that if X exists, then Y also exists. In this study, association rule mining is adapted to determine associations between defects in pipe segments, and

in clusters. $X \rightarrow Y_{pipe\ segment}$ indicates that if defect X exists in a pipe segment, then defect Y also exists in the same pipe segment.

Various metrics have been defined in literature to measure the strength of association rules. Three of the most commonly used metrics are support, confidence, and lift. Support measures how frequently an itemset (e.g., X, Y) appears in a dataset. Hence for a rule $X \rightarrow Y_{pipe\ segment}$, the support can be calculated as follows:

$$Support (X \rightarrow Y_{pipe\ segment}) = \frac{No. of pipe segments containing X and Y}{Total no. of pipe segments} \quad (8.3)$$

Confidence is an indication of how often the rule is found to be true, and can be calculated as follows:

$$Confidence (X \rightarrow Y_{pipe\ segment}) = \frac{No. of pipe segments containing X and Y}{No. of pipe segments containing X} \quad (8.4)$$

Hence, confidence can also be interpreted as an estimate of the conditional probability of finding item Y given item X .

Lift is the ratio of confidence to the expected confidence of a rule. A lift value greater than 1 indicates that an itemset appears more often together than expected, whereas a lift value less than 1 indicates that an itemset appears less frequently than expected. Lift can be calculated as follows:

$$Lift (X \rightarrow Y_{pipe\ segment}) = \frac{No. of pipe segments containing both X and Y}{\frac{No. of pipe segments containing Y}{Total no. of pipe segments} * \frac{No. of pipe segments containing X}{Total no. of pipe segments}} \quad (8.5)$$

8.3 Experiments and Discussions

The defect cluster identification algorithm described in the previous section was applied to a dataset containing 7193 sewer inspections (each inspection corresponds to a single sewer pipe segment between two manholes) of vitrified clay pipes. Note: pipe segment in this context refers to the region of pipe between consecutive manholes. The pipe segments are located in cities in South Carolina, Florida, and Ohio, and have a total length of approximately 457 kilometers. Figure 8.5 shows the pipe materials and sizes. The predominant pipe material is vitrified clay and most of the pipes are 30 cm (12 inches) in diameter. The length of pipe segments ranged from 88 feet to 550 feet, with the average length of pipe being approximately 64 m (210 ft). Since, all inspections in this dataset were manhole-to-manhole inspections, the length of inspected pipe equals the total length of pipe. The pipe segments contained a total of 15,527 instances of structural defects. Figure 8.6 shows the ten most frequent structural defects in the dataset.

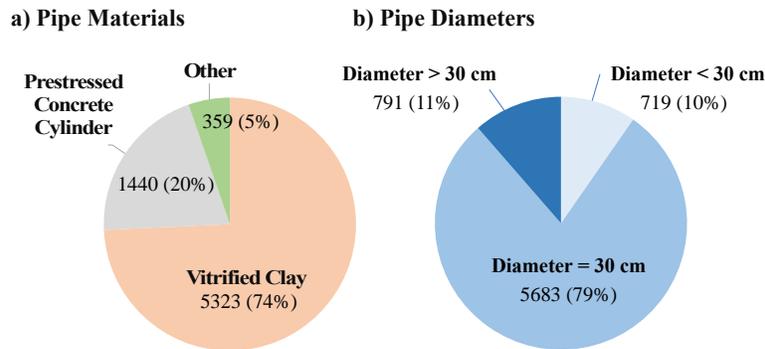


Figure 8.5 Number of pipe segments by: a) materials and b) diameters

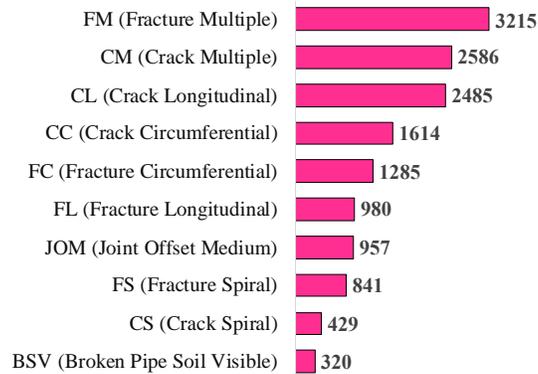


Figure 8.6 Number of instances of structural defects

8.3.1 Identification of Defect Clusters

In this study, DCA is applied to the sewer inspection dataset as follows. First, the defect cluster identification algorithm is executed with a threshold distance (i.e., S) of 0.1 m (0.3ft), which is approximately three times the average diameter of the pipes. Based on the selected value of S , the algorithm outputs a list of defect clusters, with associated cluster severities and cluster severity densities. The algorithm is then executed with the threshold distance (i.e., S) relaxed to 0.3 m (1.0 ft) and 0.9 m (3 ft). Table 8.4 lists the number of defect clusters of various sizes, along with their maximum, minimum, and average cluster severity densities. With $S = 0.9$ m (3 ft), the algorithm identified 627 clusters that contained two or more structural defects, with the longest cluster containing 16 defects (see Table 8.3). With $S = 0.3$ m and 0.1 m, the algorithm identified 197 and 28 defect clusters, respectively (see Tables 8.4 and 8.5). The top five highest severity density clusters are visually depicted in Figure 8.7.

Based on the values of the average, maximum, and minimum cluster severity densities, it was observed that lower values of S lead to the identification of clusters with higher cluster severity densities (see Tables 8.3, 8.4, and 8.5). However, the number of clusters identified by the algorithm are fewer for lower values of S . The identified clusters are then sorted according to their cluster severity densities in descending order. Duplicates, i.e., the same clusters which are identified across different values of S are deleted, leaving behind a sorted list of unique clusters sorted according to their cluster severity densities. In practice, we recommend using the algorithm in a staged approach, starting with a low value of S (e.g., 0.1 m). The low values of S will help identify the most severe clusters (i.e., clusters with highest severity density) and hence the pipe segments which are most prone to failure. The algorithm can then be executed with incrementally higher values of S to identify clusters with lower severity densities.

Table 8.3 Defect clusters containing structural defects with (S = 0.1m (0.3ft))

Cluster Size (i.e., number of defects)	Number of Occurrences	Average Cluster Severity Density (grade/meter)	Maximum Cluster Severity Density (grade/meter)	Minimum Cluster Severity Density (grade/meter)
2	22	25.12	33.90	20.00
3	6	29.50	33.82	21.10

Table 8.4 Defect clusters containing structural defects with (S = 0.3m (1ft))

Cluster Size (i.e., number of defects)	Number of Occurrences	Average Cluster Severity Density (grade/meter)	Maximum Cluster Severity Density (grade/meter)	Minimum Cluster Severity Density (grade/meter)
2	125	10.31	20.15	6.66
3	56	13.17	25.10	5.30
4	13	13.98	35.19	5.37
5	4	29.19	36.10	20.9

Table 8.5 Defect clusters containing structural defects with (S = 0.9m (3ft))

Cluster Size (i.e., number of defects)	Number of Occurrences	Average Cluster Severity Density (grade/meter)	Maximum Cluster Severity Density (grade/meter)	Minimum Cluster Severity Density (grade/meter)
2	304	7.22	20.15	2.11
3	153	7.68	25.10	2.36
4	82	9.31	32.16	3.07
5	47	9.54	36.14	3.14
6	25	8.71	33.00	3.83
7	5	9.74	16.10	6.44
8	5	4.92	5.54	3.99
9	2	9.14	12.05	6.30
10	1	17.59	17.59	17.59
11	1	9.17	9.17	9.17
12	1	4.75	4.75	4.75
16	1	8.48	8.48	8.48

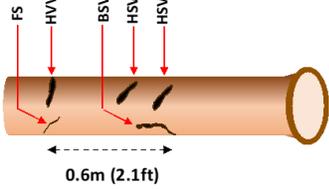
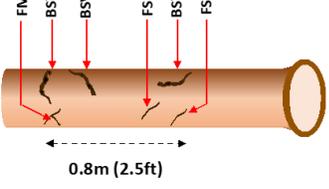
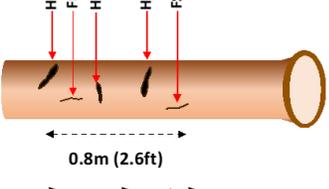
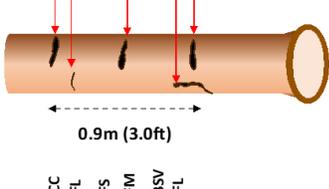
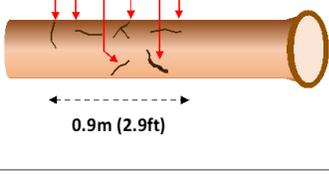
Cluster Severity (grade)	Length of Cluster (m)	Cluster Severity Density (grade/m)	Visualization of Cluster
23	0.64	36.1	
25	0.76	33.0	
21	0.79	26.73	
21	0.91	23.1	
20	0.88	20.9	

Figure 8.7 Visualization of the five clusters with the highest cluster severity score

DCA is not intended to be used as the sole determiner of a pipe’s likelihood of failure. The defect clusters should be analyzed alongside contextual information of the pipe, such as soil profiles or ground water table levels. To facilitate such analyses, a prototype web-based graphical user interface (GUI) tool was created to provide a visual depiction of clusters on a pipeline system map. The tool takes inspection spreadsheets and map files containing the coordinates of pipes as input and plots the identified defect clusters on the pipeline system map (see Figure 8.8). At the current stage, the tool allows this plot to be overlaid with information about soil profiles enabling the identification of potential failure scenarios (see Figure 8.8b). For instance, a high severity cluster which is located in a region of sandy soil could pose a threat of void formation over the pipe, leading to the formation of sinkholes. Hence, by incorporating contextual information sources with the identification of clusters, the web-tool enables a comprehensive analysis of pipe condition. Future work will focus on incorporating additional sources of contextual information, such as the locations of trees, ground water table levels, soil corrosivity, etc., into the web-tool.

