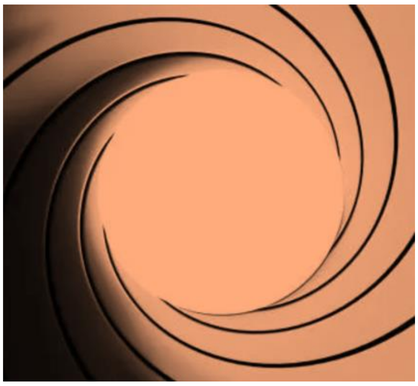
# Multi-Label Classification for Sewage Pipe Anomalies

Bachelor Thesis

Creative Technology, University of Twente

Matthijs Jaron Berkhout



Supervisors: Bram Ton

Critical Observer: Faizan Ahmed

Client: Rolsch Assetmanagement

Client: Ambient Intelligence, Saxion University of Applied Sciences and Rolsh Assetmanagement

Enschede, 2024

# Contents

# 1. Introduction

The Dutch sewage infrastructure encompasses approximately 150.000 kilometers of pipes [1], with an average lifespan of around 70 years, necessitating continuous maintenance to ensure public hygiene, sanitation, and environmental protection [2]. However, sewage maintenance is costly [1] with annual municipal expenses in the Netherlands reaching €1.8 billion in 2022 [2] largely due to expensive full pipe renovations. Municipalities aim to avoid increasing construction costs, as their funding relies on sewerage and water care charges collected from residents and often requires long-term loans. To reduce costs, RIONED recommends first inspecting the internal and functional conditions of the pipes [3] to identify risks and maintain structural integrity [2]. In particular, Subsequent appropriate maintenance actions should be undertaken [2] strategically planned alongside other infrastructural repairs [3]. Hence, accurate data collection and analysis of pipe conditions are essential for developing a long-term maintenance strategy [1]. This project is part of the Primavera initiative under the Dutch National Research Agenda, aiming to increase infrastructure reliability while reducing maintenance costs. Especially, predictive maintenance is the objective where analytical tools should predict potential issues to repair them in time, thereby maximizing the lifespan of existing resources. Therefore, novel optimization methods can be developed with collaborating partners to detect outliers and improve planning tools. Rolsch Assetmanagement, the client for this project, aims to enhance the consistency of its maintenance advice through improved sewage data analysis. Therefore, this thesis explores how integrating automated artificial intelligence (AI) systems could effectively capture pipe conditions. Consequently, the client's context and problem are considered providing a clear research direction on how potential technology improvements can contribute to the scope of automated sewage pipe inspection.

## 1.1. Specific Context

Rather than relying on human inspection, utilizing robots equipped with fish-eye cameras offers a practical and cost-effective inspection alternative, given their ability to frequently operate in unhygienic environments and provide interpretable data [2]. Utilizing Closed Circuit Television (CCTV) as a data acquisition on the robot enhances its intelligence by allowing for on-site manual control for the revision of detailed examination of potential anomalies. Eventually, these captured images from the cameras, mounted on both the front and rear of the robot, are provided to the sewage inspector through a web application developed by the client of this graduation project. This allows instructors to annotate both the frequency and severity of localized anomalies in specific frames based on the Standardized European Annotation Format. This annotation data supports the client's statistical models to assess the conditional evolution of certain sewage pipes. Ultimately, the findings could recommend what appropriate repair actions local sewage companies can pursue to minimize disruptions, increase productivity, and reduce costs [3].

## 1.2. Problem

Nevertheless, inspection data may be perceived as less trustworthy due to assessment inconsistencies between human inspectors. Robots supply inspectors with extensive CCTV data that could lead to time-consuming manual assessments caused by infrequent defect detection. This labor-intensive task may result in fatigue potentially impacting the accuracy and robustness of annotations. Besides, the subjective interpretation of the various types and severities of found defects and structural elements (anomalies) can rely on differing estimations, judgmental boundaries, and personal circumstances. Moreover, missing defects could be a consequence of an inspector's lack of experience and knowledge. Consequently, error-prone annotation data could impact the margin of error of statistical results to provide incorrect maintenance advice. Hence, a delay in appropriate maintenance decisions can lead to environmental casualties and an increased repairment budget. Additionally, since this is an anomaly classification problem, often its applied datasets are imbalanced which make these datasets not yet implementable into the training process to create a model which can generalize over underrepresented rare minority observations.

## 1.3. Current Solutions

State-of-the-art solutions can be evaluated compared to previous solutions which may have been constrained in their ability to effectively and accurately detect sewage-related features. Image processing techniques using handcrafted morphological features proved to be inefficient in complex and dynamic environments due to illuminated or noisy conditions. Integrating these techniques with machine learning methods can establish a diagnostic framework aimed at enhancing the consistency and performance of sewage defect classification and detection. Especially, deep learning techniques can automate accurate condition assessment by assisting the inspector during swage pipe inspections. To illustrate Xianfei [1] customized a YOLOv3 Residual Network based on pre-processed videos to supply specifically selected videos with detected and classified defects. In addition, Johannes [2] trains a Fully Resolution Residual Network using unwrapped, pose-estimated, and illuminated fish-eye images to improve the reliability and assistance in segmenting localized and classified defects. Besides, Srinath [3] employs a Fully Residual Convolutional Neural Network (CNN) to increase accurate defect classification rather than localization. Syed [4] fine-tunes an AlexNet CNN model using feature-labeled and augmented data to provide inspectors with notifications regarding classified defects. The evaluation of the integration of these state-of-the-art model solutions could be relevant to the argument of the technical objectives of this project.

## 1.4. Literature Recommendations

Moreover, addressing the gaps in the literature suggests that future research should focus on training multiple combined models with increased data diversity to output informative labeled images. Enhancing data preparation through processing methods should increase the data variety of exposed conditions by integrating diverse databases and performing augmentation to compensate for inconsistencies and human biases. Furthermore, improving misclassification and defect distinction can involve utilizing temporal depth data from capturing sewage pipe shapes with 3D stereovision and wrap-around techniques. Regarding the design of an image recognition model, multiple pre-trained image detection models could be merged where the design choices of localization over classification could have higher attention. The performance should be benchmarked on generalized datasets to decrease the amounts of false positives through the iteration of improved model parameters. Particularly speed over accuracy is prioritized for real-time remote inspection systems. Besides, informative labels should specify the type and severity of conditions, which should be automatically translatable into CSV files. Additionally, providing a smaller selection of video or frames containing potential anomalies could reduce wasted inspection time. Therefore, integrating this recommendation is crucial to developing a robust sewage defect detection framework, especially when dealing with inexperienced or poorly trained inspectors.

## 1.5. Objective

This thesis aims to train a neural network to automate sewage pipe defect inspection by facilitating inspector assistance in the localization and classification of defects and structural elements. A pre-trained convolutional neural network, such as a single residual neural network (ResNet) or a combination, can undergo transfer learning by training on provided fish-eye images to adapt pre-trained features to specific sewage data. The dataset's size and diversity can be scaled by utilizing multiple datasets or applying augmentation techniques to expose the model to a wider range of sewage conditions. Image processing techniques can be applied to change the representation of distorted fish-eye images to increase feature distinction. Training the model with extracted sewage pipe shape properties can enhance its spatial awareness. Furthermore, metric results should prioritize accuracy over speed and minimize false negatives. Therefore, the F2 score can be utilized to give greater weight to recall over precision, aiming for a recall value approaching 1 while minimizing false positives as much as possible. Ultimately, web application inspectors are assisted in a web application with highlighted points containing frames of localized defects or structural elements. The frames are supplemented with a CSV file containing annotations according to the European Standards EN 13508-2:2003+A1 coding system. This thesis solution should address the issue of inconsistency to enhance the reliability and accuracy of future automated sewage inspections.

## 1.6. Research Questions

How to automate sewage pipe annotation with supervised anomaly classification by training a neural network that is robust to inconsistent labelling and data imbalance?

1. What pre-trained image classification neural networks can be fine-tuned to increase the recall value while maintaining an optimized F2 score?
2. How should the imbalanced data and images be pre-processed to increase the recall and f2 score of the model's performance?
3. To what extent does the trained model increase the objectivity of classifying anomalies compared to manual inspection?

## 1.7. Approach

Chapter 1 describes the research questions arising from contextual problem analysis combined with a proposed solution. Chapter 2 discusses the utilized methodology and this approach in further detail. In Chapter 3, which could be considered as the business understanding, consists of conducting interviews and a literature review to find potential data preparation techniques and state-of-the-art pre-trained neural network models that could comply for he modeling part of this thesis. Chapter 4 concatenates, explores and prepares the data by following a filtering chain. Chapter 5 describes what experiments have been conducted to assess the potential methods found to comply with the first two sub-research questions and interprets the extreme results throughout the experiments. Chapter 6 discusses the results related to the sub-questions and discusses the framework’s limitations and future recommendations. Finally, Chapter 8 concludes by answering the main research question and shows how certain methods used in this research could supplement the related research.

# 2. Methodology

## 2.1. CRISP-DM

This thesis adopts a methodology based on the CRISP-DM (Cross-Industry Standard Process for Data Mining) cycle. This flexible framework facilitates iterative model development by comparing training results based on prioritized metrics. The CRISP-DM methodology is tailored to prioritize short-term model deployment validation over long-term monitoring considerations. Instead, emphasis is placed on short-term deployment within the company's operational workflow, allowing for experimental validation and iterative improvements to the framework.

### 2.1.1. Business Understanding

The reasons behind initiating the development of a new deep learning framework depend on the investigated objectives from the company’s contextual circumstances. This involves conducting interviews with employers who encounter the relevant challenges to gain their expectations and supplement the requirements for the project goals. Additionally, the company’s resources for neural network employment and potential integration limitations can be investigated. Furthermore, a literature study could supplement the thesis objectives by analyzing the problem context and limitations of previous research. The results from the business understanding are mainly incorporated into the introduction chapter since the main priority of this thesis focuses on the next three chapters. Throughout the business understanding is considered that instead of using the term anomalies, the term ‘observation’ is used instead from the standardization protocol for inspection, which can consist of multiple types of anomalies.

### 2.1.2. Data Understanding

The provided sewage pipe inspection dataset consists of annotated images that can be analyzed to interpret the structure of the data. For instance, image quality validation, annotation completeness, the surface characteristics of the data, and the intercorrelated property keys can be explored. Additionally, organizing the data into a diagram can provide a comprehensive overview of relationships to aid the querying of images and their annotations more efficiently.

### 2.1.3. Data Preparation

The data must undergo careful selection and preparation to align the representation of the input data with the mandatory model input. Different textural annotation data files can be combined to connect the related data. Nevertheless, careful data selection based on inclusion and exclusion criteria should increase data cleanness by mitigating error-prone information. Besides, fish-eye image data should be processed into a compatible format for the neural network input by utilizing different image processing techniques. Hereby, algorithms capable of interpreting spatial information from sewage pipe images be considered for improved prediction accuracy. Eventually, training data can be increased by either combining datasets or augmenting the current data.

### 2.1.4. Modeling and Evaluation

Various pre-trained image detection models can be with transfer learning to compare the different metrical results. Iterative experimentation can be done by splitting the data into training, testing, and validation sets to mitigate overfitting. This process should guide the selection of a particular architecture, followed by iterative optimization of the hyper-parameters to enhance specific metric performance. The model's output findings can be compared against the company's stated criteria and requirements. Therefore, reviewing the adequacy of the model and identifying any framework deficiencies alongside user feedback. Subsequently, decisions can be made regarding framework deployment or further iteration based on improved business criteria.

# 3. Literature Research

The following literature reviews are written to get an understanding of the various models that could be implemented, the training balancing methods and a collection of metrics which could represent the results such that could be interpreted if the model verifies to the aim of this thesis. Particularly, the first literature review is written during the academic writing course in quartile 3 of the 2023-2024 academic year.

## 3.1. Deep Learning Methods

Inspecting the conditions of sewage pipes is crucial for identifying risks to undertake appropriate actions for the maintenance of a functional sewage system and ensure public hygiene [1]. Rather than relying on human inspection, utilizing robots equipped with fish-eye cameras offers a more practical solution, given their ability to operate in unhygienic environments and provide interpretable data [2]. These images are provided to the sewage inspector through a web application developed by the client of this graduation project, which enables instructors to annotate specific frames. This annotation data supports the client's statistical models to provide maintenance recommendations for local sewage companies to minimize disruptions, increase productivity, and reduce costs for repairs [3].

Nevertheless, their statistical analysis is affected by inconsistent annotation data due to variations in the interpretation of defect severity and localization among instructors, despite utilizing standardized annotation notations. This can be caused by the fatigue of inspectors from the time-consuming assessment of big data or a lack of expertise [2].

For this graduation project, sewage defect detection and classification should be automated with a neural network that assists the inspector with objective annotation suggestions. Hereby, accuracy and recall metrics are essential for increasing the number of correctly predicted defects. Thus, various neural networks need to be utilized for this target performance. Therefore, the goal of this review is to gain insight into which neural network prioritizes accuracy and recall performance in the detection of defects in sewage pipes. First, the review explores how convolutional neural networks are the backbone for improved deep learning methods. Secondly, the review describes which neural network architectures are applicable for the multi-label classification of objects in images. Secondly, a comparison between state-of-the-art neural networks for sewage defect detection is conducted based on disparities in performance metrics of accuracy and recall.

### 3.1.1. Convolutional Neural Network as the Object Detection Backbone

Various sources indicate the widespread use of deep learning methods that learn abstract features and patterns from images that are used as generic feature descriptions to detect objects in new images [1, 2, 3, 4, 5, 6] whereas convolution-based neural networks are commonly used as robust deep learning methods for object detection. Convolutional Neural Networks (CNNs) are essential to advance deep learning object detection methods serving as the backbone from which various architectures can be improved and adapted. The fundamental operation of CNNs involves classifying images or regions of interest (ROIs) into binary categories such as ‘yes’ or ‘no’ [2]. CNNs generate feature maps serving as edge detectors [4] to accurately classify image features across diverse conditions due to their capability to extract multi-dimensional features [5]. Sun et al. [5] elaborate on how CNNs classify new data by extracting low-level features which are subsequently synthesized into higher-level complex features to facilitate the recognition of larger objects. Despite the advantage of requiring less training data, Haurum and Moeslund [2] note that applying a single CNN for each defect detection in an image demands computational efforts and suggests exploring different neural network architectures as backbone networks. Furthermore, constraints in the size of training data and the training time impact CNN accuracy. Therefore, Sun et al. [5] and Hassan et al. [4] advocate for transfer learning to refine a model's parameters for specific problem cases by re-training pre-trained models with case-specific data to preserve their generic features learned from extensive datasets. Additionally, Haurum and Moeslund [2] report from their survey that, besides binary classification, multi-class and multi-label classification architectures are used in the literature.