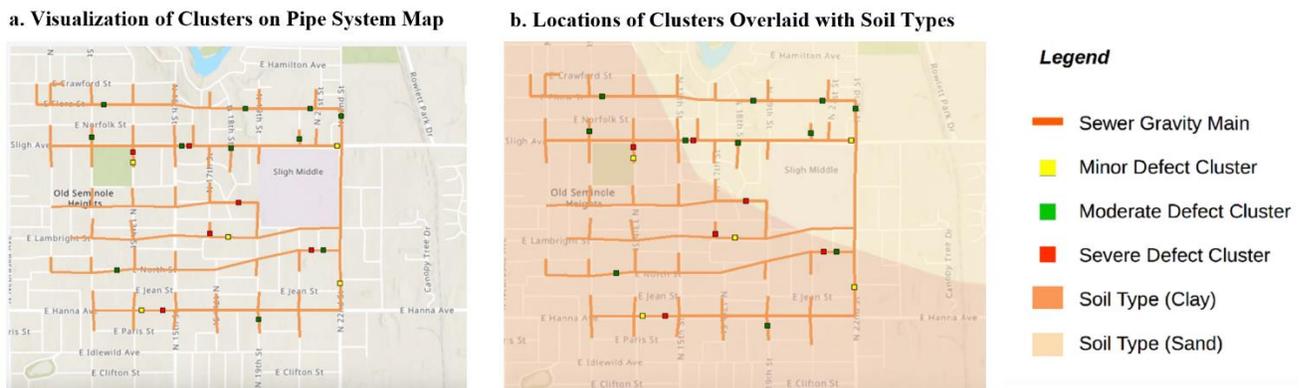


Figure 8.8 Example cluster visualizations generated by the web-tool

8.3.2 Defect Co-Occurrence Mining

The proposed method for calculating association rules (as described in Section 8.2.2) was also applied to the structural defects in the dataset. Based on discussions with subject matter experts, four key association rules were identified and are listed in Table 8.5. These rules have *confidence* > 0.5 and *lift* > 1.

Table 8.6 Association rules with confidence greater than 0.5 and lift greater than 1

Rule (X→Y)	Number of Pipes with X	Number of Pipes with X and Y	Confidence	Lift	Support
FL→FM	421	304	0.72	1.81	0.042
FS→FM	294	199	0.68	1.70	0.028
JOL→FM	69	36	0.52	1.31	0.004

The first rule states that if a pipe contains a longitudinal fracture (grade 3 defect), then the probability that the pipe also contains a multiple fracture (grade 4 defect) is 72%. The lift value, which is 1.81 (>1) indicates that these two defects occur more frequently than expected. Had the defects been randomly distributed among the pipe segments, the expected number of occurrences of longitudinal fracture and multiple fracture occurring simultaneously would have been 168. However, there were 304 observed co-occurrences of longitudinal fracture and multiple fracture. The second rule states that if a pipe contains a spiral fracture (grade 3 defect), then the probability that the pipe also contains a multiple fracture (a grade 4 defect) is 68%. The lift value, which is 1.70 (>1) indicates that these two defects occur together more frequently than expected. Had the defects been randomly distributed among the pipe segments, the expected number of occurrences of spiral fracture and multiple fracture occurring simultaneously would have been 117. However, there were 199 observed co-occurrences of fracture spiral and fracture multiple. The third rule states that if a pipe contains a joint offset (large), which is a grade 2 defect, then the probability that the pipe also contains a multiple fracture (a grade 4 defect) is 52%. The lift value, which is 1.31 (>1) indicates that these two defects occur together more frequently than expected. Had the defects been randomly distributed among the pipe segments, the expected number of pipes in which joint offsets and multiple fracture occur simultaneously would have been 27. However, there were 37 observed co-occurrences of joint offset large and fracture multiple.

When considered together, the three rules indicate that given the presence of a less severe defect (i.e., FL, FS, or JOL), the propensity of the pipe to also contain a defect of higher severity (i.e., FM) increases. These rules could indicate that the two types of defects may have a common underlying cause and could be used by asset managers to determine maintenance activities that consider the presence of frequently occurring pairs of defects.

8.3.3 Validation of the Approach

Validation of the approach involved collecting feedback about the assumptions, methodologies, and findings of the study from nine subject matter experts (SMEs) representing eight different organizations (i.e., Brown Equipment Company, CTSpec, Evanco Environmental, Greely and Hansen, Hazen and Sawyer, HDR, Hydromax USA, and SewerAI). Each of the nine SMEs had prior work experience in the area of sewer condition assessment, ranging from 3.5 years to 49 years, with the work average experience being 16.5 years. The validation process included face-to-face meetings as well as electronic communications via email, during which the DCA and defect co-occurrence mining approaches were explained to the SMEs in detail. The overall methodology was presented to the SMEs in the form of four themes: methodological assumptions, methodological approach, outputs and findings, and operational validity. Table 8.7 lists the components of the methodology that were validated by the SMEs.

Table 8.7 Validation scores based on assessment by SMEs

Validation Component	Validation Features	Average Score* (Standard Deviation)	Median Score*
Methodological Assumptions	Pipes with closely spaced defects are more likely to fail than pipes with the same defects not closely spaced.	4.0 (0.7)	4
	Defects that are in proximity could propagate into more severe defects; multiple defects in proximity can lead to soil infiltration and sinkholes; and multiple closely spaced defects lead to a localized region of weakness.	4.1 (0.9)	4
	The identification of defect clusters and their cluster severity densities could provide asset managers with useful information to determine repair/rehabilitation actions for pipelines.	3.9 (0.9)	4
Methodological Approach	A new metric to gauge the closeness of defects (in addition to their severities) would be beneficial to sewer asset managers.	4.2 (0.8)	4
	The cluster severity density (defined as the sum of defect grades in a cluster divided by the length of the cluster) represents a metric that can be used to gauge the closeness and severities of defects.	3.6 (0.5)	4
Outputs and Findings	The clusters depicted in Figure 8.8, represent pipe segments that could have a high propensity for failure.	3.9 (0.9)	4
	The identified co-occurrence characteristics among defects (shown in Table 8.7) could be used to prioritize pipes for inspection and/or maintenance.	3.9 (0.9)	4
Operational Validity	The information presented in the web-tool (i.e., the locations and severities of defect clusters plotted on a pipeline system map), could be beneficial in planning repair/rehabilitation.	4.0 (0.9)	4
	The graphical/animation outputs of the web-tool provide asset managers with a convenient method for identifying highly deteriorated regions.	4.2 (1.2)	5

* 1: Strongly Disagree, 2: Disagree, 3: Neutral, 4: Agree, 5: Strongly Agree

From the SMEs' evaluation of the methodological assumptions it is evident that they supported the assumptions underpinning this study, i.e., all other factors being equal, pipes with closely spaced defects are more likely to fail than pipes with defects that are spaced further apart. The SMEs recommended having separate classification systems based on the pipe material, because the modes of failure of rigid pipes (e.g., clay and concrete) are different from those of flexible pipes (e.g., polyvinyl chloride and high-density polyethylene). For instance, rigid pipes tend to develop visible cracks, whereas flexible pipes tend to deflect and fail abruptly. At the current stage, the methodology presented in this study (as well as the experimental results) apply only to rigid pipes. Future work will seek to develop a method of analysis for flexible pipes.

The SMEs agreed about the need for creating a metric to gauge the proximity of defects (in addition to their severities) and the SMEs agreed that at the current stage, the cluster severity density could be used to quickly identify pipes with multiple severe defects in proximity to one another. However, they suggested that additional considerations related to pipe material, pipe diameter, and soil type should be considered when creating such a proximity metric, since in their experiences pipe-soil interactions play a crucial role in determining the likelihood of failure. Hence, a future direction of this research would explore the creation of a metric which accounts for the pipe-soil interactions.

The output of the methodology, i.e., the clusters sorted according to their cluster severity densities (see Figure 8.8), was also judged to be of benefit to the asset management practice. The SMEs believed that the clusters identified using the DCA approach (see Figure 8.7) represented highly deteriorated pipe sections which could face an imminent threat of failure due to the large concentration of defects. They also agreed that identifying and quantifying the severity of the clusters would be useful for making repair/rehabilitation decisions. The SMEs agreed about the usefulness of identifying co-occurrence characteristics among defects in supporting inspection and maintenance prioritization.

Finally, the SMEs recognize the value provided by the web-tool for identifying defect clusters. They agreed about the usefulness of the graphical/animation outputs in identifying highly deteriorated pipes and in making repair/rehabilitation activities. However, one SME pointed out

that requiring users to manually upload CSV files increases the potential for errors in data handling (e.g., incorrect file uploads). According to that SME, CSV files could be used during initial tests, however, the web-tool should be integrated into GIS eventually.

8.3.4 Conclusions and Chapter Summary

Due to the rise of rapid sewer inspection technologies, the amount of pipe condition data is likely to grow at an unprecedented pace. Many municipalities in the US have begun sharing information about the condition of sewer pipes through initiatives such as ‘Data.gov’, ushering in the creation of large datasets of publicly accessible pipeline condition information. This chapter discussed the development of techniques to mine such large datasets of pipe condition information to reveal insights that can be used to guide maintenance prioritization efforts. Specifically, this chapter develops two novel techniques for pipeline deterioration, i.e., DCA and defect co-occurrence mining, which consider spatial information of defects, such as the locations, densities, and co-occurrence characteristics of defects in pipes.