### 3.1.2. Multi-Label Classification in CNNs

Three modified convolutional neural network architectures are frequently mentioned in the literature which compromise detection speed and accuracy for detecting defects in sewage pipe images that include detecting and classifying found defect regions.

First, One-stage object detection methods achieve high processing speeds but often compromise on extracting detailed low-level spatial information. Sun et al. [5] outline the Single Shot Detector (SSD) method that uses a feedforward approach to directly localize and filter out overlapping anchor boxes. Furthermore, Yin et al. [7] describe how the You Only Look Once (YOLO) method uses pre-determined grid cells instead of ROI to find possible defect areas more quickly. However, limitations regarding the lack of spatial information it can interpret. Zhang et al. [3] point out that the extended sequence of top-down layers hinders the interpretation of low-level features, often resulting in the oversight of smaller defects.

Secondly, to overcome the challenge of missing scaled-down features, CNNs can be improved into two-staged regional-based networks to enhance speed and accuracy in object detection. First, the literature repeatedly describes the utilization of Region-based Convolutional Neural Networks (R-CNN). The framework consists of a Selective Search component to generate ROIs with bounding boxes from Bounding Box Regression that potentially contain defects. Subsequently, a CNN classifies a defect for each proposed region with a Single Vector Machine [6, 7, 1, 5]. Nevertheless, Kumar et al. [1] and Cheng and Wang [6] note from their literature study the demand for computational resources since a single CNN per ROI is used. Therefore, the R-CNN framework is modified into a Fast R-CNN, which utilizes a single CNN to identify image features that the Selective Search then uses to determine each ROI [1, 5, 6]. To try to increase the speed and accuracy a trained CNN called the Region Proposal Network to detect the ROI instead of the Selective Search approach introduces Faster R-CNN [5, 6]. Cheng et al. [6] argue that by sharing convolutional layers between Fast R-CNN and RPN, the computational effort is reduced and the quality of the suggested regions is increased. Nevertheless, this should not need to mean Faster R-CNN increases in speed since Zhang et al. [3] argue that Faster R-CNN is slower but rather obtains a higher accuracy.

Lastly, a modified architecture introduces a dual pipeline hierarchical structure to enhance multi-classification across various CNN-based systems, moving beyond the previous two-stage localized detection methods. These architectures are based on Residual Neural Networks which are grounded on a CNN where Yin et al. [7] state that multiple applied Residual Blocks could increase the detection of various scaled features. This method can be further explained by the development of Künzel et al. [8] to develop a semantic defect classification method by adjusting a Full-Resolution Residual Network (FRRN) to reduce computational resources and time. This hierarchical method employs two data pipelines that extract various-sized feature maps for detecting robust features to merge into a full-resolution residual stream that performs segmentation by extracting detailed features. Therefore, the RNN method seems to have the potential for multiple applications for object detection. From a background review of multiple usable networks, Haurum et al. [2] found that R-CNN frameworks can consist of Residual Neural Networks (RNN). In addition, Kuman et al. [1] evaluated the performance of a Faster R-CNN which contained a pre-rained ResNet.

This concludes that while one-stage CNN architectures offer detection speed, two-stage models provide accuracy, and hierarchical residual networks can be used as the backbone for interpreting multi-scale features in regional-based networks.

### 3.1.3. Accuracy Comparison of Employed Defect Detection Models

Experimental studies improved the classical, one- and two-stage convolutional-based neural networks to outperform both traditional and researcher-developed models, to achieve better accuracy and recall in defect detection.

First, improved and fine-tuned classic convolutional neural network-based models tend to be affected by the quality of the training data. Meijer et al. [9] developed a CNN with 16 layers with a VGG-16 model to perform multi-label classification to find different defects per image which was argued to have outperformed a reference state-of-the-art CNN by detecting around 90% of the defects. Besides, Hassan et al. [4] fine-tuned the pre-trained ImageNet AlexNet on case-specific training data which got an average accuracy of 96.60%. The difference between the performances can be concluded by the fact that Hassan et al. [4] augmented the data with blur which increased the accuracy ranging between 15% to 32%, this could indicate that applying transfer learning or augmenting the training data of Meijer et al. [9] could increase the accuracy significantly.

Secondly, a comparison indicates that replacing architectural components with multi-scale feature detection elements can improve the accuracy of one-stage object detection models. First, Kumar et al. [1] evaluated the performance of an SSD with a VGG-16 just like Meijer et al. [9] , however, the SSD reached an mAP of 54.4% and neglected defects, which might be due to the lower training set size. This performance was also compared to the YOLOv3 which was slower and missed smaller-sized objects but reached an mAP of 74.5% [1]. Furthermore, Li et al. [10] found that YOLOv3 reached an mAP of 68.1% for defect localization and classification. However, Yin et al. [7] challenged this outcome by arguing that the multi-scale feature extraction capabilities of the YOLOv3 are still proficient and modified the method by inserting residual blocks for an accuracy increase, which reached an mAP of 85.37%. Moreover, Zhang et al. [3] show that the YOLO model can be improved as well by constructing a YOLOv4-D-SPP3 by choosing the optimal loss function and fusing local and global scaled features. This is done with a Feature Pyramid Network (FPN) to improve accuracy for detecting minor defects the YOLOv4-D-SPP3 model reaches an mAP of 92.3%, however, it jeopardizes processing speed.

Lastly, experiments with two-stage object detection methods enhanced Faster R-CNN architectures by improving region proposal techniques and integrating pre-trained models to increase accuracy. Li et al. [10] experimented with different region proposal techniques such as Selective Search, RPN, and Stregntened Region-based Proposal Network (SRPN) together with a VGG-16-based Faster-RCNN, where Li et al. [10] claims that the SRPN together with a Faster-RCNN outperformed other methods with an mAP of 72.5% for defect classification. Just as component improvement of the YOLO network, Kumar et al. [1] proposed a Faster R-CNN which performed classification based on a VGG-12 or pre-trained ResNet and outperformed the YOLOv3 and SDD network by obtaining an average accuracy of 76.2% mAP. Furthermore, Cheng et al. [6] improved the classification CNN in the R-CNN based on a modified pre-trained Zeiler-Fergus with additional layers which resulted in an mAP of 83%. From this study, Cheng et al. [6] report from their experiments that when training with a larger dataset increases accuracy. These results could suggest that a pre-trained backbone model consisting of Residual Blocks can increase the accuracy. Therefore, the survey of Haurum and Moeslund [2] found that various studies ground R-CNN networks with ResNets with different layers pre-trained on ImageNet to extract robust and detailed features. Künzel et al. underscore this statement since the performance of their Full-Resolution Neural Network reaches an mAP of 84.5%, however, they find regions based on semantic pixel collections rather than region estimations.

In conclusion, substituting regional proposal techniques with regional proposal networks should input their regions of interest through defect classification backbone models pre-trained on ImageNet to improve the accuracy of defect detection.

### 3.1.4. Conclusion

This literature review aimed to identify which neural network can perform defect detection in sewage pipes by prioritizing accuracy and recall. Object detection methods generally consist of frameworks consisting of different sequentially ordered modules that rely on each other to perform object localization and classification. Hereby, most elements use convolutional neural networks (CNNs) as a backbone which might be either optimized in their architecture or could be pre-trained on existing large trained models. Particularly, implementing pre-trained CNNs on a benchmarked dataset or CNN-based Residual Neural Networks into a two-stage hierarchy to form a Region-based network seems to be key to extracting multi-dimensional features to increase the model's accuracy for regional localizations and defect classification.

This literature review is relevant for informing decisions on which current state-of-the-art models for defect detection could be adapted through iterative experiments to optimize either accuracy or speed. By providing evidence of the metric performances of already existing models can be considered what types of neural networks, modules, or structures can be integrated into existing architectures to improve defect detection related to the type of input data and preferred output data. Nevertheless, this review primarily on general descriptions and quantitative performance metrics of commonly used CNN-based architectures by excluding relevant optimization factors. For instance, discussions regarding non-CNN-based object detection architectures, data quality effects such as data augmentation, class imbalances, and the impact of varying annotation standards are not mentioned. Additionally, comparisons of state-of-the-art models were based solely on the mean Average Precision (mAP) due to the lack of overlapping standardized metrics.

To guide future sewage defect detection research, studies should examine how reordering architectural components and modifying layer structures can enhance performance metrics of interest, especially when focusing on the comparison of the metric performances of specific defects. Practically, iterative experiments can be performed to evaluate the effects of architectural adjustments on model performance. Therefore, enhancing data quality could be crucial, as it significantly impacts the outcomes of existing models identified in the literature. Additionally, training single models on various datasets with sewage images will help identify which networks consistently deliver high accuracy across different conditions.

## 3.2. Imbalanced Datasets

Imbalanced data distributions commonly occur in sewage inspection datasets due to the significant amount of normal images and fluctuating anomaly class quantities. The scarce frequency and varying diversity of occurring anomalies originate from contextual factors [12]. Consequently, uneven data class distributions cause a classifier's accuracy performance to decrease significantly [13] which is demonstrated by Dang et al. [13] from significantly decreased AUC values by absent balancing methods compared to their presence. This imbalanced data problem (IDP) challenges the performance of classifiers due to their dependency on training data quality [12]. Classifiers could bias and overfit towards the majority class, although likely underperforming the minority class [14] since less distinguishable features between minority classes can be learned [14]. Especially for multi-class classification, the model must choose one anomaly between underrepresented classes [14]. Additionally, IDP affects the model's learning process and the generalizability of minority classes in the test set [14] [5]. Nevertheless, due to the originality of each dataset for each project, distinct data balancing methods could be applied per individual case. Therefore, this review explores which data balancing methods to improve the imbalanced data problem to increase the robustness of multi-label classification models. This is initiated by first assessing which state-of-the-art dataset balances are applied to increase the robustness of the model. Secondly, mitigation methods are explored to adjust the data distribution during data preparation. Lastly, modification strategies to the architecture or training process of the classifier are considered.

### 3.2.1. Data Distributions

Four data distribution balances are mentioned in related research to confirm the IDP which considers distribution differences between the normal, total defect, and individual classes. In particular, two methods are commonly used and recommended for … First, a uniform distribution between the defect class and the normal images is generally mentioned [11]. The hierarchical classification network of Li et al. [14] and Xie et al. [12] contains a balanced distribution between the total defect and normal images in the higher-level classification. Important to note, is that the quantities between defect classes in the total defect class of Xie et al. [12] stays naturally imbalanced. Secondly, the distribution equalization of the individual defect classes within the total defect class [11]. Xie et al. [12] includes this balance for multi-defect classification where only the most commonly and similar occurring defects were included. The smaller-sized excluded classes were combined into another class, and not used during the fine-tuning of the multi-class CNN.

Secondly, two uncommoner data balances are found in the literature which may have proficiencies outside the training set and for datasets with extremely rare anomalies. Thirdly, dataset imbalance can be maintained to replicate the initial real-life distribution [2]. Particularly, the literature makes a distinction between preferably balancing the training set while maintaining the imbalance of the test and validation set [9] to ensure a realistic testing distribution as one would encounter anomalies in sewage inspections. Lastly, a rare data distribution is where the distribution is equalized across all separate classes, including the normal class [2], which may lead to overfitting of the oversampled rare anomalies and losing valuable information about the normal class due to relative sample reduction. In conclusion, Haurum and Moeslund [2] recommend a distribution consisting of a balanced number of normal and anomaly images where each anomaly class consists of an appropriate sample amount [2]. Rare classes with fewer samples can be excluded to neglect a long-lasting distribution [2]. Therefore, overfitting by skewed datasets toward the majority class can be mitigated.

### 3.2.2. Resampling

Five commonly used re-sampling methods found in related research balance data distributions by changing the frequency of occurring sample samples in the minority and minority classes. He et al. [16] found by performing a literature review that most classifier accuracy increases from applying sampling methods alleviating imbalanced data.

First, random oversampling duplicates class samples to often increase minority data class quantities to preferred levels [16]. The method is promisingly utilized by Li et al. [14] with different oversampling multiplications per class to increase the size of each minority class toward the size of the largest defect class. Meanwhile, Meijer et al. [9] performed oversampling on defective images in the training set by a multiplication of five to ensure the chance of each batch containing at least one defect. However, the main disadvantage of oversampling minority classes is the possibility of model overfitting [15] on specific image feature rules from the chance of observing identical images, causing the test sets to perform poorly [16]. Therefore, adaptive synthetic sampling methods based on SMOTE (Synthetic Minority Over-sampling Technique) are effective in dealing with the IDP [16] by synthetically generating samples with image interpolation considering the k-nearest sample neighbors and preventing over-generalization from preventing overlapping images of nearing neighboring samples. Borderline-SMOTE improves the classifier's distinguishing between classes and ADASYN (Adaptive Synthetic Sampling) focuses on increasing the classifier's capacity to classify hard-to-classify samples [16].

Secondly, random undersampling can be performed by randomly neglecting samples [14], often from the majority class [16], with for instance the RUS method [15]. Nevertheless, undersampling of the normal class could decrease the model's accuracy [14] since relevant image features from the majority class can be neglected [16]. Therefore, informed undersampling methods, such as EasyEnsemble, BalanceCascade, and K-nearest neighbor algorithms [16], can be utilized to prevent the loss of relevant data information [15]. Additionally, previously mentioned oversampling algorithms could increase the amount of overlapping images and classes [16]. Thus, data-cleaning methods could improve classification by supporting distinct class clusters by image reductions to preserve samples with the nearest common class instances, by for example Tomek links [16] [15]. Nonetheless, data cluster arrangement could rely on both up- and downsampling by cluster-based sampling methods to ensure robust representations of rare anomalies [16]. In addition, this approach is flexible improving rather within or between class balances, and allows for oversampling technique integrations [16].

### 3.2.3. Cost-Sensitive Learning

Rather than adjusting imbalanced data levels, cost-sensitive learning implementations adjust certain classes' relevancy by assigning misclassification costs per class or sample. This could be a promising technique since He et al. [16] found that cost-sensitive learning likely outweighs resampling methods. In general, cost-sensitive strategies handling data imbalance in neural networks can be distinguished into modifying the loss function or hyperparameters [16].

First, cost-sensitive modifications can be applied to either the classifier's initial output or calculated probability values after forward propagation [16]. The latter is mainly utilized in current sewer anomaly classification studies where the loss function is multiplied by district inversive weights per class, based on the occurrence frequency of the specific class in the dataset [11]. This approach allowed Meijer et al. [9] to prioritize false negatives against false positives by implementing relative weights in the cross entropy loss and granted Haurum and Moeslund [11] to improve the model's learning of minority classes by lowering their loss contribution. Assigning heavier weights to minority class misclassifications helps prevent bias towards majority classes, thereby avoiding a reduction in classifier performance [13] [9], and can also enhance model learning [11]. While this strategy improves anomaly classification, insignificant model improvements and model performance variations of this technique can be compensated by combining this method with ensemble techniques by introducing cross-validation [16].