The DCA approach introduces the concept of defect clusters (i.e., pipe regions that contain multiple defects in close proximity) and develops an algorithm (with linear execution time) to identify defect clusters in sewer inspection databases. A web-based GUI tool was created to provide a visual depiction of clusters on a pipeline system map and to assist asset managers in identifying clusters of various severity levels. The proposed approach was evaluated on a dataset of 7193 inspections of 8-in, 10-in, and 12-in diameter VCP sewers from South Carolina, Florida, and Ohio and led to the identification of 627 defect clusters that contained two or more structural defects. Among the 627 clusters, the five clusters with the highest severity densities were presented to nine subject matter experts—who judged the clusters to pose imminent threat of failure due to high concentrations of defects. In the absence of the DCA approach, these high severity clusters would have most likely been overlooked. When combined with contextual information, DCA may provide additional insights, such as the likelihood of void formation due to sand infiltration. Additionally, given inspection data pertaining to the same pipe across multiple time frames, the progression of defects in clusters could be analyzed facilitating the calculation of pipe deterioration rates. The identification of defect clusters could also inform rehabilitation decisions. For instance,

the identification of defect clusters could lead to insights into whether whole length rehabilitation or local patch repairs should be pursued.

The defect co-occurrence mining approach identifies defects which occur frequently together. Evaluating the approach on the dataset of 7193 pipe inspections led to the insight that pipes which contained longitudinal fracture defects, spiral fracture defects, or large joint offsets exhibited a greater propensity to also contain multiple fracture defects. When combined with contextual information (e.g., pipe age, pipe location, soil types, etc.), the co-occurrence characteristics could highlight common underlying causes for deterioration and further our understanding of sewer pipe deterioration. For example, the approach can be extended to analyze co-occurrences between external factors, such as the presence of trees and heavy industries, and specific types of defects in pipes. Considered together, the two approaches (i.e., DCA and co-occurrence mining) address the limitations of existing deterioration modeling approaches (i.e., the lack of consideration to spatial information about defects) and could provide new insights in pipe asset management and rehabilitation decision-making.

CHAPTER 9. CONCLUSIONS AND RECOMMENDATIONS

Municipalities across the U.S. rely on visual CCTV inspections to assess the condition of sewer pipelines. The tedious and subjective nature of CCTV inspections limits the consistency and accuracy of data collected. Additionally, operator fatigue due to lengthy inspection sessions could lead to erroneous assessment of sewer condition. However, CCTV inspections only provide a snapshot about the condition of pipes at a particular instance of time. Asset managers use deterioration models to predict the future condition of pipes which is essential for developing long-term maintenance, renewal, and rehabilitation plans. However, current sewer deterioration modeling approaches do not account for spatial information about defects, which often play a crucial role in determining a pipe's likelihood of failure. The first two objectives of this research address the challenge of improving the defect interpretation consistency and speed of sewer CCTV inspections by developing algorithms to facilitate automated defect interpretation and autonomous navigation. The third objective of this research attempts to develop approaches for assessing sewer deterioration by analyzing spatial information about defects in pipes. The first section of this chapter summarizes the methodologies, findings, and limitations of this study. The second and third section of this chapter discusses the study's technical and practical contributions, respectively. Finally, the fourth section recommends potential future research directions.

9.1 Summary of the Research

Table 9.1 lists the methodologies, datasets, findings, and limitations of the various components of this research study. Prior research on automated defect identification used feature extraction and morphological methods, leading to poor generalization capabilities and low defect classification accuracies. To improve upon the generalization capabilities of these previous approaches, the automated framework proposed in Chapter 3 utilized deep neural networks trained on a set of 12,000 images from over 200 sewer pipeline inspections. The data were obtained from 8-in, 10-in, and 12-in DIP, PCCP, and VCP sewers located in Georgia and California. The proposed framework passes images through multiple binary CNNs, each trained to classify a particular class of defect, in order to identify multiple types of defects. To reduce overfitting, the images in the training set were augmented by inducing changes in brightness, contrast and motion blur. The

automated system was evaluated on the basis of classifying root intrusions, deposits, and cracks in 2000 CCTV images, and yielded 86.2% classification accuracy, 87.7% precision, and 90.6% recall on this dataset. This approach did not facilitate localization of defects in images, which is a necessary first step in determining the circumferential location of defects in pipes.

In Chapter 4, the development of a deep-learning-based framework for the classification and localization of sewer defects was discussed. Three state-of-the-art object detection models (i.e., SSD, YOLO, and Faster R-CNN) were evaluated for their speed and mean average precision in detecting sewer defects (note: object detection is a regression task and the accuracy of such models are represented using mean average precision instead of classification accuracy, precision, and recall). The three models were evaluated in the context of detecting root intrusions and deposits, since the presence of these two defects plays a crucial role in determining maintenance activities for sewers. Additionally, root intrusions and deposits, when left untreated, could progress into more severe structural defects. Each model was trained using 3,420 images (i.e., 3,040 images in the training set and 380 images in the validation set) and subsequently tested on 380 images that were not present in the training or validation sets. The Faster R-CNN model was found to yield the highest accuracy of detecting root intrusions and defects in CCTV images, with a mAP 0.718 on the testing set (mAP of YOLO and SSD were 0.695 and 0.530, respectively). However, the Faster R-CNN was also the slowest model, requiring approximately 110 ms to process each image, when evaluated using an Nvidia P4000 Quadro CUDA GPU (processing time for YOLO and SSD were 57 ms and 33 ms, respectively). To test automated defect detection capabilities in practice, a prototype system was developed for detecting root intrusions and deposits and evaluated on inspection videos televising 335 meters of sewer laterals. The objective of this evaluation was to determine the number of defects that could be correctly detected and the number of false positives, when automated defect detection was performed on actual CCTV videos. Hence, the Faster R-CNN was used as the underlying model, since it had the highest mAP among the previously three evaluated models. The prototype system detected 51 out of 56 instances of root intrusions and deposits and generated seven false positives. A limitation of this method was that it applied the Faster R-CNN model to every image frame of a CCTV video, even though only few images contained defects. As a result, this method results in unnecessary computations on images that do not contain defects.

Chapter 5 discussed the development of a two-step framework to improve the speed of defect detection in sewer CCTV videos. The two-step framework (1) uses a CNN to classify whether images contain defects or not, and (2) subsequently applies a Faster R-CNN only to the images that contain defects. The two-step framework was evaluated on the basis of detecting cracks/fractures, root intrusions, and lateral connections in 10 videos of 8-inch diameter VCP sewer mains (from Alabama and Ohio) which have a total length of 2200 feet. The two-step framework correctly detected 112 out of 124 (90.3%) instances of cracks/fractures, 88 out of 98 (89.8%) instances of root intrusions, and 54 out of 59 (92%) instances of lateral connections in the videos. The two-step framework also led to 45 false positive detections of cracks/fractures, 29 false positive detections of root intrusions, and one false positive detection of lateral connections. The lack of interpretability of CNNs makes diagnosing these errors difficult, resulting in a loss of generalization capability (i.e., CNNs may produce unexpected results when exposed to edge cases).

Hence, Chapter 6 discussed the use of CNN interpretation techniques to improve the generalization capability of automated defect identification models. Specifically, a CNN interpretation technique called CAM was used to facilitate an ‘under-the-hood’ analysis of a ResNet34 CNN classifier and guide the development of more generalizable models. Note, the ResNet34 was selected since it has been demonstrated to achieve one of the highest image classification accuracies on benchmark datasets (He et al. 2015). A dataset of 12,000 pre-labeled images (i.e., images with defect labels imprinted on them) were used for training a CNN model. The accuracy of the model was then evaluated on 2,400 unlabeled images (i.e., images without defect labels imprinted on them). An analysis of the heatmaps generated by CAM suggested that CNNs trained using pre-labeled images learned to recognize the defect labels imprinted on the images rather than the defect pixels. However, our study showed that the sensitivity of the ResNet 34 CNN classifier to defect labels could be minimized by augmenting the training images with random rotations and horizontal flips. We hypothesize that by incorporating random rotations in the images, the labels would not be confined to the same locations in the images, resulting in an increased difficulty for the CNNs to learn the labels. The automated defect detection framework that was developed in this study, takes RGB images as input (i.e., images that contain red, green, and blue channels). Using monochromatic images as input (i.e., images that contain single channel) instead of RGB images could lead to improved processing speeds, since monochromatic images are one third the size of

RGB images. However, the loss of color information, from using monochromatic images, could also lead to reduced defect detection accuracies. Hence, future work could quantify the tradeoffs between speed and defect detection accuracy, when using monochromatic images as input to the automated system. The current automated system does not address defect tracking, i.e., identifying and counting unique defects. Future development efforts could incorporate tracking algorithms such as Simple Online and Realtime Tracking (SORT) to count individual defects in videos.

Chapter 7 discussed the development of a vision-based method for estimating the orientation of sewer CCTV cameras in pipes. The proposed method takes a CCTV video as input and uses deep learning-based object detection and optical flow to classify the camera orientation into five categories (i.e., left-wall view, left-wall/forward view, forward view, right-wall/forward view, and right-wall view). The estimated camera orientation, which is an output of this method, could allow for autonomous sewer navigation, by enabling inspection robots to estimate their trajectories and take corrective actions while traversing pipes. Additionally, the proposed method could facilitate automated circumferential localization of sewer defects, which has not addressed in prior research. The proposed method was evaluated on 500 image frames from 10 CCTV inspections videos of 8-inch diameter VCP sewers from Florida and Ohio. It was found that 79.8% of the image frames were correctly classified, meaning that the orientation estimates produced by the proposed method matched the manually estimated orientation. An additional 8.2% of the evaluated frames were ‘soft matches’, that is, the orientation estimate produced by the automated method differed from the manually estimated orientation by one class. A limitation of this method is that it cannot address rotations along the forward axis of a camera, which result in tilted image frames (see Figure 7.13). This limitation can be addressed by expanding the current approach to analyze the position of water flow lines in images to detect such rotations. Although the focus of this study was to develop algorithms for CCTV inspection robots, the proposed approaches and techniques can be expanded to facilitate automated detection and navigation in multi-sensory inspection technologies, such as the SSET and PANORAMO.