Furthermore, aside from the previously mentioned weighted loss functions, loss functions can be completely replaced by other loss functions. As such, a cost minimization function minimizes the expected cost of misclassifications instead of reducing the number of incorrect predictions by including weights in the loss function [16]. Additionally, during pixel segmentation, Kunzel et al. [8] utilize another form of cost-sensitive learning by utilizing a bootstrapped cross-entropy loss function with a threshold incorporated to focus on misclassified or low-classified confidence pixels to increase the model performance on these harder examples. Another proficiency of the BCEloss is the mitigation of misclassified or noisy labeled images in the dataset to increase the model's robustness and generalization. Secondly, distinct learning rates can be defined depending on separate class costs to increase the impact of gradients for prioritized classes. This method greatly improves the classifier and reduces the training time for low-cost examples [16]. Lastly, it is important to note that case-specific purposes determine which classes are weighted heavier than others. Therefore, cost-sensitive learning can also be used to not tackle imbalanced datasets, as Haurum and Moeslund [11] applies class-importance weights (CIWs) prioritizing classes with higher economic consequences.

### 3.2.4. Two-Level Hierarchical CNN

Lastly, model architectures can be adjusted in five ways based on sequentially utilizing the different level features. Two-level hierarchical CNNs can be integrated to significantly improve the model's performance by dealing with an imbalanced dataset [12]. Li et al. [14] integrate a two-step hierarchical softmax classification model for easier multi-classification, by performing binary classification for normal and defect images, followed by multi-classification of various defects [14] by dividing a level convolutional layer into binary classification and multi-class classification to detect a specific anomaly in the image earlier detected as an anomaly for easier distinguishing between different classes. Similarly, Xie et al. [12] deploy a two-step hierarchical network, although a different implementation method strategy is deployed than Li et al. First, a CNN is trained for binary normal and defect classification, then duplicating and fine-tuning the previous CNN but performing multi-class classification.

Regarding the previously mentioned methods, although the hierarchical method increases scene robustness and classifier accuracy, defects with less distinguishable features go unnoticed [18]. This definition counts for Li et al. [14] as differentiating between defects during multi-class classification is difficult, while Xie et al. [12] mention that their model is limited in detecting smaller defects in the binary normal-defect classification. Therefore, the type of two-step hierarchical implementation type can be weighted while considering the improvement of classifying rare defects [12]. Both studies consider image quality, resolution reduction, defect co-occurrence, and indecent labeling impacting model estimations [12] [14].

Compared to Li et al, utilizing a distinct CNN for inter-defect classification could enhance model performance by allowing optimization of the second networks specifically for defect classification. Additionally, other classifier-level methods can be considered, for instance, a CNN fusion structure including an inceptionV3 for binary ND classification combined with a ResNet which raises model accuracy [19].

## 3.3. Quantitative Evaluation Metrics

Quantitative evaluation metrics are crucial in sewage anomaly classification to evaluate model performance and guide ongoing adjustments. These metrics are essential for comparing state-of-the-art models to put the trustworthiness of the developed model in perspective. Especially, a collection of metrics can supplement the model's quality assessment [9]. The choice of metrics should align with the project's objectives [9]. Regarding, multi-class and multi-label it is necessary to select metrics that not only assess overall performance but also distinguish performance per anomaly class [16]. This distinction is important for identifying classes that may require additional focus during data preparation or training, especially when dealing with smaller sample sizes and high-dimensional datasets to assure the model's generalizability.

However, the selection of certain metrics is not yet standardized for sewage anomaly classification [2] [5]. On top of that, the mentioned metrics in related research might not comply with the focus of this project to measure model consistency. From the literature, anomaly classification applications often prioritize the recall metric, aiming to find real occurring anomalies while minimizing the model to miss them [11]. Nevertheless, solely relying on recall may be in-comprehensive. Hence, incorporating additional metrics should be considered to form an informative perspective on the model performance. Sequentially, for this project, the question arises of which quantitative valuation metrics can measure a multi-label classification model's consistency while training models on imbalanced datasets. This specific review first focuses on providing pre-knowledge about the base metrics for other metrics. Secondly, the advantages and disadvantages of primary and auxiliary metrics are argued. Whereafter, independent auxiliary metrics are considered overarching multiple-adjusted classifiers. In closing with conditional metrics.

### 3.3.1. Basic Confusion Matrix Elements

The varying formula constructions of relevant metrics are mainly based on single and adjusted confusion matrix matrices. Primary and supplementary metrics are calculated based on the basic confusion matrix elements as seen in Fig […]. This matrix includes TP and TN which define the correctly predicted occurring and non-occurring anomalies, whereas FP and FN describe the incorrectly predicted occurring and non-occurring anomalies. Consequently, these metrics can be measured in rates to determine their occurrence chance. Two of these metrics in rates are auxiliary metrics. First, the TPR, or sensitivity/recall, determines the proportion of actual positives correctly identified by the model. Its inverse, the FNR, measures the proportion of real positives incorrectly predicted as negatives, representing the miss rate. Secondly, the TNR, or specificity, measures the proportion of actual negatives correctly identified by the model, while its inverse, the FPR, measures the number of incorrectly predicted anomalies [9].

### 3.3.2. Primary Metrics

From the literature are five commonly used metrics argued for their effectiveness for utilizing for image classification. Starting with accuracy, recall, and precision as auxiliary metrics. Since these insights likely not fully capture the model's performance when used in isolation, primary metrics are shown:

The commonly used accuracy can be misleading for model evaluation performance due to its sensitivity to unbalanced datasets. The primary metric is frequently utilized to compare different state-of-the-art datasets in the literature to measure the proportion of correctly predicted samples among all samples [9]. For balanced datasets, accuracy can be a good metric [2]. Nevertheless, Meijer et al. [1] argue that accuracy is misleading for imbalanced datasets with sparse anomalies, as it increases performance by focusing on the abundance of correctly predicted non-anomalous images. He and Garcia [16] strengthen this by explaining that in a dataset with 95% normal images, an accuracy of 95% can be achieved trivially. Additionally, accuracy is sensitive to data deviations [16], such as changes in dataset complexity and attributes [14]. This sensitivity arises because metrics incorporating both columns of the confusion matrix (see Fig […]) are influenced by the sizes of both the majority and minority classes, with the majority class often being negative and the minority positive [16]. Consequently, the metric is less informative for relative analysis during model benchmarking, as compared models might be trained on differently skewed class distributions [14] [19]. Therefore, supplementary metrics that do not rely on both columns of the confusion matrix, such as recall, precision, F1-score, and class confusion matrix, are recommended for use with imbalanced datasets [2].

### 3.3.3. Auxiliary Binary Metrics

Three auxiliary metrics are widely found in literature as comprehensive metrics while training a model with imbalanced datasets. Of the possible auxiliary metrics, Haurum and Moeslund [2] recommends recall, f1-scores, and average precision for image classification metrics. Li et al. [14] notes that these metrics mainly depend on class distributions, mitigating the imbalance between defects and normal images. First, precision measures the number of true positives overall predicted occurring anomalies to determine the exactness of correctly labeled anomalies [19]. Nevertheless, precision is distribution-sensitive since both columns of the confusion matrix are used during the metric calculation [16]. Owing to this, recall measures the completeness of predicted anomalies by focusing on capturing the percentage of undetected anomalies by rationing true positives over the ground-truth positive instances [19]. Nevertheless, it does not penalize false positives which could increase the inspection range for a second revision by inspectors.

To comprehend both metrics' limitations, the F$\beta$-score provides a weighted trade-off between recall and precision to gain insight into the model functionality. Specifically, literature often defines the metric as the "harmonic mean" ($\beta$ =1) being less sensitive to data distribution changes. Additionally, the F-score improves over the accuracy [16] since it can even validate accuracy [19]. Furthermore, Haurum and Moeslund [11] point out utilizing tailorable metrics reflecting the real-world importance of certain auxiliary metrics for sewer inspections, for instance, the F$\beta$-metric for linearly weighting recall with the $\beta$ value. Eventually, the maximum output for the F-score can determine the optimal threshold to convert classifier probabilities into binary predictions. Moreover, regarding the project's aim to focus on recall, the $\beta$ value can be set higher than 1, often to 2, to prioritize higher recall values. Haurum et al. [11] showed the results per class with the F2CIW metric where the classes had weight-importance, and the normal images were measured with the F1-score. Still, the F$\beta$-score may be sensitive to distribution variations and does not yet overarch the evaluation of a collection of models with varying distributions [16].

### 3.3.4. Independent Averaged Binary Auxiliary Metrics

As previously stated, the binary auxiliary metrics measure performance per model while affected by distribution shifts. Instead, Meijer [1] recommends using metrics that evaluate the performance of a collection of model adjustments, independent of specific models. Ha and Garcia [16] affirms this by finding other metrics that compare various classifiers with varying sample distributions to mitigate the inadequacy of the F-score.

First, the AUROC visually graphs various model performances at a range of thresholds that may still be affected by varying class distributions. less sensitive to data imbalances. In particular, the metric can guide the trade-off between the chance of incorrectly predicted anomalies (FPR) and missed anomalies (FNR) to consider different thresholds of which probabilities are mapped into binary predictions [9]. Furthermore, the area under the curve is a good estimator for model performance with balanced datasets [2]. However, limitations occur from using the AUROC to determine the average model performance with imbalanced datasets. A high AUC score does not necessarily indicate uniform performance across all regions in the ROC [16] and may indicate poor performance in the minority classes. Haurum and Moeslund [2] argue the ROC's resemblance in accuracy which Meijer et al. [9] supplement by mentioning the limited benefits of the ROC curve due to the TN in the ROC's FPR dimensions, where TN is also a component of accuracy. Additionally, the ROC misleads a good performance while evaluating over an unbalanced dataset or also per class, it doesn’t show the certainty of the classifier's performance with confidence intervals, and whether the performance deviations between classifiers are statistically significant. Therefore, other methods that are less sensitive to data skewness can be oriented to improve the insights of the performance evaluation [2].

Therefore, a precision-recall curve provides a visual understanding of the trade-off between the number of false positives and false negatives. In particular, the precision-recall curve is an informative metric while evaluating the performance of the minority class in unbalanced datasets for different thresholds for converting the classifier's probabilities into binary predictions. Furthermore, the area under the PR curve is independent of the threshold ratio of the AUROC and distribution changes while evaluating the performance of an overarching family of adjusted model versions, which makes this metric relevant [9]. Even with the limitations of the ROC, the PR curve contains the effective advantages of the ROC space [16]. Furthermore, the PR curve properly informs deviations in model performances, for instance, the precision metric captures deviations in the amount of false positives [16]. Eventually, an optimal threshold can be chosen based on the project's needs regarding precision or recall. However, the curve encompasses the entire range of precision and recall values, allowing fewer adjustments to preferred metric results [9].

### 3.3.5. Conditional Metrics

Meijer et al. [9] discuss two conditional metrics that focus on metrical adjustments of precision and specificity while keeping recall, also known as sensitivity, at a preferred constant value. Unlike AUROC and PRCurve, which require the $\theta$ threshold to be varied to plot the visual representation of the whole curve, conditional metrics are useful when dealing with substantially imbalanced datasets. The advantage of this method is its ability to relate the importance of specific metrics to the actual needs in the anomaly classification context. classification context. In this case, the recall metric is prioritized since optimally the number of correctly predicted anomalies over all occurring anomalies should be 1, to not miss any.

### 3.3.6. Metrical Averaging

Micro-averaging and macro-averaging for each metric can be utilized to evaluate a classification model's performance across different classes. Micro-averaging averages across all predictions to consider the overall model performance. However, macro-averaging calculates the performance metrics independently for each class by consequently averaging these results across all classes to ensure that each class contributes equally to the final metric, regardless of the number of instances in each class in for instance an unbalanced dataset [14]. The characteristic of analyzing each class equally makes macro-averaging proficient in providing evaluation classification for classifiers trained on imbalanced datasets [19].

# 4. Data Understanding and Preparation

This chapter outlines the processes of various data source concatenations to form a final dataset that can be explored and understood to prepare the data for the modeling phase.

## 4.1. Data Collection

The project's custom dataset is based on sewage inspection footage and labelled data provided by the client through local sewage inspection companies. The data originated from the pipe system of Apeldoorn, the Netherlands, where a robot collected internal sewage pipe footage with fish-eye cameras at the front and back, incrementally capturing images from the start to end of a pipe section. The footage was captured between 2009 and 2023, although the annotation data ranges from 2009 to 2022. The fish-eye footage polar coordinates are transformed into a rectilinear coordinate system and integrated into the client's inspection software, creating 360-degree panoramic images. Trained inspectors analyze these images for occurring observations and annotate them according to NEN norm 13508-2, including their location, type, and severity. Eventually, this annotated data is stored and utilized for predictive analysis for the client's sewage pipe maintenance predictions.

## 4.2. Data Source Descriptions and Concatenation

The final dataset is constructed from the information of five data sources spread over the image directory and CSV files. These five data sources are the images directory, two metadata files from each image folder, the pipe data and the annotated inspection data. From assessing the different attributes of each source, it can be considered which attribute relationships can be made to form a final dataset.

### 4.2.1. Data Sources

First, the most important information from the five data sources is mentioned which is relevant to concatenate the final dataset. First, the image data consists of frame collections representing time-series data from historical inspection records for each pipe, with each frame measured at specific intervals. As illustrated in Fig. A.1 in Appendix A, images are organized into folders named by a pipeid_videoid title, indicating the type of registration pipe with a unique video inspection identifier. Each folder contains an original number of captured images depending on the length of the pipe due to the 5 cm capturing incrementation of the robot through the pipe. Each image is provided in JPEG format with a resolution of 1040x1040 pixels and named with a numerical image index, encoding the distance in increments and the camera direction (front or back). Additionally, overview images are provided, which are concatenations of sequential frame sets, likely used for visualizing the entire pipe within the client's software interface.