Given the emergence of rapid sewer inspection technologies, access to pipe condition data is likely to grow at an unprecedented pace. Additionally, many municipalities in the US have begun sharing information about their pipelines through initiatives such as ‘Data.gov’, leading to the creation of

large datasets of publicly accessible sewer pipeline condition information. Hence, Chapter 8 discussed the development of techniques to mine these large datasets and discover insights that can be used to guide maintenance prioritization efforts. Specifically, this chapter discusses a novel methodology for assessing sewer deterioration by incorporating spatial information such as the locations, densities, and co-occurrence characteristics of defects in pipes. A methodology called DCA was developed to mine sewer inspection records and identify pipe segments which contain multiple defects in close proximity. A web-based GUI tool was created to provide a visual depiction of clusters on a pipeline system map and to assist asset managers in identifying clusters of various severity grades. Additionally, an approach to mine co-occurrence characteristics among defects was also introduced (i.e., identification of defects which occur frequently together). Together the two approaches (i.e., DCA and co-occurrence mining) address the limitations of existing deterioration modeling approaches (i.e., the lack of consideration to spatial information about defects) and could provide new insights in pipe asset management and rehabilitation decision-making.

Table 9.1 Summary of methodologies, datasets, findings, and limitations

Research Component	Methodology and Datasets	Results and Findings	Limitations
CNN-based Automated Defect Classification (Chapter 3)	<ul style="list-style-type: none"> An ensemble of binary CNNs were trained to classify root intrusions, deposits, and cracks Number of images used for training, validation, and testing are 7500, 2500, and 2000, respectively Images originate from 8-in, 10-in, and 12-in diameter DIP, PCCP, and VCP sewers from Georgia and California 	<ul style="list-style-type: none"> Using an ensemble of CNNs enables multi-defect classification, i.e., multiple categories of defects can be identified in the same image The classification accuracy on the testing set was 90.9%, 86.0%, and 84.0% for root intrusions, deposits, and cracks, respectively 	<ul style="list-style-type: none"> Visual similarities between the silhouettes of defects lead to misclassification errors (see Figure 3.10) The proposed framework cannot address defect localization in images The framework was evaluated on 1000 images of defects and 1000 images without defects, however, in CCTV videos, the number of image frames without defects far exceeds the number of defect frames (Meijer 2019)
CNN-based Automated Defect Detection (Chapter 4)	<ul style="list-style-type: none"> Three object detection models i.e., SSD, YOLO, and Faster R-CNN were evaluated for speed and mAP in detecting root intrusions and deposits Number of images used for training, validation, and testing are 3040, 380, and 380, respectively Images originate from 8-in, 10-in, and 12-in diameter PVC, PCCP, and VCP sewers from Virginia and Ohio The defect detection capabilities of the models were evaluated on CCTV videos from 8-inch diameter VCP sewers (with a total length of 1100 feet) 	<ul style="list-style-type: none"> Proposed approach can facilitate defect localization in images, which is a necessary first step for automated circumferential location identification of defects The Faster R-CNN model yielded the highest accuracy with a mAP of 71.8% compared to the SSD (53.0% mAP) and YOLO (69.5% mAP) The trained Faster R-CNN model could correctly detect 51 out of 56 known instances of defects and generated 7 false positives when inferred on the three CCTV videos 	<ul style="list-style-type: none"> Inconsistencies in the annotation of ground-truth images lead to biased models (see Figure 4.9) The proposed framework applies the Faster R-CNN to every image frame, although only a small fraction of the images contains defects, resulting in unnecessary computations

Table 9.1 continued

<p>Two-Step Defect Detection Framework (Chapter 5)</p>	<ul style="list-style-type: none"> • A two-step framework consisting of a ResNet34 image classifier in the first step and a Faster R-CNN in the second step, was developed to facilitate defect detection in CCTV videos • The framework was trained using a dataset consisting of 30,000 images from 8-inch and 10-inch diameter VCP sewers in Florida, Georgia, and Ohio • The defect detection capabilities of the framework were evaluated on 10 CCTV videos from 8-inch diameter VCP sewers (with a total length of 2200 feet) 	<ul style="list-style-type: none"> • The proposed framework correctly detected 112 out of 124 (i.e., 90.3%) instances of cracks/fractures, 88 out of 98 (i.e., 89.8%) instances of root intrusions, and 54 out of 59 (i.e., 92%) instances of lateral connections, in the CCTV videos • 45 false positive detections of cracks/fractures, 29 false positive detections of root intrusions, and 1 false positive detection of lateral connections, were generated 	<ul style="list-style-type: none"> • Due to the large number of parameters and complex interconnections of the underlying Faster R-CNN and ResNet34 models, it is difficult to diagnose the reason behind false positives and other misclassification errors • Although the proposed framework facilitates localization of defects in images, it falls short of identifying the circumferential location of defects in images
<p>CNN Interpretation Techniques (Chapter 6)</p>	<ul style="list-style-type: none"> • A CNN visualization technique called CAM was used as a diagnostic tool, to guide the development of an automated defect interpretation system • 12,000 pre-labeled images, which contained defect labels imprinted on them were used for training, whereas 2400 images without imprinted labels were used for evaluating the defect classification accuracy of a ResNet34 CNN 	<ul style="list-style-type: none"> • Models trained using the pre-labeled images resulted in high training and validation accuracies, however, the accuracies when evaluated on the un-labeled testing set were significantly lower • An analysis of the CAM outputs indicated that the models learned to ‘cheat’ by recognizing the pixels corresponding to the labels, rather than the defect pixels • Augmenting the training images by inducing random rotations and horizontal flips minimized the tendency of the CNNs to recognize the labels 	<ul style="list-style-type: none"> • CAM cannot be directly applied to object detection models such as Faster R-CNN, which do not contain a global average pooling layer • Future work could consider incorporating methods such as image occlusion sensitivity for interpreting object detection models

Table 9.1 continued

<p>Vision-based Orientation Estimation of CCTV Robots for Defect Localization and Autonomous Navigation (Chapter 7)</p>	<ul style="list-style-type: none"> • A vision-based method for estimating the orientation of CCTV cameras in sewer pipes was developed to facilitate autonomous sewer navigation and the identification of circumferential position of defects • The framework uses deep learning-based object detection in conjunction with optical flow, to classify the orientation of cameras (in CCTV videos) into five categories (i.e., left-wall view, left-wall/forward view, forward view, right-wall/forward view, and right-wall view) • Five hundred images from 10 CCTV videos of 8-in diameter sewers in Florida and Ohio were used for evaluating the accuracy of the proposed method 	<ul style="list-style-type: none"> • The proposed method correctly classified the orientation of 79.8% of the image frames • An additional 8.2% of the evaluated frames were ‘soft matches’, i.e., the orientation estimate produced by the automated method differed from the manually estimated orientation by one class 	<ul style="list-style-type: none"> • At the current stage, the proposed method classifies the orientations of cameras into five categories, but cannot estimate the camera orientation in degrees • Image frames that contained rapid brightness changes due to the reflection of camera light on pipe walls, led to errors in computation of optical flow vectors
<p>A Framework for Mining Spatial Characteristics of Sewer Defects from Inspection Databases (Chapter 8)</p>	<ul style="list-style-type: none"> • Two approaches (i.e., DCA and defect co-occurrence mining), which incorporate spatial information about defects in pipes, were developed • A web-based GUI tool was created to provide a visual depiction of clusters on a pipeline system map and to compare clusters against contextual data • The proposed approaches were evaluated on a dataset of 7193 inspections of 8-in, 10-in, and 12-in diameter VCP sewers from South Carolina, Florida, and Ohio 	<ul style="list-style-type: none"> • The DCA approach identified 627 clusters that contained two or more structural defects—the longest cluster had 16 defects • The five highest severity density clusters were presented to nine subject matter experts, who judged the clusters to pose imminent threat of failure • The DCA approach led to the insights that pipes which contained longitudinal fracture defects, spiral fracture defects, or large joint offsets exhibited a greater propensity to also contain multiple fracture defects 	<ul style="list-style-type: none"> • At the current stage, the proposed approaches do not account for the complexities of pipe-soil mechanics when evaluating defect clusters and co-occurrences • Future work could track the progression of defect clusters over time in order to determine the rate of pipe deterioration and predict the remaining service life of pipes

9.2 Contributions to the Body of Knowledge

There is a considerable body of previous research, which uses feature engineering methods for automated classification of defects in sewer pipes. The previously used methods include (1) edge detection and Fourier descriptors (Xu et al. 1998, Moselhi and Shehab 2000), (2) discrete wavelet transforms (Yang and Su 2008), (3) scale invariant feature transform (SIFT) (Guo et al. 2009), morphological segmentation (Su et al. 2011, Hawari et al. 2018), and histograms of oriented gradients (HOG) (Halfawy and Hengmeechai 2014, Halfawy and Hengmeechai 2015, Moradi and Zayed 2017). However, the automated systems proposed in these studies are constrained by the low generalization capabilities (i.e., the ability to classify images that exhibit significant variations in shape, color, texture, illumination, etc.) of the underlying feature engineering methods, and lead to low defect identification accuracies when applied to sewer CCTV images. Furthermore, the automated systems proposed in these prior studies could not identify multiple categories of defects in the same image (for instance, the simultaneous identification of the presence of roots and deposits). The approach in Chapter 3 leveraged CNNs for automated defect identification in sewer CCTV images. CNNs significantly outperform the generalization capabilities of feature engineering methods (LeCun et al. 2015) and are thus better equipped to deal with sewer CCTV images. To the best of our knowledge, ours was the first study to demonstrate the superior accuracy of CNNs over feature engineering methods in automated sewer CCTV defect identification. Additionally, by using an ensemble of CNNs our approach facilitates identification of multiple defect categories in images, which was not addressed by prior studies.

Building upon the initial successes of CNNs in sewer CCTV defect classification, Chapter 4 explored the use of CNN-based object detection models to classify and localize defects in sewer CCTV images. Defect localization is a necessary first step for identifying the circumferential locations of defects in pipes and was partially addressed by Cheng and Wang (2018), who evaluated the detection capabilities of the Faster R-CNN model on approximately 180 unique images of defects without cross-validation. Our study extended the discussion presented in Cheng and Wang (2018), by evaluating the speed and accuracy of three state-of-the-art object detection models (i.e., SSD, YOLO, and Faster R-CNN) in the context of sewer defect detection, and by cross-validating the tests to minimize sampling biases. Our study found that although the Faster R-CNN model yielded the highest accuracy, it also emerged as the slowest model among the three

and led to several false positive detections. This study also revealed that applying object detection models to every image frame of a sewer CCTV video resulted in unnecessary computations and a large number of false positives, because only a small fraction (i.e., approximately 10%) of the image frames contain sewer defects. Hence, Chapter 5 developed a prototype automated defect detection system that utilized a two-step framework to improve the speed and accuracy of defect detection in CCTV videos. The framework uses a CNN to classify whether images contain defects, and only passes the defect images to the object detector, facilitating increased processing speeds and lower false positive detections.

One of the challenges in training CNNs for defect identification is their lack of interpretability. These interpretation challenges result in the loss of generalization capability, that is, the algorithms may produce unexpected results when exposed to edge cases. This issue is prevalent in CNNs used for automated sewer defect identification since the training and testing images differ significantly. For example, municipalities typically maintain large numbers (i.e., several thousands) of labeled sewer defect images as an outcome of their sewer inspection programs; however, these images typically contain the defect labels imprinted on the images. Cropping the labels out or obscuring them is manually cumbersome and typically results in loss of information from the images. Hence, CNNs trained with these labeled images may exhibit unusual behavior, such as being sensitive to defect labels and other markers in the images and may result in significantly lower classification accuracies when tested on unlabeled images. Chapter 6 demonstrates how a visualization technique called CAM can be used as a diagnostic tool to improve the generalizability of CNN models. Our study showed that CNNs trained with labeled images (i.e., image with defect labels imprinted on them) were sensitive to the defect labels. However, augmenting these labeled images by inducing random rotations forced the CNNs to learn the defect pixels rather than the image labels, resulting in improved generalization capabilities. To the best of our knowledge, ours was the first study to use CNN interpretation techniques to improve the accuracy of automated sewer defect identification.

Automated circumferential localization of defects in sewer CCTV videos remains a crucial yet unaddressed problem. Determining the circumferential location of defect is challenging because the orientation of sewer CCTV cameras in pipes is typically unknown. Chapter 7 proposed a

vision-based method to estimate the orientation of sewer CCTV cameras and paves the way for automated identification of circumferential location of defects. This method also has the potential added benefit of facilitating autonomous navigation of sewer CCTV robots in pipes. A considerable body of research exists in the development of autonomous navigation systems for multi-sensor sewer robots. However, most approaches use depth cameras, lasers, LiDAR, or orientation sensors to determine the location and orientation of robots in pipes—sensors which are not typically found on sewer CCTV inspection robots. The method proposed in this study, uses information from a single camera to estimate the orientation of a camera, thereby enabling the development of autonomous navigation systems for sewer CCTV robots.

Chapter 8 introduces two new approaches: DCA and defect co-occurrence mining. These methods consider spatial information about defects when analyzing sewer deterioration. The DCA approach introduces the concept of defect clusters (i.e., pipe regions with multiple defects in close proximity) and develops an algorithm to identify defect clusters in sewer inspection databases. When combined with contextual information, DCA may provide additional insights, such as the likelihood of void formation due to sand infiltration. The DCA approach could be applied periodically to determine how the distribution of defects and their clustering progresses over time and reveal pipeline deterioration patterns. The defect co-occurrence mining approach could highlight groups of defects which occur frequently together in pipes. When combined with contextual information (e.g., pipe age, pipe location, soil types, etc.), the co-occurrence characteristics could highlight common underlying causes for deterioration and further our understanding of sewer pipe deterioration. The information obtained from co-occurrence mining of defects could be used as apriori knowledge for automated defect detection algorithms. This apriori information could reduce the search space of the algorithms in order to facilitate a more efficient implementation of automated sewer defect detection.

9.3 Contributions to the Body of Practice

The research presented in Chapters 3, 4, 5, 6, and 7 develops algorithms that offer important advancements to the current state of automated sewer defect interpretation. Automated defect interpretation, as proposed in this study, could enhance the manner in which CCTV inspections are currently conducted. For instance, CCTV operators typically code the defects during inspection,

while simultaneously navigating the inspection crawler in pipes. This practice requires operators to inefficiently stop the crawler to enter a defect code every time a defect is encountered during inspection, incurring additional time in the field. As a result, CCTV operators are gradually adopting a ‘inspect in the field, code in the office’ approach, whereby the coding of defects is performed after inspection, rather than during inspection. Automated defect interpretation would enable simultaneous inspection and coding of the pipes, without requiring operators to stop the crawler during inspection. This approach could significantly reduce the time spent by inspectors in the field, while enabling improving the speed and efficiency of inspections. The current NASSCO PACP protocol recommends the speed of inspection crawlers to be 1 foot/sec, which is based on human operators’ response times. However, if videos were to be interpreted using computer processes, it could be feasible to inspect pipes at much higher speeds.

Asset managers are typically concerned with identifying major defects in pipes, i.e., structural defects which are classified as level 4 or level 5 according to the NASSCO PACP. Examples of such defects include: Collapsed Section, Deformed Section, Broken Pipe, Hole, Hinge Fracture, etc. However, the NASSCO PACP protocol requires inspectors to also report the locations of level 1, level 2, and level 3 defects, which are often more frequent than level 4 or 5 defects. Hence, operators spend a significant amount of time coding minor defects, which are often overlooked when making repair/rehabilitation decisions. Algorithms for automated defect interpretation could be used to automatically code level 1, level 2, and level 3 defects, while assigning the more severe level 4 and level 5 defects to be coded by experienced human operators. The two-step framework proposed in Chapter 5 could be extended to accommodate a third step, which facilitates this approach (see Figure 9.1).

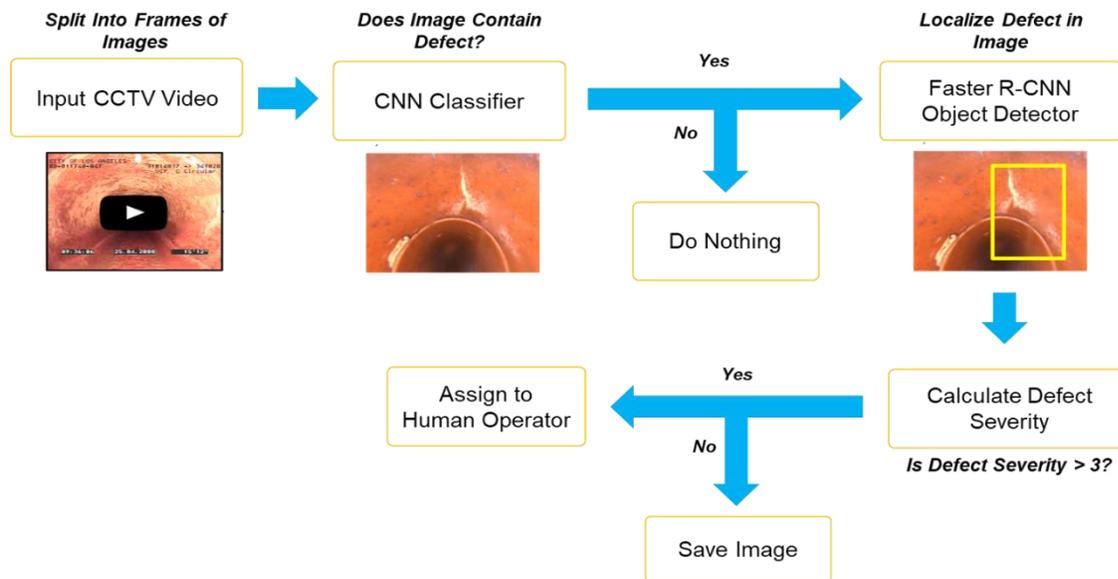


Figure 9.1 Framework for defect interpretation—assigning minor defects to computer processes and severe defects for human interpretation

Automated defect interpretation could also facilitate the processing of large volumes of historical inspection videos, which would not be practically feasible if performed manually. Information about the frequency, location, and severity of defects could be used to develop deterioration models to predict the future condition of sewers. For example, by comparing condition information from the same pipeline inspected at different instances of time, deterioration/defect progression rates can be calculated for various types of sewers.

The vision-based camera orientation estimation algorithm discussed in Chapter 7, leads to advancements in automated defect interpretation as well as autonomous robot navigation. could be integrated onboard UAVs or similar inspection technologies to facilitate rapid inspection and assessment of sewers. Furthermore, these technologies could be integrated into wireless sensor networks, paving the way for real-time condition monitoring of sewer infrastructure. The development of UAV-based sewer pipeline inspection systems is an emerging area of research (Rizzo et al. 2016) and the method proposed in this study could be integrated with such UAV systems, to facilitate rapid inspection and assessment of sewers at a fraction of the cost of current methods. Additionally, using a network of wireless sensors, such inspection technologies could relay information about the condition of pipes in real-time, which would enable quick decision-

making related to maintenance and repair. Furthermore, since the proposed method is vision-based, it may also find application in other GPS-denied environments, such as utility tunnels, boreholes, and underground mines.