Secondly, each image inspection folder contains two metadata folders with crucial information for the data relations with the annotation data. The fileid.json includes the relevant *videoid*, *pipeid*, and inspection date as textual data. The meta.json file contains most importantly the *startPosition* and *StartPos* attributes that indicate the distance from the pipe's starting point to where the camera began inspecting. Further details of the information from the metadata files can be found in Appendix A. Furthermore, the pipe.csv file contains 30,372 rows, representing individual pipes and includes twelve attributes as seen in Table A.1 in Appendix A. Five of these are crucial for the data exploration and preparation phase of this project and are described in Table 1. The "pipe content" defines whether the pipe carries waste, rainwater, or a mixture.

| Attribute | Data Type | Value |
|---|---|---|
| pipeid | textual | pipe Uuid, eg. 00094c07091a4c689781f6c0bf9f31f2 |
| pipecontent | Categorical | MixedSewer, WasteSewer, RainSewer, Unkown |
| pipefunction | Categorical | Transport, Infiltration, Culvert, Unkown, Connection, Drain |
| pipeshape | Categorical | Round, Unkown, Ovoid, Square |
| material | Categorical | Normalized material of the pipe: Concrete, PVC, Other, PE, Unkown, HDPE, VitrifiedClay, ReinforcedConcrete, Steel |

*Table 1: Pipe dataset and its attributes, their importance, datatypes, and explanations or examples.*

Additionally, the inspection data contains rows corresponding to single observations, uniquely identified by a combination of a *pipeid* and *inspectionid* as can be seen in Table 2. Important to note is that the *inspecitonid* of the annotation data and the *videoid* of the fileid.json differ since one *inspectionid* can consist of multiple video folders with distinct *videoid's* as multiple inspections can occur during the same pipe inspection at the same inspection date time. Furthermore, certain inspections can contain the same pipe, as a single pipe can be inspected at different date-times throughout the years. Additional attributes of the inspection data on top of Table 2 can be seen in Table A.2 in Appendix A which may be of future interest, such as the severity rate per observation. The *code* attribute consists of three capital letters indicating an observation and providing additional information. Each code belongs to a specific code class, which can pertain to the pipeline's fabric, operation, inventory, or other aspects.

| Attribute | Data Type | Value |
|---:|---|---|
| pipeid | textual | pipe uuid |
| inspectionid | textual | inspection uuid |
| datetime | date/time | inspection date, eg. 20/08/2010 |
| position | numerical | in-pipe position in m |
| end_position | numerical | end position in pipe in m |
| code | categorical | damage code, eg. BCA |

*Table 2: Annotation dataset and its attributes, their importance, datatypes, and explanations or examples.*

In conclusion, the five data sources were concatenated based on overlapping attributes based on the relations in the entity-relationship diagram in Fig. A.3 of Appendix A.

### 4.2.2. Final Dataset

To identify correlations between different datasets, the pipe and inspection dataset attributes are merged with the image data frame throughout a framework to form a final dataset as seen in Fig. 1. In the following sections, the framework is explained in a five-step-wise manner where the number of rows is often considered including front and back images. By dividing the number of frames, the number of front and back images separate can be obtained.

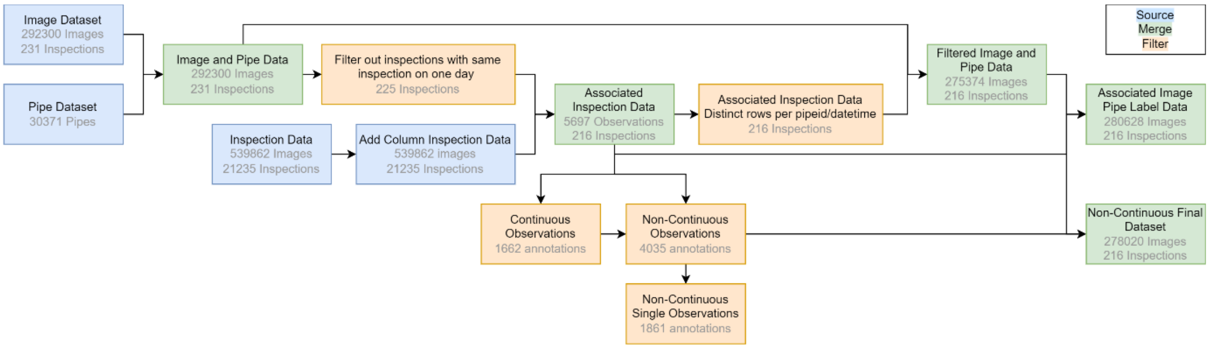


*Fig. 1: Total image, pipe, and inspection datasets merging into a single dataframe.*

#### 4.2.2.1. *Image Dataset Extraction*

The image data spread over the image directory is converted into a single data frame, allowing collective assessment to facilitate image data exploration and future data preparation. The Python-based conversion framework returning a Pandas data frame is depicted in Fig. 2, and the corresponding code is available in the included *data _understanding_preparation.ipynb* Jupyter Notebook. The framework includes a primary function that returns a data frame, incorporating results from five subfunctions that gather necessary data for each image.

First, all image folders within the directory are processed using the *new_image_df* function by iterating over each image folder. Secondly, the pipeid and videoid are extracted from the folder names, as each inspection folder is titled using a combination of these identifiers (pipeid_videoid). Thrithy, for each folder path, two functions extract metadata from the folder's JSON files. The inspection date is obtained by accessing the fileid.json file and extracting the datetime string in the year-month-day format, as shown in Fig.. Similarly, The start position offset of the camera, relative to the official start of the pipe, is extracted from the meta.json file by loading the file and retrieving the startPosition value. If the startPosition value exceeds 1000, indicating an error, the startPos value from the properties list is used instead. Next, image files are extracted using the corresponding folder path where each image is processed to extract the image index number and direction from the string format of the image titles. Hereafter, For each image, a constructed record is appended to the data frame, containing data points such as pipeid, videoid, start position, inspection datetime, image path, direction, and image number. Finally, the data frame is returned after all images are processed.

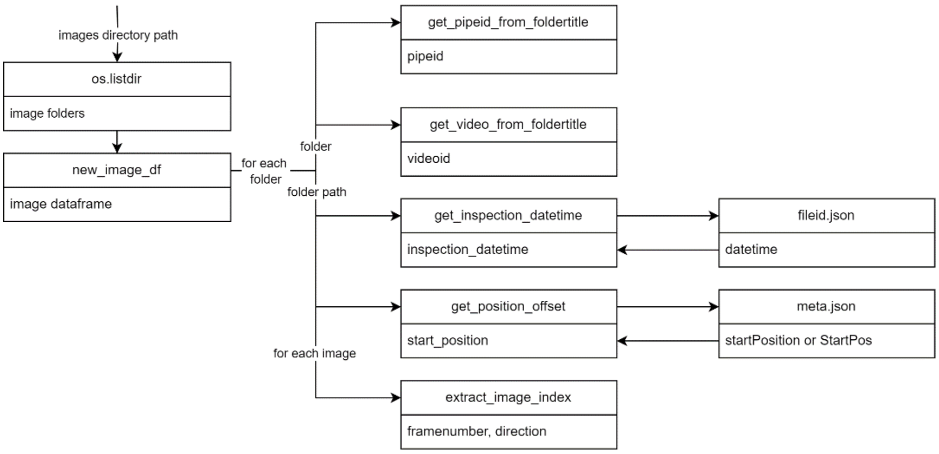
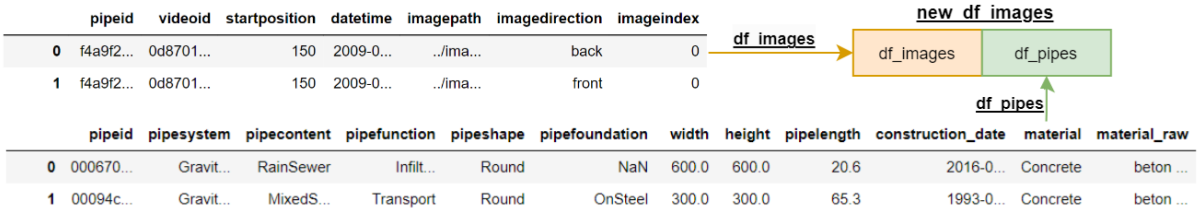


*Fig. 2: Image directory to data frame conversion by inherited function call-and-return structure.*

### 4.2.2.2. *Image and Pipe Data*

Subsequently, the properties of each distinct pipe can be merged with the records from the image data frame (see Fig. 3) and those from the pipe data frame where the *pipeid* matches. In this case, all pipe properties are maintained such that in-depth exploratory data analysis can be performed. The resulting data frame contains the same number of records as the image data frame (292300 records) as the data is based on the number of available images.



| | pipeid | videoid | startposition | datetime | imagepath | imagedirection | imageindex |
|---|---|---|---|---|---|---|---|
| 0 | f4a9f2... | 0d8701... | 150 | 2009-0... | ../ima... | back | 0 |
| 1 | f4a9f2... | 0d8701... | 150 | 2009-0... | ../ima... | front | 0 |

| | pipeid | pipesystem | pipecontent | pipefunction | pipeshape | pipefoundation | width | height | pipelength | construction_date | material | material_raw |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 000670... | Gravit... | RainSewer | Infilt... | Round | NaN | 600.0 | 600.0 | 20.6 | 2016-0... | Concrete | beton ... |
| 1 | 00094c... | Gravit... | MixedS... | Transport | Round | OnSteel | 300.0 | 300.0 | 65.3 | 1993-0... | Concrete | beton ... |

*Fig. 3: Merging of the pipe data into the image data.*

### 4.2.2.3. *Two Separate Inspections Forming One*

Even though multiple inspections can occur at the same pipe, this can also happen at the same inspection datetime, which can be an issue during the connection with the inspection data. Particularly, It was found that 6 image folders were of the same pipe and inspection datetime (see ), although a different startposition, framecount and videoid. For each double inspection, there is either a blockade by a root or a deposit which seems to let the inspection start again, but then from the other side of the pipe. Eventually, the pipe is annotated as one inspection file, however, the inspector has been started at the end frame of the second inspection to go sequentially again through the pipe. Therefore, the inspection file is usable, but the image order of the second inspection order should be reversed to connect it with the annotation data. Since the concatenation of the annotation data to the reversed order of images on top of the beginning images may be error-prone, these inspections are left out. However, future investigations may construct a framework to include the reversed image order. In conclusion, this resulted in 225 inspection image folders were available to merge the inspection data with the image data.

| pipeid_videoid | startposition | framecount | datetime | quality |
|---|---|---|---|---|
| 2cad0ff9589f4277aa08b1da031aec76_ a0c3d505ac7f4f2d894e3a9f56873166 | 55 | 13 | 2009-07-01 | At the end stopped by roots |
| 2cad0ff9589f4277aa08b1da031aec76_ aecb9a2db9df47e7bbe78ebdc9d76185 | 140 | 856 | 2009-07-01 | At the end stopped by roots |
| 940f179616b94803b0c8ba9c556f68cf_ 959a3a288972407ba81e839f43407519 | 65 | 310 | 2020-02-12 | At the end blocked by attached deposits |
| 940f179616b94803b0c8ba9c556f68cf_ b591e65a357e4f2db6983a7aee20300d | 35 | 462 | 2020-02-12 | At thend blocked by deposits |
| 87100098a6ab4bc3bb5a60b72d90f354_ 31f19b23048b48bcbc0c29cb2270b7a5 | 65 | 406 | 2020-01-27 | At the end blocked by roots |
| 87100098a6ab4bc3bb5a60b72d90f354_ a7eb076e64b94d3bba9bcf52f53979d7 | 85 | 218 | 2020-01-27 | At then end blocked by roots |

*Table 3: Image folder pairs which belong to the same inspection at the same datetime, however belong to different image folders.*

### 4.2.2.4. *Associated Inspection Data*

The associated inspection data represents all the inspection data available for the current project and image folder. The initial inspection CSV data contains 21,235 distinct inspections with 539862 rows of distinct observations. Certain columns of this data were filtered out to remain the most relevant attributes (see Table 2). Hereafter, the image and pipe dataset has been reduced to 225 rows, only obtaining records with distinct *pipeid*, *datetime*, and *startposition* attributes to act as a filter during the merge with the inspection data. This merge resulted in an associated inspection data frame with 5697 records. Additionally, to provide the incremental index location of each observation connectable to the image indexes, two additional image index and end image index columns were constructed for each observation in the associated inspection data frame. In, particular, the image index represents the position of the anomaly for 5 cm increments and the start position is the offset due to a different start point of the inspection compared to the initial pipe start. The values of the values are calculated based on the formula in Fig. 4 where the start position values are from the new image data frame along with the position and end position attributes from the inspection data frame. In conclusion, the associated inspection data is created which responds to the available inspection data for this project. However, not every image inspection folder has available annotations, and the annotations data needs to be located in the corresponding frame records at certain positions.

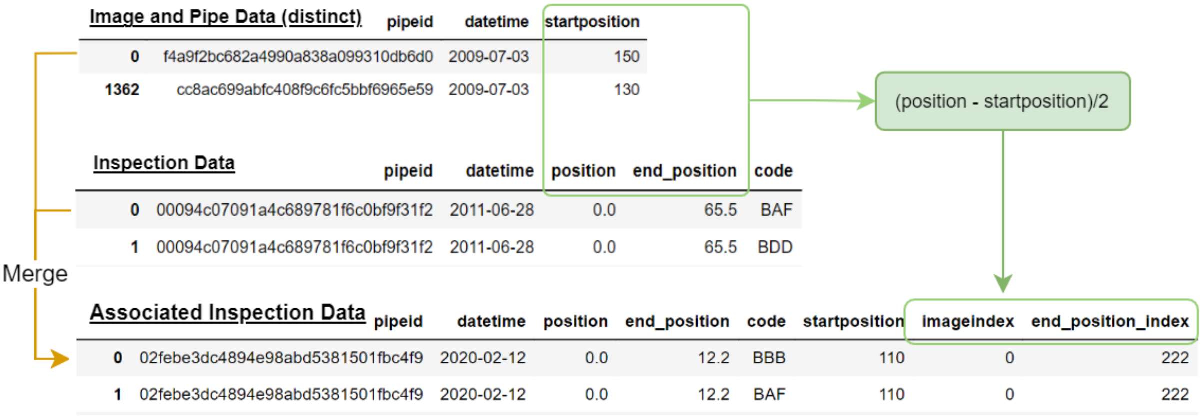


Image and Pipe Data (distinct)

|  | pipeid | datetime | startposition |
|---|---|---|---|
| 0 | f4a9f2bc682a4990a838a099310db6d0 | 2009-07-03 | 150 |
| 1362 | cc8ac699abfc408f9c6fc5bbf6965e59 | 2009-07-03 | 130 |

Inspection Data

|  | pipeid | datetime | position | end_position | code |
|---|---|---|---|---|---|
| 0 | 00094c07091a4c689781f6c0bf9f31f2 | 2011-06-28 | 0.0 | 65.5 | BAF |
| 1 | 00094c07091a4c689781f6c0bf9f31f2 | 2011-06-28 | 0.0 | 65.5 | BDD |

Associated Inspection Data

|  | pipeid | datetime | position | end_position | code | startposition | imageindex | end_position_index |
|---|---|---|---|---|---|---|---|---|
| 0 | 02febe3dc4894e98abd5381501fbc4f9 | 2020-02-12 | 0.0 | 12.2 | BBB | 110 | 0 | 222 |
| 1 | 02febe3dc4894e98abd5381501fbc4f9 | 2020-02-12 | 0.0 | 12.2 | BAF | 110 | 0 | 222 |

*Fig. 4: Uitilzing the inspection startposition and the anomaly positions to calculate the image index in the image titles.*

#### *4.2.2.5. Filtered Image and Pipe Data*

Certain image inspection folders are excluded during data collection as specific available image inspection folders lack recent annotation data while creating a new merged dataset with located annotations linked to image frames. The available associated annotation data is decreased to 216 distinct inspections by removing duplicates while keeping a unique combination of *pipeid*, *datetime*, and the *startposition* for each row. As a confirmation, by removing duplicates based on the *inspecitonid*, the decreased associated annotation data will also contain 216 rows. Eventually, this would mean that for our dataset, there are 216 inspections available for some of the 231 image inspection folders. Since this project mainly focuses on supervised learning, the unlabeled images are not of interest and may reduce the model performance if they are included in the dataset. Therefore, to filter out the image inspection folders, the reduced associated inspection data frame is merged with the image and pipe data, resulting in a filtered image and pipe data frame. The size of the filtered data frame (275374) compared to the image pipe label data (292300) is reduced since 12 image inspection folders do not have available inspection data.