The DCA algorithm presented in Chapter 8, provides a novel approach that can be used to consider defects and their proximity in a way that has previously not been possible. DCA could be used periodically to analyze how the distribution of defects and their clustering progresses with time and could reveal trends in pipe deterioration. The identification of defect clusters could also inform rehabilitation decisions. For instance, the identification of defect clusters could lead to insights into whether whole length rehabilitation or local patch repairs should be pursued. Furthermore, the mining of co-occurrence characteristics among sewer defects could highlight groups of defects which occur frequently together, allowing for the creation of customized maintenance plans. For instance, the identification of pipes which contain high water marks and fractured walls could indicate a high propensity for sewage exfiltration during sewer surges. Additionally, the identification of successive exfiltration and infiltration locations along a pipe could indicate the presence of external water flows that have the potential to compromise the pipe bedding, leading to sewer collapses.

9.4 Recommendations for Future Research

Based on the insights gained from this study, the exploration of the following three research themes are recommended: (1) autonomous UAVs for rapid inspection and monitoring of sewers, (2) data-driven prediction of water and sewer pipeline failures, and (3) smart, secure, and open source pipeline asset information model.

9.4.1 Autonomous UAVs for Rapid Inspection and Monitoring of Sewers

Current methods of sewer inspection rely on camera equipped robotic crawlers, which tend to be expensive and prone to getting stuck in pipes. Given the rapid advancements in UAVs and battery technologies, there is potential to develop pipeline inspection systems that leverage swarms of autonomous micro-UAVs (Rizzo et al. 2016). The better maneuverability and relatively low hardware costs of UAVs, in comparison to traditional camera crawlers, could facilitate rapid

assessment of pipelines at a fraction of the cost of current methods. Lightweight CNNs could be integrated onto the onboard processing units of these UAVs to facilitate automated defect detection on the edge. These UAVs could also be programmed to replicate the capabilities of traditional CCTV robots, such as panning and tilting to look inside sewer lateral connections. Furthermore, these inspection technologies could communicate using wireless sensor networks, paving the way for real-time condition monitoring of sewer infrastructure.

9.4.2 Data-driven Prediction of Water and Sewer Pipeline Failures

Pipeline deterioration is strongly influenced by contextual influences, such as trees, hydrology, and soil conditions. However, few studies have explored the effects of these factors, owing to a lack of publicly available datasets containing such information. Hence, there is potential to utilize data mining techniques to extract contextual information from disparate sources, such as satellite imagery and geological surveys, and investigate their impact on pipeline deterioration. For instance, computer vision could be used to identify the locations of trees and heavy industries from satellite imagery and map this information to existing pipeline condition databases. By considering this entire gamut of information, highly accurate deterioration prediction models can be developed to improve maintenance prioritization efforts.

9.4.3 Smart, Secure, and Open Source Pipeline Asset Information Model

There is a need for the creation of a smart, secure, and open source water and wastewater pipeline asset information model. The need for this research arises from the “data silo” problem faced by many municipalities in the US. Specifically, this problem refers to difficulties in accessing and sharing data due to the use of numerous incompatible software applications. Due to the existence of data silos, many municipalities are unable to utilize their data for predictive analytics and revert to antiquated asset management practices. The building construction management industry has addressed this problem by developing the Industry Foundation Classes (IFC) data model, which is an open-source format for representing building information. The development of a similar data model will be hugely beneficial for the water and wastewater industry since it would break down data silos and enable information sharing regardless of which software is being used.

REFERENCES

- Agrawal, R., and Srikant, R. (1994). "Fast algorithms for mining association rules." Proceedings of the 20th International Conference on Very Large Databases (VLDB), 487-499.
- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado G., Davis, A., Dean, J., Devin, M., and Ghemawat, S. (2016). "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." arXiv preprint arXiv:1603.04467.
- Ahmadi, M., Cherqui, F., De Massiac, J-C., and Le Gauffre, P. (2013). "Influence of available data on sewer inspection program efficiency." Urban Water Journal, 11(8), 641-656.
- Ahrary, A., Nassiraei, A., and Ishikawa, M. (2007). "A study of an autonomous mobile robot for a sewer inspection system." Artificial Life and Robotics, 11(1), 23-27.
- Alejo, D., Caballero, F., and Merino, L. (2019). "A robust localization system for inspection robots in sewer networks." Sensors, 19(22), 4946.
- Alejo, D., Mier, G., Marques, C., Caballero, F., Merino, L., and Alvito, P. (2020). "SIAR: A ground robot solution for semi-autonomous inspection of visitable sewers." Proceedings of Advances in Robotics Research: From Lab to Market, Springer, Cham.
- Association of Metropolitan Sewerage Agencies (AMSA) – currently National Association of Clean Water Agencies). (2003). "Wet weather survey-final report." <<http://www.nacwa.org/images/stories/public/WetWeatherFinalReport.pdf>> (November 20, 2018).
- American Society of Civil Engineers. (2017). "Report card for America's infrastructure." <<https://www.infrastructurereportcard.org/wp-content/uploads/2017/01/Wastewater-Final.pdf>> (August 1, 2017).
- American Water Works Association. (2012). "Buried no longer: confronting America's water infrastructure challenge." <<https://www.awwa.org/Portals/0/files/legreg/documents/BuriedNoLonger.pdf>> (August 1, 2017).
- Ana, E. V. (2009). "Sewer asset management - sewer structural deterioration modeling and multicriteria decision making in sewer rehabilitation projects prioritization." PhD thesis, Department of Hydrology and Hydraulic Engineering, Vrije Universiteit Brussels, Brussels.
- Ana, E. V. and Bauwens. W. (2010). "Modeling the structural deterioration of urban drainage pipes: the state-of-the-art in statistical methods." Urban Water Journal, 7(1), 47–59.
- Ariaratnam, S., El-Assaly, A. and Yang, Y. (2001). "Assessment of infrastructure inspection needs using logistic models." Journal of Infrastructure Systems, 7(4), 160–165.

- Atha, D. J., and Jahanshahi, M. R. (2017). "Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection." *Structural Health Monitoring*, 17(5), 1110-1128.
- Baah, K., Dubey, B., Harvey, R., and McBean, E. (2015). "A risk-based approach to sanitary sewer pipe asset management." *Science of the Total Environment*, 505, 1011-1017.
- Baur, R. and Herz, R. (2002). "Selective inspection planning with ageing forecast for sewer types", *Water Science and Technology*, 46(6), 389-396.
- Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). "Speeded-up robust features (SURF)." *Computer Vision and Image Understanding*, 110(3), 346-359.
- Bradski, G. (2000). "The OpenCV library." *Dr. Dobb's Journal of Software Tools*.
- Caradot, N., Rouault, P., Clemens, F., and Cherqui, F. (2018). "Evaluation of uncertainties in sewer condition assessment." *Structure and Infrastructure Engineering*, 14(2), 264-273.
- Caradot, N., Sonnenberg, H., Kropp, I., Ringe, A., Denhez, S., Hartmann, A., and Rouault P. (2017). "The relevance of sewer deterioration modelling to support asset management strategies." *Urban Water Journal*, 14(10), 1007-1015.
- Cha, Y. and Choi, W. (2017). "Deep learning-based crack damage detection using convolutional neural networks." *Computer Aided Civil and Infrastructure Engineering*, 32, 361-378.
- Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). "Return of the devil in the details: delving deep into convolutional nets." *arXiv preprint arXiv:1405.3531*.
- Chen, F. C., and Jahanshahi, M. R. (2018). "NB-CNN: deep learning-based crack detection using convolutional neural network and naive Bayes data fusion." *IEEE Transactions on Industrial Electronics*, 65(5), 4392-4400.
- Chen, K., Hu, H., Chen, C., Chen, L., and He, C. (2018). "An intelligent sewer defect detection method based on convolutional neural networks." *Proceedings of 2018 IEEE International Conference on Information and Automation (ICIA)*, IEEE.
- Cheng, J., and Wang, M. (2018). "Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques." *Automation in Construction*, 95, 155-171.
- Chughtai, F. and Zayed, T. (2008). "Infrastructure condition prediction models for sustainable sewer pipelines." *Journal of Performance of Constructed Facilities*, 22, 333-341.
- Dalal, N. and Triggs, B. (2005). "Histograms of oriented gradients for human detection." *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ, 886-893.

- Dang, L. M., Hassan, S. I., Im, S., Mehmood, I., and Moon, H. (2018). "Utilizing text recognition for the defects extraction in sewers CCTV inspection videos." *Computers and Industrial Engineering*, 99, 96-109.
- Dirksen, J., Clemens, F., Korving, H., Cherqui, F., Le Gauffre, P., Ertl, T., Plihal, H., Muller, K and Snaterse, C. (2013). "The consistency of visual sewer inspection data." *Structure and Infrastructure Engineering*, 9(3), 214-228.
- Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., and Brox, T. (2015). "FlowNet: learning optical flow with convolutional networks." *Proceedings of IEEE International Conference on Computer Vision*.
- Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). "The PASCAL Visual Object Classes (VOC) Challenge." *International Journal of Computer Vision*, 88(2), 303-338.
- Feeney, C. S., Thayer, S., Bonomo, M., and Martel, K. (2009). "White paper on condition assessment of wastewater collection systems." National Risk Management Research Laboratory, Office of Research and Development, US Environmental Protection Agency.
- Fuchs-Hanusch, D., Günther, M., Möderl, M., and Muschalla, D. (2015). "Cause and effect oriented sewer degradation evaluation to support scheduled inspection planning." *Water Science and Technology*, 72(7), 1176-1183.
- Girshick, R. (2015). "Fast R-CNN." *Proceedings of IEEE International Conference on Computer Vision*, IEEE, Piscataway, NJ, 1440–1448.
- Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation." *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ, 580–587.
- Greenspan, H., Van Ginneken, B., and Summers, R. M. (2016). "Guest editorial deep learning in medical imaging: overview and future promise of an exciting new technique." *IEEE Transactions on Medical Imaging*, 35(5), 1153-1159.
- Guo, W., Soibelman, L., and Garrett, J. H. (2009). "Visual pattern recognition supporting defect reporting and condition assessment of wastewater collection systems." *Journal of Computing in Civil Engineering*, 3, 160–169.
- Halfawy, M. R., and Hengmeechai, J. (2014a). "Automated defect detection in sewer closed circuit television images using histograms of oriented gradients and support vector machine." *Automation in Construction*, 38, 1-13.
- Halfawy, M. R., and Hengmeechai, J. (2014b). "Integrated vision-based system for automated defect detection in sewer closed circuit television inspection videos." *Journal of Computing in Civil Engineering*, 29(1).

- Halfawy, M. R., and Hengmeechai, J. (2014c). "Optical flow techniques for estimation of camera motion parameters in sewer closed circuit television inspection videos." *Automation in Construction*, 38, 39-45.
- Harvey, R., and McBean, E. (2015). "A data mining tool for planning sanitary sewer condition inspection." *Proceedings of Conflict Resolution in Water Resources and Environmental Management*, Springer, Cham.
- Harvey, R.R. and McBean E. A. (2014). "Predicting the structural condition of individual sanitary sewer pipes with random forests." *Canadian Journal of Civil Engineering*, 41, 294-303.
- Hassan, S., Dang, M., Mehmood, I., Im, S., Choi, C., Kang, J., Park, Y., and Moon, H. (2019). "Underground sewer pipe condition assessment based on convolutional neural networks." *Automation in Construction*, 106.
- Haurum, J. B., and Moeslund, T. B. (2020). "A Survey on image-based automation of CCTV and SSET sewer inspections." *Automation in Construction*, 111, 103061.
- Hawari, A., Alamin, M., Alkadour, F., Elmasry, M., and Zayed, T. (2018). "Automated defect detection tool for closed circuit television (CCTV) inspected sewer pipelines." *Automation in Construction*, 89, 99-109.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). "Deep residual learning for image recognition." *arXiv preprint arXiv:1506.01497*.
- Hernández, N., Caradot, N., Sonnenberg, H., Rouault, P., and Torres, A. (2018). "Optimizing SVM model as predicting model for sewer pipes in the two main cities in Colombia." *Proceedings of International Conference on Urban Drainage Modelling*, Springer, Cham.
- Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). "Mobilenets: efficient convolutional neural networks for mobile vision applications." <<https://arxiv.org/abs/1704.04861>> (July 15, 2018).
- Jahanshahi, M. R. (2011). "Vision-based studies for structural health monitoring and condition assessment." Ph.D. Dissertation, University of Southern California.
- Konig, A. (2005). "CARE-S WP2 external corrosion model description." SINTEF Technology and Society, Trondheim, Norway.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "ImageNet classification with deep convolutional neural networks." *Proceedings of Advances in Neural Information Processing Systems*, Neural Information Processing Systems Foundation, Stateline, NV, 1097–1105.
- Kumar (2018). "Automated anomaly detection in sewer cctv video." <<https://youtu.be/QKHI-EO9jM4>> (December 25, 2018).

- Kumar, S. S., Abraham, D. M., Jahanshahi, M. R., Iseley, T., and Starr, J. (2018). “Automated defect classification in sewer closed circuit television inspections using deep convolutional neural networks.” *Automation in Construction*, 91, 273-283.
- Kumar, S. S., Wang, M., Abraham, D. M., Jahanshahi, M. R., Iseley, T., and Cheng, J. C. (2020). “Deep learning–based automated detection of sewer defects in cctv videos.” *Journal of Computing in Civil Engineering*, 34(1).
- Laakso, T., Kokkonen, T., Mellin, I., and Vahala, R. (2018). “Sewer condition prediction and analysis of explanatory factors.” *Water*, 10(9), 1239.
- Le Gat, Y. (2008). “Modelling the deterioration process of drainage pipelines.” *Urban Water Journal*, 5(2), 97–106.10.1080/15730620801939398
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). “Deep learning.” *Nature*, 521, 436–444.
- Lee, D. H., Moon, H., and Choi, H. R. (2011). “Autonomous navigation of in-pipe working robot in unknown pipeline environment.” *Proceedings of 2011 IEEE International Conference on Robotics and Automation*, IEEE.
- Li, D., Cong, A., and Guo, S. (2019). “Sewer damage detection from imbalanced CCTV inspection data using deep convolutional neural networks with hierarchical classification.” *Automation in Construction*, 101, 199 – 208.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., and Berg, A. C. (2016). “SSD: Single shot multibox detector.” *Proceedings of European conference on computer vision*, 21-37.
- Lucas, B. D. and Kanade, T. (1981). “An iterative image registration technique with an application to stereo vision.” *Proceedings of 1981 DARPA Imaging Understanding Workshop*, pp. 121-130
<<http://www.cse.ucsd.edu/classes/sp02/cse252/lucaskanade81.pdf>> (September 20, 2019)
- Marlow, D., Boulaire, F., Beale, D., Grundy, C., and Moglia, M. (2011). “Sewer performance reporting: factors that influence blockages.” *Journal of Infrastructure Systems*, 42–51.
- Meeks, M. T. (2016). “Evaluating storm sewer pipe condition using autonomous drone technology (No. AFIT-ENV-MS-16-M-167)”. Air Force Institute of Technology Wright-Patterson AFB OH Wright-Patterson AFB United States.
- Meijer, D., Scholten, L., Clemens, F., and Knobbe, A. (2019). “A defect classification methodology for sewer image sets with convolutional neural networks.” *Automation in Construction*, 104, 281-298.
- Micevski, T., Kuczera, G., and Coombes, P. (2002). “Markov model for storm water pipe deterioration.” *Journal of Infrastructure Systems*, 8(2), 49-56.

- Mohammadi, M. M., Najafi, M., Tabesh, A., Riley, J., and Gruber, J. (2019). “Condition prediction of sanitary sewer pipes.” Proceedings of Pipelines 2019: Condition Assessment, Construction, and Rehabilitation, Reston, VA.
- Moradi, S., and Zayed, T. (2017). “Real-time defect detection in sewer closed circuit television inspection videos.” Proceedings of Pipelines 2017, Phoenix, Arizona, doi.org/10.1061/9780784480885.027.
- Moselhi, O., and Shehab-Eldeen, T. (2000). “Classification of defects in sewer pipes using neural networks.” Journal of Infrastructure Systems, 97–104.
- Nassiraei, A. A., Honda, M., and Ishii, K. (2010). “New approach to the self-localization of autonomous sewer inspection robots.” Proceedings of 2010 World Automation Congress, IEEE.
- North American Society of Sewer Service Companies (NASSCO) (2018). “Pipeline assessment and certification program (PACP®).” <<https://www.nassco.org/pipeline-assessment-and-certification-program>> (August 1, 2018).
- Radford, A., Metz, L., and Chintala, S. (2015). “Unsupervised representation learning with deep convolutional generative adversarial networks.” arXiv preprint arXiv:1511.06434.
- Redmon, J. and Farhadi, A. (2016). “YOLO9000: Better, Faster, Stronger.” <<https://arxiv.org/abs/1612.08242>> (January 15, 2018).
- Redmon, J. and Farhadi, A. (2018). “YOLOv3: An Incremental Improvement.” <<https://arxiv.org/abs/1804.02767>> (July 15, 2018).
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). “You only look once: Unified, real-time object detection.” Proceedings of IEEE conference on computer vision and pattern recognition, 779-788.
- Ren, S., He, K., Girshick, R., and Sun, J. (2015). “Faster R-CNN: Towards real-time object detection with region proposal networks.” Proceedings of Advances in Neural Information Processing Systems, Neural Information Processing Systems Foundation, Montréal, 91–99.
- Ridgers, D. (2007). “Weakness of current joint design and options for future development.” Proceedings of International Symposium of Trees and Underground Pipes, Institute for Underground Infrastructure, Gelsenkirchen, Germany, 66–72.
- Rizzo, C., Cavestany, P., Chataigner, F., Soler, M., Moreno, G., Serrano, D., Francisco, L., and Villarroel, J. L. (2017). “Wireless propagation characterization of underground sewers towards autonomous inspections with drones.” Proceedings of Iberian Robotics Conference. Springer, Cham.

- Rokstad, M. M., and Ugarelli, R. M. (2016). "Improving the benefits of sewer condition deterioration modelling through information content analysis." *Water Science Technology*, 74(10), 2270-2279.
- Schrock, B. J., ed. (1994). "Existing sewer evaluation and rehabilitation." *WEF Manual of Practice FD-6, ASCE Manuals and Reports on Engineering Practice No. 62*, 2nd Ed., ASCE, New York.
- Simonyan, K., and Zisserman, A. (2014). "Very deep convolutional networks for large-scale image recognition." <<https://arxiv.org/abs/1409.1556>> (July 15, 2018).
- Sinha, S. K., and Fieguth, P. W. (2006a). "Automated detection of cracks in buried concrete pipe images." *Automation in Construction*, 15(1), 58-72.
- Sinha, S. K., and Fieguth, P. W. (2006b). "Neuro-fuzzy network for the classification of buried pipe defects." *Automation in Construction*, 15(1), 73-83.
- Sinha, S. K., and Fieguth, P. W. (2006c). "Segmentation of buried concrete pipe images." *Automation in Construction*, 15(1), 47-57.
- Soukup, D. and Huber-Mork, R. (2014). "Convolutional neural networks for steel surface defect detection from photometric stereo images." *Proceedings of International Symposium on Visual Computing*, 668-677.
- Sousa, V., Matos, J. P., and Matias, N. (2014). "Evaluation of artificial intelligence tool performance and uncertainty for predicting sewer structural condition." *Automation in Construction*, 44, 84-91.
- Sousa, V., Matos, J. P., Matias, N., and Meireles, I. (2019). "Statistical comparison of the performance of data-based models for sewer condition modeling." *Structure and Infrastructure Engineering*, 15(12), 1680-1693.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). "Dropout: a simple way to prevent neural networks from overfitting." *The Journal of Machine Learning Research*, 15(1), 1929-1958.
- Stål, Ö. (2007). "SSLU—Research on pipe joints." *Proceedings of International Symposium of Trenches and Underground Pipes*, Institute for Underground Infrastructure, Gelsenkirchen, Germany, 60–65.
- Su, T. C., and Yang, M. D. (2014). "Application of morphological segmentation to leaking defect detection in sewer pipelines." *Sensors*, 14(5), 8686-8704.
- Tafari, A. N., and Selvakumar, A. (2002). "Wastewater collection system infrastructure research needs in the USA." *Urban Water Journal*, 4(1), 21-29.