#### *4.2.2.6. Final Dataset*

The final dataset is constructed by merging the filtered Image and Pipe data with the associated inspection data, providing the annotation data for each image. The merge is based on the established relationship based on the *pipeid*, inspection *datetime*, *startposition*, and frame index. The final dataset counts 280628 rows covering 193 pipes, which is an increase compared to the actual relevant and available images in the filtered image and pipe data (275374 records). This increase is explainable due to multiple annotations made for the same frame location in an inspection. Particularly, records in the beginning and ending often contain more annotations for the same beginning or ending frames, causing image duplications while the annotation rows change.

## 4.3. Data Exploration

For the following data exploration phase is explored of the final dataset how the different inspections vary in their inspection datetime, framecount, pipe attributes and available observaitons. Particuarly, this allows to find the relationships between certain attributes to condier which relevant data points to include.

### 4.3.1. Images and Inspection Datetime

The reason why the final dataset is made for a filtered selection of image inspection folders is due to the mentioned lack of annotation data for all available images. To gain more insight into the 12 inspection image folders lacking annotation data, Fig. 5 shows that especially, the newest inspection image folders do not yet contain correlated annotation data. These could be provided by the client soon. Nevertheless, three inspections missed data occurred in 2009 and two in 2020. Furthermore, from this figure can be concluded that most inspections were performed on 28-1-2020. This means that multiple inspections can occur on the same day, although the pipes that were inspected twice on the same day are excluded from this dataset. Additionally, pipe inspections occur frequently around every 10 years, since there is a big gap between 2009 and 2020.

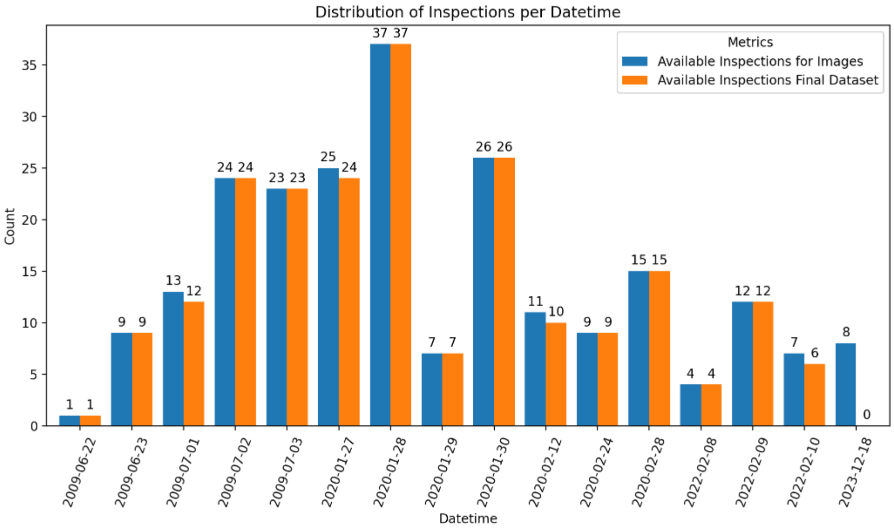


*Fig. 5: Number of inspections with available annotation data.*

### 4.3.2. Inspection Framecount Distribution

The exploration of the deviations in frame count per inspection could be relevant in determining how distinct inspection image folders can be divided over the train and test set to account for overfitting on the same pipe appearance. Fig. 6 shows the frequency distribution of the number of labelled inspections consisting of certain image sizes (each image could be either shot in the front or back direction). From observation can be seen that there is a major inconsistency between pipe inspections where the shortest pipe contains 8 images and where the largest pipe contains 3044 images. Important to note is that the inspected pipes do not need to be of the same length as the inspection sizes, often inspections do not capture full pipes. Furthermore, these frame count variations likely affect the source of imbalance in this dataset. Especially, since larger inspections contribute a more significant proportion of the total dataset with their own static visual appearance. Therefore, dividing inspections between training and test sets should be performed with care to ensure the training set contains a variety of appearances, and not for instance being trained solely on the largest inspection folders. Stratified sampling can be used to split the inspection images into training, validation, and test sets such that the image inspection length characteristics are proportionately represented in both sets.

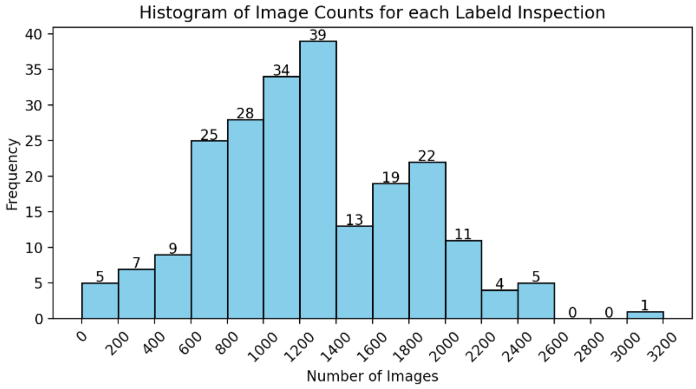


*Fig. 6: Histogram of the distribution of the frequency of pipe frame sizes, b) and corresponding statistical measures..*

### 4.3.3. Pipe Attributes

The categorical attribute comparison between pipe materials and shapes shows the dominance of concrete round pipes throughout the dataset. In particular, the inspection and pipe count is compared since is found that for 29 pipes, two inspections occur. For the comparison is a logarithmic counting distribution utilized to gain intuitive deviating insights with clear numerical labels. Besides, three main findings are noted. First, by obtaining the number of concrete pipes from Fig. 7 is calculated that concrete round pipes dominate for 94% of the amount of available labelled images. This percentage contains also 28 of the pipes with two inspections. Secondly, ovoid concrete and round PVC and HPDE pipes are the most rare material attributes occurring in the dataset. In closing, as the pipe material contributes mostly to a different visual appearance, additional categorical attributes besides the pipe shape can be investigated to find more prominent correlations.

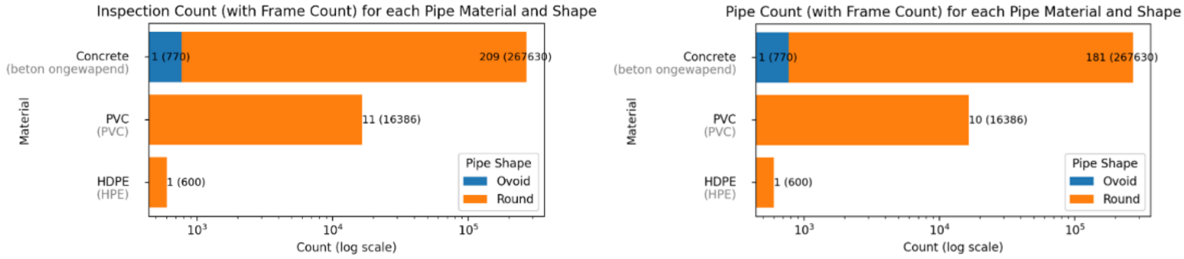


*Fig. 7: Image and a) inspection count and b) pipe count for each pipe material and shape. The pipe count is lower since inspections can occur twice in the same pipe.*

The qualitative and quantitative comparison between categorical attributes of the pipe content, material, and function shows three main correlations. Meanwhile, a sub-comparison is performed between the inspection and pipe count, where the inspection count includes inspections occurring in the same pipes. The following assumptions are grounded in the differences observed in Fig. 8, Table 4 and Fig. 9. First, rain sewer pipes function solely as infiltration pipes and are mostly constructed of PVC. Similarly, wastewater pipes consist solely of concrete, although function as transport. Secondly, mixed sewer pipes, used exclusively for transport, are primarily made of concrete, with a minor presence of HDPE and PVC pipes. Thirdly, a qualitative image inspection in Table 4 shows the distinct pipe appearances between the combinations of material and content observed in Fig. 8. In particular, PVC pipes used as mixed sewers and rain sewers contain varying appearances due to a grey plain shape, while PVC in sewage pipes contains green neon stripes. Concrete pipes in mixed, rain, and wastewater systems also appear differently in colour. However, even the images of concrete pipes in mixed sewers (appendix B) reveal that concrete pipes can vary in colour from reddish and beige to silverish. This variation likely depends on the lighting from the inspection robot or the effects of gases and substances inside the pipe.

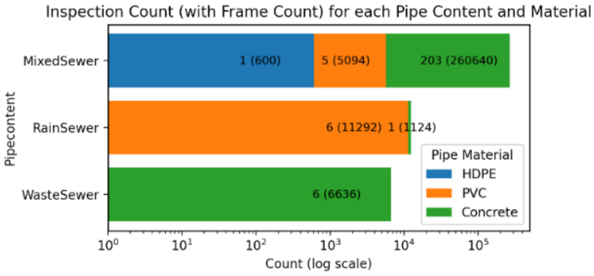


*Fig. 8: Pipe material and content proportions for the number of pipes (a) and inspections (b).*

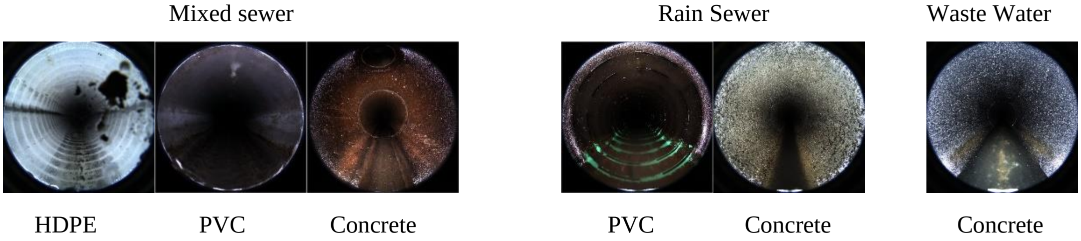


*Table 4: Qualitative comparison between pipe content and pipe materials.*

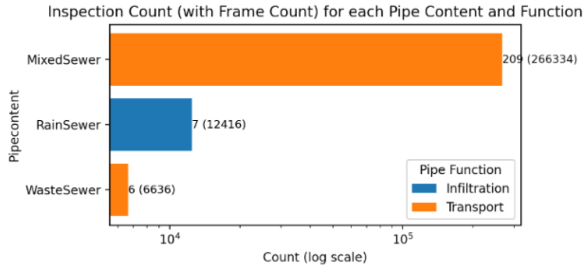


*Fig. 9: Proportions between the number of pipes (a) and inspections (b) for pipe content and function.*

The dataset focuses solely on commonly found concrete pipes in mixed and wastewater sewers functioning as transport sewers to help the model learn specific observation deviations without the material being a confounding factor. In particular, creating a controlled model environment allows the classifier to accurately distinguish various observations without the complexity introduced by different pipe materials. Concrete pipe images are significantly represented in real-life sewage systems [4], accounting for 94% of the dataset (see Fig. 7). Besides, the mixed and wastewater sewers mostly consist of concrete pipes (seeFig. 8).

### 4.3.4. Observation Exploration

To perform multi-label classification containing at least a defect and structural elements the occurrence of available observations is explored. Therefore, first is the relevance of including certain observation codes assessed. Secondly, prominent remarks are explained from continuous and non-continuous observations from observation distributions. Lastly, the impact of co-occurrence is considered on the selection of the observations.

#### *4.3.4.1. Available Observations*

The descriptions of the code-represented observations are written out and categorized according to NEN norm 13508-2 to inform the inclusion or exclusion of certain observations from the dataset. Table 5 shows the available observations of the final dataset sorted into five different code classes where each observation is assigned the relevancy to be included in the final dataset. This relevancy depends on whether an observation is considered a defect and structural element since other observation types do not yet belong to the aim of this project. In Table 5 can be seen that mainly the relevant observations belong to the *Codes relating to the fabric of the pipeline* and *Codes relating to the operation of the pipeline* code groups. However, regarding the other three code groups, including the connections (BCA) is relevant to the client's request to include the basic model skills and including the water level (BDD) could be important the water level could be a defect when its level increases.

Additionally, the exclusion and usefulness of less relevant observations are justified. The general photograph (BDA), general remark (BDB) and the observations from the *Codes changing header information* are not relevant to be classified for this project purpose and, therefore excludable. However, three observations could be useful during the data filtering process. The start node (BCD) and finish node (BCE) are important to determine the interval of captured frames between the corresponding start and finish nodes of the inspected pipe. Nonetheless, the nodes may not align with the start and end of the pipe due to the corresponding start position offset. Moreover, the finish node may not be present in each inspection since these inspections likely end with an *inspection terminated before finish node are notated* observation. These latter inspections likely exclude images with manhole pipes at the end of the inspection which contain a different appearance compared to the tubular sewage pipes due to their geometric inconsistency. Thirdly, the loss of vision observation is important to locate the frames which may be filtered out due to inconsistent vision.

| **Code Groups and Observations** | **Relevancy** | **Description** |
|---:|:---:|:---|
| Inventory Codes | | |
| BCA | x | Connection |
| BCD | | Start node type |

| | | |
|---|---|---|
| BCE | | Finish node |
| Other Codes | | |
| BDD | x | Water Level |
| BDA | | General photograph |
| BDB | | General remark |
| BDC | | Inspection terminated before finish node |
| BDG | | Loss of vision |
| Codes relating to the fabric of the pipeline | | |
| BAF | x | Surface damage |
| BAJ | x | Displaced joint |
| BAB | x | Fissure |
| BAG | x | Intruding connection |
| BAH | x | Defective connection |
| BAN | x | Porous pipe |
| BAA | x | Deformation |
| Codes relating to the operation of the pipeline | | |
| BBF | x | Infiltration |
| BBD | x | Ingress of soil |
| BBB | x | Attached deposits |
| BBA | x | Roots |
| BBC | x | Settled deposits |
| BBE | x | Other obstacles |
| Codes changing header information | | |
| AEF | | Pipe Unit Length |
| AEA | | Video volume reference |
| AEC | | Shape |

*Table 5: All present observation codes are sorted into code classes with relevant assigned codes.*

#### *4.3.4.2. Observation Distribution*

The 5697 available observations can be shown in a frequency distribution to highlight findings related to data imbalance and the division between continuous and non-continuous observations throughout the pipe. First, the frequency observation distribution in Fig. 10 shows a significantly skewed and imbalanced observation set, where certain observations are severely underrepresented compared to the two highest majority classes. The occurrence rate of connections (BCA) and water levels (BDD) can be clarified by the fact that connections (inlets) occur at regular intervals since they are structural elements, and water levels are often standardly annotated since mainly there is a small water stream in the pipe after the pipe has been cleaned before inspection. However, certain observations occur for a longer duration throughout the pipes compared to observations occurring at one frame location [5].