- Tran, D. H., Perera, B. J. C., and Ng, A., (2007). “Neural network-based prediction models for structural deterioration of urban drainage pipes.” *Land, Water, and Environmental Management: Integrated Systems for Sustainability, Proceedings, Christchurch*, 2264-2270.
- Tzutalin. (2015). “LabelImg.” <<https://github.com/tzutalin/labelImg>> (January 15, 2018).
- United States Environmental Protection Agency. (2012). “Clean water sheds needs survey.” <<https://www.epa.gov/cwns>> (August 1, 2017).
- United States Environmental Protection Agency. (2014). “Quick Guide for Estimating Infiltration and Inflow.” <<https://www3.epa.gov/region1/sso/pdfs/QuickGuide4EstimatingInfiltrationInflow.pdf>> (July 15, 2018).
- United States Environmental Protection Agency. (2016). “Sanitary Sewer Overflows (SSOs).” <<https://www.epa.gov/npdes/sanitary-sewer-overflows-ssos>> (July 15, 2018).
- Vitorino, D., Coelho, S. T., Santos, P., Sheets, S., Jurkovic, B., and Amado, C. (2014). “A random forest algorithm applied to condition-based wastewater deterioration modeling and forecasting.” *Procedia Engineering*, 89, 401-410.
- Wells, T., and Melchers, R. E. (2014). “An observation-based model for corrosion of concrete sewers under aggressive conditions.” *Cement and Concrete Research*, 61, 1-10.
- Wilson, D., Filion, Y., and Moore, I. (2017). “State-of-the-art review of water pipe failure prediction models and applicability to large-diameter mains.” *Urban Water Journal*, 14(2), 173–184.
- Wirahadikusumah, R., Abraham, D., and Iseley, T. (2001). “Challenging issues in modeling deterioration of combined sewers.” *Journal of Infrastructure Systems*, 7(2), 77-84.
- Xie, Q., Li, D., Xu, J., Yu, Z., and Wang, J. (2019). “Automatic detection and classification of sewer defects via hierarchical deep learning.” *IEEE Transactions on Automation Science and Engineering*, 1 – 12.
- Xu, K., Luxmoore, A. R., and Davies, T. (1998). “Sewer pipe deformation assessment by image analysis of video surveys.” *Pattern Recognition*, 31(2), 169–180.
- Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). “Learning deep features for discriminative localization.” *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2921 – 2929.

APPENDIX A. ASCE REPRINT PERMISSION (JOURNAL OF COMPUTING IN CIVIL ENGINEERING)

4/10/2020 <https://marketplace.copyright.com/rs-ui-web/mp/license/56d90f4e-27b2-41a3-8a48-427a2c8bd48e/fa54a568-62cc-4705-8bb7-67e7f5807...>



American Society of Civil Engineers - License Terms and Conditions

This is a License Agreement between Srinath Shiv Kumar ("You") and American Society of Civil Engineers ("Publisher") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by American Society of Civil Engineers, and the CCC terms and conditions.

All payments must be made in full to CCC.

Order Date	06-Apr-2020	Type of Use	Republish in a thesis/dissertation
Order license ID	1025350-2	Publisher	American Society of Civil Engineers
ISSN	0887-3801	Portion	Page

LICENSED CONTENT

Publication Title	Journal of Computing in Civil Engineering	Country	United States of America
Author/Editor	American Society of Civil Engineers	Rightsholder	American Society of Civil Engineers
Date	01/01/1987	Publication Type	Journal
Language	English		

REQUEST DETAILS

Portion Type	Page	Rights Requested	Main product
Page range(s)	1-13	Distribution	Worldwide
Total number of pages	13	Translation	Original language of publication
Format (select all that apply)	Print, Electronic	Copies for the disabled?	Yes
Who will republish the content?	Academic institution	Minor editing privileges?	Yes
Duration of Use	Life of current edition	Incidental promotional use?	Yes
Lifetime Unit Quantity	Up to 999	Currency	USD

NEW WORK DETAILS

Title	Leveraging Big-Data and Deep Learning for Economical Assessment of Sewers	Institution name	Purdue University
Instructor name	Srinath Shiv Kumar	Expected presentation date	2020-04-03

ADDITIONAL DETAILS

Order reference number	N/A	The requesting person / organization to appear on the license	Srinath Shiv Kumar
------------------------	-----	---	--------------------

REUSE CONTENT DETAILS

Title, description or numeric reference of the portion(s)	Deep Learning–Based Automated Detection of Sewer Defects in CCTV Videos	Title of the article/chapter the portion is from	Deep Learning–Based Automated Detection of Sewer Defects in CCTV Videos
Editor of portion(s)	Srinath Shiv Kumar	Author of portion(s)	American Society of Civil Engineers
Volume of serial or monograph	Volume 34 Issue 1	Issue, if republishing an article from a serial	N/A
Page or page range of portion	1-13	Publication date of portion	2019-10-25

PUBLISHER SPECIAL TERMS AND CONDITIONS

Thesis reuse approved. Reuse of paper is permitted, provided it does not account for more than 25% of the new work.

APPENDIX B. ASCE REPRINT PERMISSION (INTERNATIONAL CONFERENCE ON COMPUTING IN CIVIL ENGINEERING)

RE: Permission to Reprint

PERMISSIONS <permissions@asce.org>

Fri 4/17/2020 1:02 PM

To: Srinath Shiv Kumar <kumar256@purdue.edu>

Dear Srinath,

Thank you for your inquiry. As an original author of an ASCE journal article or proceedings paper, you are permitted to reuse your own content (including figures and tables) for another ASCE or non-ASCE publication (including your dissertation), provided it does not account for more than 25% of the new work. This email serves as permission to reuse your work, A Deep Learning Based Automated Structural Defect Detection System for Sewer Pipelines from ASCE International Conference on Computing in Civil Engineering 2019.

A full credit line must be added to the material being reprinted. For reuse in non-ASCE publications, add the words "With permission from ASCE" to your source citation. For Intranet posting, add the following additional notice: "This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers. This material may be found at <https://doi.org/10.1061/9780784482445.029>."

Each license is unique, covering only the terms and conditions specified in it. Even if you have obtained a license for certain ASCE copyrighted content, you will need to obtain another license if you plan to reuse that content outside the terms of the existing license. For example: If you already have a license to reuse a figure in a journal, you still need a new license to use the same figure in a magazine. You need a separate license for each edition.

For more information on how an author may reuse their own material, please view:

<http://ascelibrary.org/page/informationforasceauthorsreusingyourownmaterial>

Sincerely,

Leslie Connelly
Manager, Publications Marketing
American Society of Civil Engineers
1801 Alexander Bell Drive
Reston, VA 20191

PERMISSIONS@asce.org

703-295-6169

Internet: www.asce.org/pubs | www.ascelibrary.org | <http://ascelibrary.org/page/rightsrequests>

APPENDIX C. ASCE REPRINT PERMISSION (CONSTRUCTION RESEARCH CONGRESS)

RE: Permission to Reprint Conference Paper

PERMISSIONS <permissions@asce.org>

Mon 4/20/2020 10:51 AM

To: Srinath Shiv Kumar <kumar256@purdue.edu>

Dear Srinath,

Thank you for your inquiry. As an original author of an ASCE journal article or proceedings paper, you are permitted to reuse your own content (including figures and tables) for another ASCE or non-ASCE publication (including your dissertation), provided it does not account for more than 25% of the new work. This email serves as permission to reuse the final draft of your work, *Leveraging Visualization Techniques to Develop Improved Deep Neural Network Architectures for Sewer Defect Identification*, being published in ASCE Construction Congress 2020 proceedings. "Final draft" means the version submitted to ASCE after peer review and prior to copyediting or other ASCE production activities

A full credit line must be added to the material being reprinted. Please note in your credit that you are using the **pre-production version of your paper**, with permission from ASCE, and include the DOI once the proceedings have been posted online. For reuse in non-ASCE publications, add the words "With permission from ASCE" to your source citation. For Intranet posting, add the following additional notice: "This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers. This material may be found at [URL/link of abstract in the ASCE Library or Civil Engineering Database]."

Each license is unique, covering only the terms and conditions specified in it. Even if you have obtained a license for certain ASCE copyrighted content, you will need to obtain another license if you plan to reuse that content outside the terms of the existing license. For example: If you already have a license to reuse a figure in a journal, you still need a new license to use the same figure in a magazine. You need a separate license for each edition.

For more information on how an author may reuse their own material, please view:

<http://ascelibrary.org/page/informationforasceauthorsreusingyourownmaterial>

Sincerely,

Leslie Connelly
Manager, Publications Marketing
American Society of Civil Engineers
1801 Alexander Bell Drive
Reston, VA 20191

[PERMISSIONS@asce.org](mailto:permissions@asce.org)

703-295-6169

Internet: www.asce.org/pubs | www.ascelibrary.org | <http://ascelibrary.org/page/rightsrequests>