Secondly, from Fig. 11 with a logarithmic y-axis, a second finding can be made that certain observation types contain either a lot of continuing observations or a very small amount compared to the total observations. The observations with significant continuous distances can be clarified by the fact that for instance water level (BDD) may have a certain level throughout the whole inspection, or surface damages (BAF) and attached deposits (BBB) could occur for shorter-continuous distances. Nevertheless, fissures (BAB) and other obstacles (BDG) are likely to occur over a couple of frames. Additionally, the AEF represents the pipe unit length, specifying the length of the to-be-inspected pipe at the beginning of the annotation. In addition, roots (BBA) and other obstacles (BDG) are likely to range not the whole pipe long, especially if one inspection contains only roots (the small one). Subsequently, decisions can be made regarding the inclusion of continuous observations and the type of single-occurring observations.

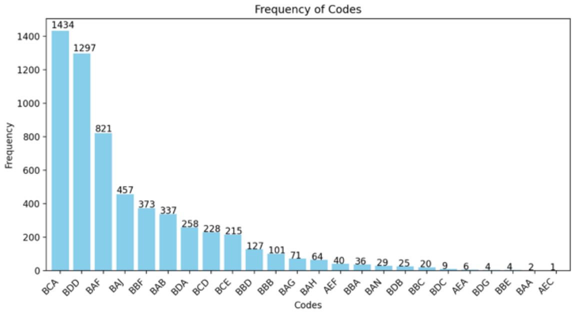


*Fig. 10: Frequency distribution of distinctly noticed observations.*

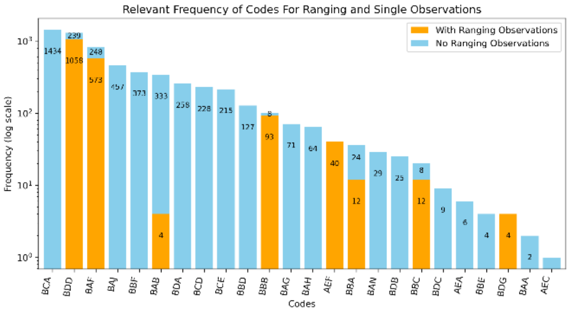


*Fig. 11: Frequency distribution on a logarithmic scale, the differences between continuous and single occurring observations.*

## 4.3.5. Code Selection

As the previous observation distribution includes all available observations, not all are equally relevant for the multi-label classification of defects and structural elements in this project as seen in Table 5. Therefore, the observation distribution can be reduced to focus on the most relevant codes and non-continuous observations based on their numerical presence to create a dataset that includes at least one defect and one structural element while mitigating data imbalance. It was already found that the most relevant codes mainly relate to the *fabric and operation of the pipeline* observation classes, as these are mostly defects.

### *4.3.5.1. Continuous Observations*

First, including continuous observations over larger pipe distances enhances the model's ability to learn temporal patterns and distinguishing features although necessitating model architecture adjustments to account for varying severity levels. In particular, the inclusion of continuous observations is supported by two grounds based on observations from Fig. 12. First, excluding the largest relative continuous observation classes (BDD, BAF, BBB, BBC) would neglect significant portions of the dataset as most observational data is already rare due to the imbalanced dataset. Especially, removing the BDD and BAF neglects to learn about a significant amount of observations throughout the real-life sewage infrastructure. Furthermore, excluding continuous observations could impact multi-label classification, as non-continuous and continuous observations may coexist in one frame, misleading the model into learning continuous patterns belonging to non-continuous observations. Additionally, incorporating labelled continuous observations necessitates classification at different severity levels, as most do not exhibit uniform severity, to eventually allow for more robust differentiation between various observations. However, neglecting severities while including continuous observations may cause model overfitting to certain severity levels, treating frames of varying severity identically and thereby introducing bias. For this project, continuous observations are excluded by retaining the corresponding frames but removing their labels, focusing solely on non-continuous observations. Still, this approach anticipates potential model adjustments for severity-level inclusion.

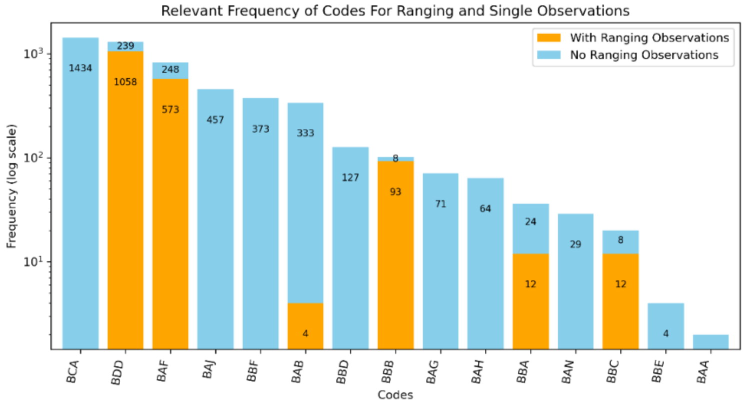


*Fig. 12: Number of observed relevant observations at both single or continuous instances.*

#### *4.3.5.2. Code Selection*

Since future work may focus on classifying continuous observations with corresponding severities, this project mainly focuses on constructing a final dataset with at least one non-continuous defect and one structural element. To consider which non-continuous observations to include, their distribution can be explored. Therefore, Fig. 12 can be filtered to show solely a distribution of non-continuous observations (see Fig. 13). It can be seen that inlets (BCA) are the dominant observation due to their frequent stationary appearance as a structural element. However, the other observations (mostly defects) occur indeterminably and significantly less. The displaced joints (BAJ), infiltrations (BBF), fissures (BAB), surface damage (BAF), and water level (BDD) are the largest minority classes. The other minority defects in the distribution's right tail contain fewer observations. Especially, the attached deposits (BBB), settled deposits (BBC), other obstacles (BBE), and deformations (BBA) are underrepresented with less than 10 observations of each class.

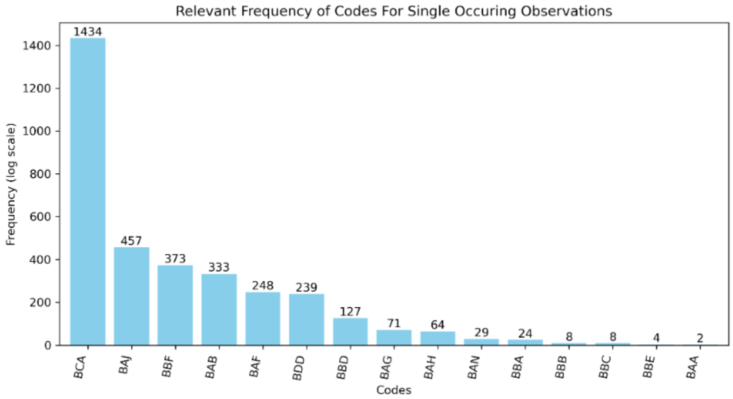
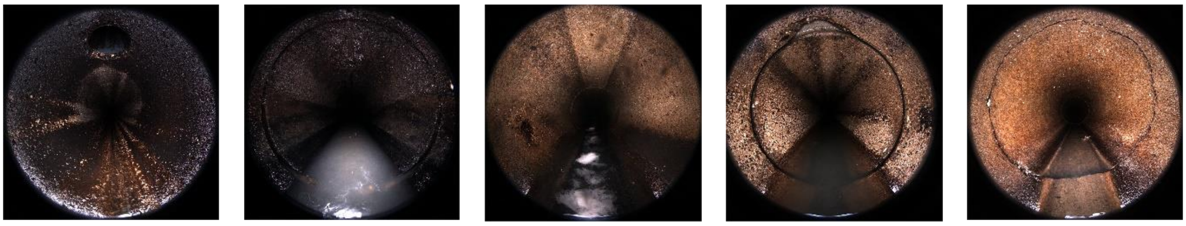


*Fig. 13: Frequency distribution of the relevant single-occurring observational codes.*

Previous exploration supports two arguments in determining which observations to include that have sufficient data for potential upsampling. Haurm and Moeslund [4] discouraged maintaining a tailed distribution with significantly underrepresented samples if no compensation is performed by adding new methods that could deal with these rare observations. Therefore, in this dataset, the nine smallest minority classes can be filtered out due to their low frequency and likely increasing overfitting from being upsampled. Nevertheless, in the case of implementing other balancing methods than upsampling, could be considered including the nine observations to increase the generalization capability of the model. Secondly, the non-continuous water level observations (BDD) seem to be inconvenient due to the temporal aspect of water influencing most of the frames around the annotated frame, therefore non-continuous water level observations are excluded. In conclusion, this results in four defects and one structural element that could be potentially included during data balancing with resampling (see Fig. 14).



Inlet
(BCA)

Displaced Joint
(BAJ)

Infiltration
(BBF)

Surface Damage
(BAF)

Fissure
(BAB)

*Fig. 14: Qualitative assessment of potential single-occurring defects and structural elements, images are taken two frames earlier than the annotated frame.*

### *4.3.5.3. Co-occurrence*

Since non-continuous observations and visually similar observations may occur together, it could be considered to include one defect and one structural element where the co-occurrence of multiple observations in one frame is minimalized. Fig. 15 can be inspected to explore the frames that contain the non-continuous observations either multiple or only one observation occurring in one frame. Particularly, describing the difference between multi-class and multi-label classification. It can be seen that half of the inlets (BCA) and almost all displaced joints (BAJ) occur unaccompanied with another observation in one frame. These single-occurring observations could be convenient when balancing a dataset with resampling methods. Especially since upsampling frames with two or more observations directly causes the occurrence amount of all included observations to increase due to the multi-label annotations. Hence, it may be handy to only implement the BCA and BAJ codes to still be able to evaluate the model for multi-label classification and develop a procedural framework that can increase codes for future improvements. Nevertheless, only focusing on frames with single-occuring observations with up- or downscaling may affect the model's capability to perform multi-label classification as increasing the frames with single codes may cause the model to rather perform as a multi-class classifier. Therefore, alternative balancing methods may be considered for multi-label classification, such as cost-sensitive or ensemble learning methods.

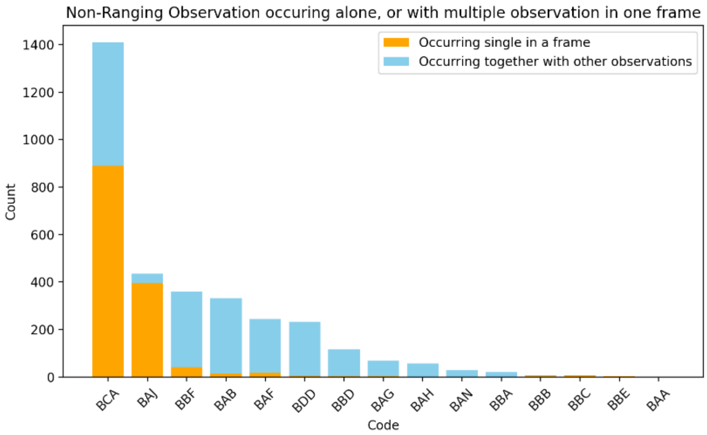


*Fig. 15: Frequency of non-continuous observations occurring alone or with multiple observations in the same frame.*

## 4.4. Data Quality

Besides assessing the frame correlations between categorical attributes, data qualities are assessed to determine potential non-representable features (model bias) by assessing four visual appearance aspects related to this dataset. Hence, this creates a foundation for informed decision-making for the inclusion and exclusion of certain frames, avoiding the model to learn on unclear and correct features. For this quality assessment, two divisions can be made between the quality assessment methods based on the complexity of improving image qualities. The deviations in brightness, error-prone images, and beginning and end frame inclusion are accessible and mitigate image aspects, although temporal mislabeling requires careful adjustment methods and experimentation to capture the correct features.

### 4.4.1. Start and End Images

Inspections could contain images outside the tube with a different appearance at the inspection start or end, causing inconsistency in visual appearance. In particular, manhole intersection shafts, at the beginning or end of a pipe, are often visible in the back images of beginning frames, and in front images at the end. However, this is only the case for the pipes with a startposition of 0 since the inspection start node aligns with the start of the inspected pipe (for 39 inspections in the dataset). In this case, if the inspection also contains the ending node, then the last front images capture the manhole shaft at the destination, while the back images obtain an error blue colour. In the following Fig. 16 can be seen that for two inspections with a start position of 0, the starting front and back images capture the beginning shaft, and the last front images capture the destination shaft. Moreover, for some inspections, the shaft may be visible for multiple frames before and after the start and end node annotation, therefore an increased amount of these visually different images may be unpreferably present in the dataset.

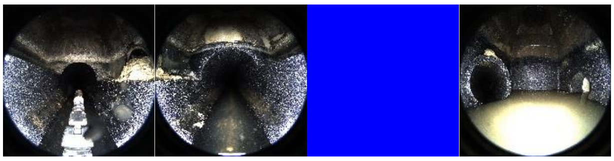


*Fig. 16: a) Start back and b) front images, c) end back and d) end front images.*

Hence, start and end image ranges are filtered out of the inspections with a startposition of 0. A qualitative assessment was performed (see Table 6) with 7 random samples, where the location when the start or end hall is not visible anymore in certain front- and back-frame indexes is determined. For the pipe beginnings, the front frames up to index 10 and back frames up to 21 are excluded as Table 6 shows that at max these frames could capture the manhole shaft. However, for the pipe endings, some inspections do not show the manhole shaft at the pipe end since the inspection is not the same length as the pipe. Therefore, the front frames after the last 38th frame are excluded only for the inspections where the end node (BCE) position corresponds to the pipe length.  Furthermore, the back images at the end of the pipe are not included in Table 6 since later is explained that the back images at inspection endings tend to get blue due to an error. Additionally, some inspections provide more images before the start-node (BCD) and after the end-node (BCE) label, however since the remaining images are not annotated, they can be filtered out. In conclusion, most non-tubular frames are excluded, although, since each pipe differs, future filtering could exclude image ranges for each pipe rather than applying a uniform filter across the entire dataset.

| | | Pipe Begin | | Pipe End | |
|---|---|---|---|---|---|
| pipeid | Frame count | Front | Back | Front | Difference with frame count |
| 37633324c714478fb407cad806d3d19b | 413 | 6 | 14 | 397 | 16 |
| a2af7a3b3faa4838b8ad76bfbfc4df0a | 771 | 4 | 13 | - | - |
| 07c2731397cc4e159edac7145c15ad2f | 827 | 7 | 15 | 807 | 20 |
| b65813d661d140b58a76d676b742f785 | 923 | 5 | 15 | - | - |
| 8142c26421e04f818c0c268593d57edc | 943 | 7 | 21 | 905 | 38 |
| 542ae8de3f1d46c28510d29589be225b | 770 | 8 | 16 | - | - |
| f45f10dc14594dd9bd78cc41fe107e32 | 621 | 2 | 8 | 599 | 22 |
| c77e237c73c9423ca5638eac81c799ea | 369 | 10 | 20 | 348 | 21 |
| 752b7fc75d07451c8e5e352e38774828 | 886 | 7 | 18 | - | - |
| 06e1355cdcfc4da7bd414abeae24769a | 261 | 2 | 10 | - | - |

*Table 6: Qualitative inspection to localize at which frames the shaft at the beginning and end of the pipe is not visible anymore.*

### 4.4.2. Lightning Conditions

A mean brightness analysis is performed to assess the lighting variations of certain inspections to target specific brightness mitigation values and importance to certain inspections. Particularly, this analysis includes inspections occurring in the same pipe. First, as found earlier, certain inspections are performed in the same pipe, indicating that nearly identical frames might be present in the dataset. However, these frames may slightly differ due to variations in the starting reference point or frame counts. In addition, further investigation into the metadata between the inspection in the same pipe at varying dates revealed that earlier inspections occurred in 2009 and the subsequent ones in 2020 (see appendix B). Consequently, the brightness difference between the two sets of inspections can be observed in Fig. 17 where the brightness decreases for the later inspections compared to the earlier ones. This finding prompted an examination of the overall brightness distribution of the images, where Fig. 18 illustrates a skewed mean brightness distribution to the left, indicating an imbalance in pipe brightness across the dataset. Finally, for a comprehensive overview, qualitative sampling was employed by plotting an inspection image from each inspection in a grid ordered by mean brightness, revealing the dominance of less bright images, with a few extremely bright images (see Appendix B).

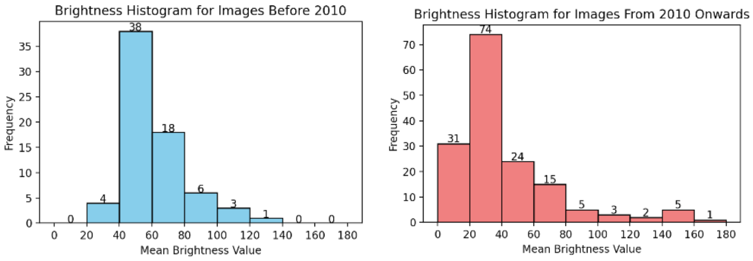


*Fig. 17: Average brightness distribution of each 7th inspection image a) before 2010, and b) afterwards.*

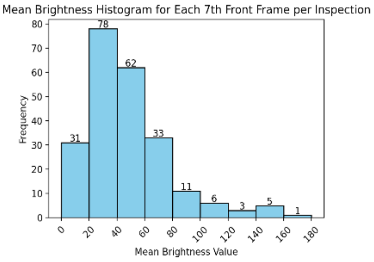


*Fig. 18: Mean average brightness distribution over each 7th image of each inspection.*

This causes two choices to consider, whether to include the double pipes and to apply color changes

#### 4.4.2.1. Multiple Inspections per Pipe

Since the visual appearance of inspections occurring in the same pipe at different inspection times could differ, two reasons motivate including earlier and later inspections to increase the number of observations while considering the risks of possible data leakage or bias. First, correlating findings from Fig. 7 and Fig. 17 are indicating that the 29 earlier inspections have improved quality due to higher brightness than the 29 later inspections. Second, the surface appearance may differ as internal pipe conditions could deteriorate over time. Both factors likely reduce classifier bias towards the same pipe appearance. However, data leakage might occur if inspections of the same pipe are divided over the test and training sets. To prevent this, inspections of the same pipe should be assigned to either the training or test set, Nonetheless, this could lead to overfitting as the same pipe's content of both inspections is nearly duplicated within the dataset.

#### 4.4.2.2. Controlled Illumination and Augmentation

Two methods are utilized to maintain controlled lighting conditions minimizing lighting and color variability, while the training data is augmented to enhance the model's robustness under extreme illumination and color variations. Especially, varying lighting conditions, influenced by the robot's light source and camera settings such as exposure, white balance, and ISO can introduce inconsistencies affecting the model's performance. First, histogram equalization is applied to colored frames to improve illumination consistency across all inspections and ensure the classifier can process colored images. This is done by only equalizing the luminance of the YUV color space of the image. Secondly, color channel normalization, based on the mean and standard deviation of the training data images, is implemented to maintain consistent features, given the color variations of concrete pipes in red, silver, and beige. These two approaches can also transform the test data, serving as pre-processing techniques for handling unseen data in real-world scenarios. Despite the potential of these methods, extreme lighting inconsistencies may persist. Therefore, two augmentation transformations are applied to the training set. First, color jittering modifies brightness, contrast, saturation, and hue to enhance the model's robustness to the color variations of concrete pipes. Additionally, random shadows are introduced to mimic scenarios where shadows and highlights obscure features, such as when the robot's light approaches an obstacle, root, or deposit closely.

…figure of all transformations…

### 4.4.3. Erroneous Images

As brightness analysis reveals images with poor illumination, error-prone images may even lack entire visually relevant imagery which can be inspected and filtered out in three ways. First, four frame ranges are labeled with a loss of vision ('BDG) observation as seen in Fig. 11. Specifically, this observation is continuously noted 4 times in the same inspection (with pipeid 30a59fb3d9194069bf92c7ead5caab05 and videoid 5f772786f5b1454fb3269d905d9dc619). Particularly, the front camera contains a small obstacle object at the bottom of the front image, but most of the area of the front and back images capture the pipe (see Fig. 19). Therefore the back images may still be usable for the dataset. Secondly, most back images start to lose vision at the 13th last back frame by showing a full blue (0,0,255) image, this can be seen in Appendix C. Subsequently, almost all inspection back images are blue at the 11th last frame. Nevertheless, it was found that the inspection with videoid bd6509746d4845a29deb2d46afcea884 contains blue front images between the 71st and 75th image index. Lastly, smaller pipes can be erroneous by showing obstacles or the ending manhole shaft as can be seen in Fig. 20. In particular, the shortest pipe inspection contains only full-black images. In conclusion, fig … shows the process of filtering out erroneous images.

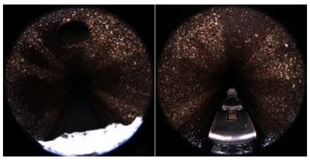


*Fig. 19: Loss of vision for the front image in one specific inspection.*

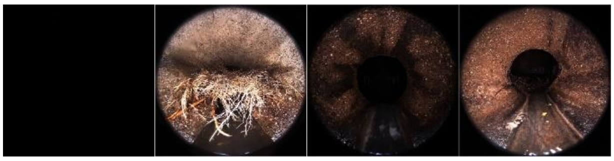


*Fig. 20: Plotting the 6th image of the four shortest inspection image sizes of a) 8, b) 26, c) 46, d) 54.*

### 4.4.4. Front- and Back Images

The dataset size is significantly influenced by the decision to include only front images rather than both front and back images, primarily due to the higher likelihood of encountering annotated observations in front images. In particular, as observations could overlap over both front- and back directions, often observations are just visible enough in the front images, for instance, circular observations such as the displaced joints (BAJ) appear often just in at the circular border of the front image. Moreover, including both front and back images may pose a risk of overfitting since almost all similar content of a pipe is duplicated. However, including both camera directions could provide a complete view of the inspected area by providing an unseen opposite view of the pipe, reducing the risk of missing observations only from a specific direction. Additionally, this dual approach mitigates the loss of visualizing the observation due to poor image quality from one view, as the other frame may still show the observation. Furthermore, the robot’s chain presence in each back image is unlikely to significantly affect the classification, as it is a factor that is constant in all back images. Nevertheless, including both front and back images may pose a risk of overfitting, although back images provide an unseen opposite view of the pipe. To conclude, alternative methods for both front- and back-inclusion could involve using edge wrap-around or mirror padding to see beyond the borders, merging both images into one entire image with for instance unwrapping, or offsetting the annotation location to ensure full observations are captured.

### 4.4.5. Final Filtering Chain

The evolution of the construction of the final filtered dataset by applying the data preparation filers in Fig. 21 shows that the final dataset consists of 126055 images of which 2958 observations are included (see Fig. 22). From the distribution of Fig. 22 is finally chosen to include the BCA, BAJ, BBF, BAB and BAF due to their proficient amount for resampling and their mainly non-continuity. The initial code count in Fig. 22 displays a value of 3192 observations since for the filtering only the 4035 non-continuous observations are implemented in the final constructed dataset in the beginning stages which consists of 278020 images (see Fig. 1). Additionally, different notations can be made to mention the most extreme observations. Throughout the filtering chain, neglecting the start- and end-frames caused a lot of change. Nevertheless, the number of observations did not change significantly, allowing the proportion of empty images to already have been lowered and decreasing the imbalance.

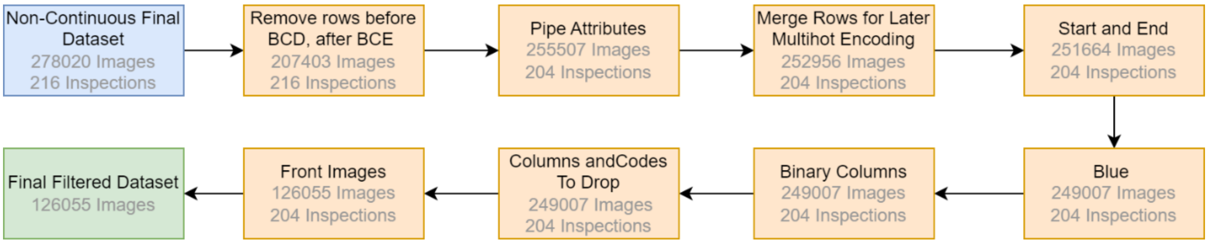


*Fig. 21: Filtering chain by applying all the data preparation methods to end up at the final dataset.*

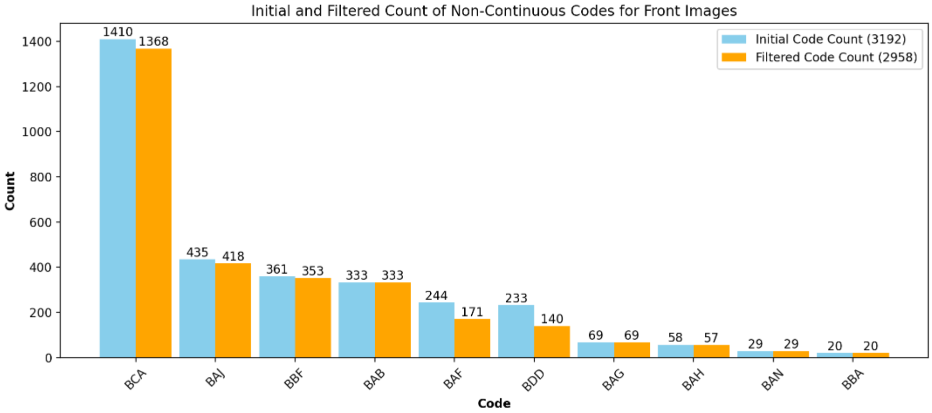


*Fig. 22: Observation count before and after applying the filter chain.*

### 4.4.6. Incorrect Temporal Labeling Location

The lack of labelling visible observations of surrounding images around a single registration annotation is a major factor impacting the chance of detecting the observation. Particularly, this occurs since inspectors only annotate the location when the observation occurs closest to them in the 360-degree image space. The example images in Fig. 23 are used for the following assumptions to argue the issue of temporal mislabeling. First, the multiple accruing observations over a certain range in the pipe, these single-occurring observations are not treated as such for all inspections. Besides, in some situations, the observation overlaps over the front and back images, which does not allow for full vision of the observation, while the other surrounding unlabeled images do. Lastly, due to the multi-label occurrence of multiple observations, two defects may occur in the same image as seen from a distance, however are not labelled. In conclusion, for multi-label image classification, this could cause major issues, for instance, multiple images showing a defect while only one of the 10 images showing that effect is classified as the defect. Therefore, the chance is increased that the classifier classifies the image as a frame not containing an observation, which causes the false negatives to increase. In conclusion, to prevent missing potential observation features by mitigating the incorrect location of observation labelling, two methods were assumed to include images around the annotation location.

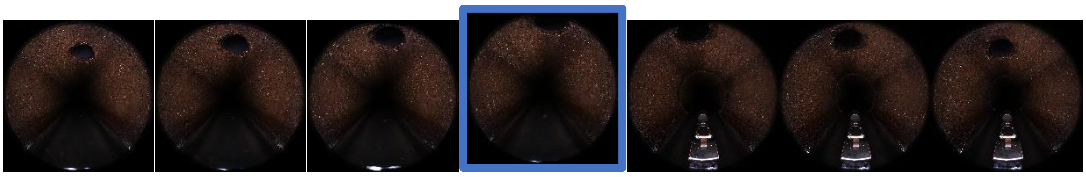


*Fig. 23: Three front images, followed by one labeled front image in the middle, and three unlabeled back images, all of the same defect.*

#### 4.4.6.1. Offsetting

A first method could involve shifting the annotation locations of labeled images to increase the chance of capturing the full observation. In particular, for each frame direction, the annotations are offset backward for the front images and forwards if the back images are included in the dataset. The optimal offset value could be determined during the modeling phase. Nevertheless, this method could lead to model predictions drifting away from the true location of features of unseen data during localization in the client's software system during future deployment. Furthermore, unlabeled surrounding images displaying the same observation as the offset image should be excluded to ensure the model learns that the feature in the offset image contains the observation to be learned.

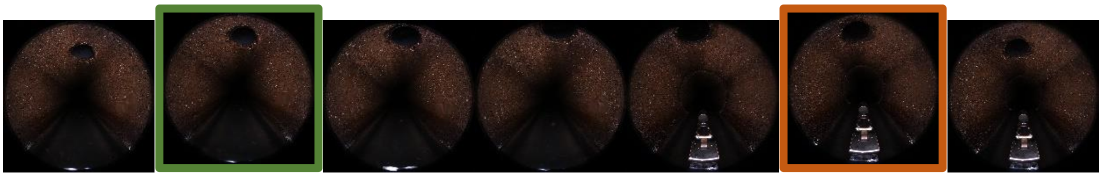


*Fig. 24: The offset annotation location method is explained where the front image (green) is offseted to frames earlier, and the back image (orange) is offset two frames later, both to capture the full observation.*

#### 4.4.6.2. Propagation

Secondly, a propagation method could be implemented to include all images between the offset-annotated frame and the initial annotated frame to increase the chance of capturing a mislocated observation. This method could be proficient in preventing images with distant observations from being mislabeled as normal, as illustrated in Fig. 25. This problem is evident in Fig. 23, showing one image identified as an observation while the other frames are labeled as normal, resulting in a low probability (1 in 7) of detecting inlets and presumably shifting model bias towards predicting normal images. Furthermore, propagation could be similar to data augmentation which increases dataset diversity by applying image transformations. In this case, propagation seems to mimic the scaling augmentation operation. However, propagation might resemble oversampling, as nearly identical images (taken 5 cm apart) are repeatedly included, potentially causing model overfitting. Moreover, to evaluate the effect and optimal propagation distance, various experiments can be conducted during the modeling phase. Alternatively, a reverse propagation method could be experimented with, excluding frames around the initial or offset-annotated frame. However, this should be carefully conditioned to avoid excluding frames with closely occurring observations within the deleted propagation range. Therefore, other methods could be considered to include the full observation. For instance, of all the propagated images around the initial labeled image, only the offset image should be included, and the rest excluded from the dataset.

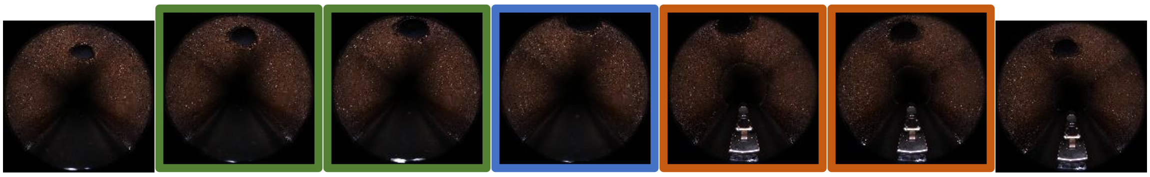


*Fig. 25: The propagation method is explained including propagated front (green) and the back images (orange), including the intial frame for both the front and back image.*

#### 4.4.6.3. Unwrapping

Since sewage inspectors annotate in panoramic images, the front and back images can be unwrapped and concatenated to provide similar information to the classifier by capturing all contextual information, including the full observation. This technique is inspired by Künzel et al. [6] who unwrapped fish-eye images to enhance a segmentation model's robustness by showing promising results for detecting varying defects. For this project, an unwrapping prototype in Fig. 27 shows that the full observation can be captured. Nonetheless, unwrapping faces challenges as mentioned by the literature and for this project. Künzel et al. [6] addresses the invisibility of obstacles due to the disappearing background in stitched unwrapped images. For this project, manual adjustments to both image centers are performed to align pixels at the seam, and distortion throughout the unwrapped image persists. Hence, Künzel et al. [6] utilized local and global pose estimation correcting camera position deviations and reducing distortions in unwrapped images.

Furthermore, this project considers two other methods to prevent including distorted images with intrusive seams. First, the earlier-mentioned annotation offset method could be implemented to either the front- or back image to create a 5 cm overlap where the stitching function has a higher chance of finding similar features to create a coherent image with a minimally distorted seam (see Fig. 28). In particular, the stitching operation is grounded by a scale-invariant feature transform (SIFT) to concatenate images based on corresponding local features. A SIFT experiment is performed in Fig. 29 which shows the initial images of Fig. 28, wherein Fig. 29 the straight long lines should find corresponding features in the overlapping regions of the two images. Nevertheless, no features are detected around the inlet, likely due to the opposite capturing direction and unoptimized SIFT parameters. Secondly, overview images (see Fig. 26), consisting of several concatenated unwrapped front frames, could be split and utilized. However, qualitative manual inspection reveals that observations are rarely displayed (see Fig. 26). In conclusion, the prototyped stitch method for unwrapped images remains broken and frames around the annotated image with the corresponding observations are still included. s. Given the importance of maintaining temporal consistency throughout the pipeline [4], alternative methods to capture the full temporal context of an annotated image should be explored, such as temporal classifiers or point cloud-based classification.

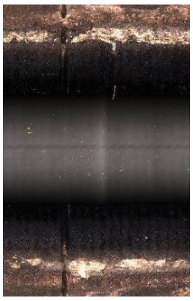


Fig. 26: Overview image consisting of 20 concatenated front images.

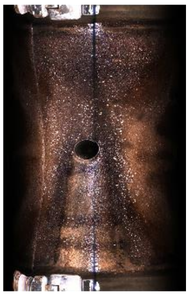


Fig. 27: Unwrapped and concatenated front- and back image.

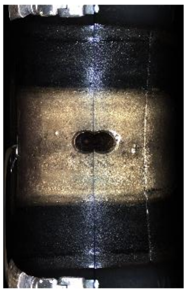


Fig. 28: Unwrapped front- and offset back image.

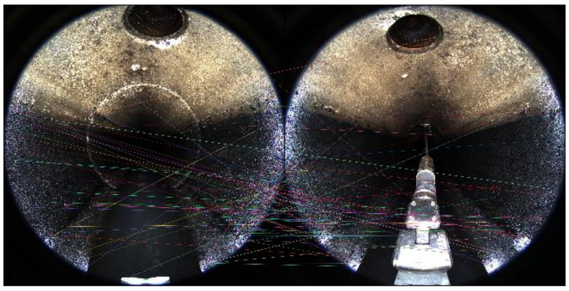


Fig. 29: SIFT feature mapping for a front- and offset-back image a backbone for stitching.

## 4.5. Data Distributions

Random resampling is utilized to balance the data distribution of the training data by oversampling the minority classes and downsampling the majority observations of the empty images. The resampling factors are applied by manual trial and error of integer values by using the same approach as Meijer et al. [9]. Eventually, the training data distributions are balanced as advised by Haurum and Moeslund [2] to equalize the distributions of the observations and maintain an equal distribution between the total defect images and the empty images. Nonetheless, this balance may not be completely mimicked with the resampling of multi-label annotation data due to the co-occurrence of observations in the same frame. In particular, upsampling a minority class towards a certain number of the largest minority class could be prone to either include doubles, (so also increasing the distributions of other classes) or use frames with single occurring anomalies to specifically upsample until a specified number. Hence, it could be considered to not include frames with co-occurring frames, which may decrease the performance of the model in being a multi-label classifier since it may shift into being a multi-class classifier. However, increasing frames with single occurring observations could strengthen the specific features for that observation such that the model can distinguish the observation classes with more ease. In conclusion, different resampling factors are experimented with to try to find an equal balance between all observations, with the risk that the observation types may increase to higher numbers as seen in Fig. 30.

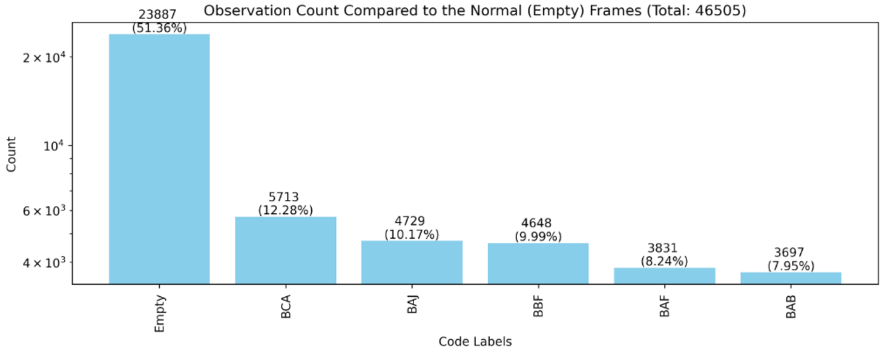


Fig. 30: Resampled observation distribution with varying higher integer resampling factors per observation.

## 4.6. Data Formatting

The final filtered dataset can be integrated into the modelling framework by preparing a custom dataset with constructed tensor input labels to facilitate efficient data loading into the model. Multiple frames are labelled with each a distinct label, causing duplicates of images. Therefore the rows with the image are combined to create a label with a list containing values of the categorical code attribute, which may be empty or filled. To generate tensor labels, binary columns are constructed in the dataset for each code. After data preparation finalizations, the binary values for all codes in a row are converted into a binary vector.

These binary vectors allow filtering out unwanted code columns. However, three methods could be considered while functionally doing this. First, code columns could be dropped from the data frame without further modifications. However, the excluded frames are still present in the dataset as normal images, which could be confusing for the model. Therefore, a second method could drop the columns and also drop the rows where the left-over codes are not present to exclude the present features of other codes. Nevertheless, still surrounding the annotated frame, the code will still appear due to the temporal labelling problem. Another option could be to merge the residual codes into an "others" code, to be able to be detected by the classifier as an observation with rare codes. Although, this observation code may be due to the variety of different code features in this code. For this project, the residual codes are summed together into another class to be able to still let the classifier learn the distinctions between other codes, compared to rare codes.

Consequently, this vector is transformed into a tensor vector in the dataset class, serving as the annotated data for each frame as neural network input. The dataset class is derived from the PyTorch module to create a custom dataset for loading input data into deep learning models. In this project, the sewage dataset class loads the binary vector labels and corresponding image paths from a specified image directory. Additionally, transformations can be applied, and the maximum data loading size can be adjusted to improve model development efficiency by using a subset of the total data. Lastly, for each item, the class module retrieves the image from the directory and its corresponding tensor label, which are then provided to the data loader to assemble the training and testing sets.

# 5. Modelling

The prepared data is utilised to find the most optimal model during an iterative model development phase. This involves experimenting with various pre-processing techniques and models to assess which model gains the optimum recall and F2-score performance. However, first, the model environment must be initialised by selecting a modelling technique, splitting the data, determining proper evaluation metrics, and building the training structure with certain hyperparameter tools and values.

## 5.1. Modelling Technique

At first, three state-of-the-art deep learning networks were selected to be used during the experiments. Additionally, modelling factors were established that allow for informed and focused further model adjustment decisions by supporting the interpretation and clarification of the results.

### 5.1.1. Modelling Selection

By referring back to the state-of-the-art different potential models for multi-label classification, one convolutional and two residual neural networks are selected to assess the different classification capabilities between different network types. From the state-of-the-art models used for sewage pipe anomaly classification, Haurum and Moeslund [5] reasoned their choice of two publicly available pre-trained models to construct a two-step hierarchical network. Particularly, their benchmark on F1-score concluded that the AlexNet network by Xie et al. [7] was optimal for the classification of anomalies without a specific classification between observations, but rather if an anomaly occurs. For the second step in their hierarchical framework, their benchmark advised to utilize a TresNetL, as was stated that ResNets perform well for the classification between the choice of different available observations. Nevertheless, the F1-score may be less relevant for this project, however, this metric comes closest to the F2-score, since Haurum and Moeslund [5] used the F2-score, although with used class importance weights which may make a significant difference in performance. In conclusion, it was chosen to start with the implementation of the pre-trained AlexNet with fine-tuned weights of Xie et al. [7] and consider the pre-trained AlexNet on default weights in later stages. Additionally, various ResNet networks such as ResNet50 could be implemented to assess the leading performance especially observation distinction by ResNet models.

The AlexNet model is implemented in three distinct manners to compare its performance based on its pre-training data and the number of fine-tuned layers. First, the last fully connected layer of the AlexNet was pre-trained on sewage images by Xie et al. [7] is fine-tuned, as it likely captures domain-specific features better than models pre-trained solely on generic datasets, for instance, ImageNet, and could potentially reduce training time during fine-tuning. However, the pre-training data of Xie et al. [7] may not fully represent the variety of sewage defects for this project's training data, causing the model to miss certain observations. Therefore, fine-tuning the last layer of a default pre-trained AlexNet leading from PyTorch on project-specific images is considered. This method is used for most of the experiments to be able to assess the diversity of applying pre-processing techniques while keeping the model type a constant factor. Nonetheless, fine-tuning only the last layer may underfit the model to the complex features of sewage images. Hence, another experiment assesses the model's performance when two of the last AlexNet layers are fine-tuned, allowing the model to capture specific features of the sewage images more effectively. However, increasing the number of learnable parameters can lead to overfitting, as the model might learn features which are too specific to the training samples, reducing its ability to generalise to new data. It is often more effective to start with fine-tuning fewer layers and expand only if the results, together with the validation performance, show no overfitting.

Haurum and Moeslund [5] note from their model benchmark that the AlexNet of Xie et al. [7] is a proficient network for classifying normal and defective images, but its fewer layers may not capture complex image features. In contrast, Residual Neural Networks (ResNets) are deeper, with more layers, achieving dominant state-of-the-art results in various image classification tasks. In addition, Haurum and Moelsund [5] found from their benchmark that ResNets can be utilised as the second stage in a hierarchical network for classification, as it can distinguish features between observations. Nevertheless, deeper ResNets, such as ResNet50, require increased training times and computational resources compared to shallower networks like ResNet18. However, while ResNet18 may require less training time and still benefit from residual connections, it may not capture as complex features as ResNet50, potentially missing distinctive details of observations. Therefore, for both the ResNet50 and ResNet18, distinct experiments are performed to assess the model's performance and computational efforts and determine their potential to replace AlexNet as the prior model during most experiments.

### 5.1.2. Modelling Factors

Six modelling factor assumptions are considered that could be taken into account while analysing the models' results to make well-informed decisions for further model adjustments.

- *Data Quality*: The dataset is assumed to be clean from poor lighting conditions, noisy or erroneous images and camera failures. Therefore, the images are expected to have uniform and consistent average brightness and tubular appearance. Still, erroneous images may occur due to the presence of manhole shaft images, as the number of initial and final images outside the pipe shape can differ in each inspection.
- *Observation Dependence*: Certain observations within the same image could be dependent on each other's presence. This is assumed since balancing the observation distributions during resampling was not yet possible because certain observations often occur within the same image as an inlet, for instance, raising both their resampling presence at the same time. However, the model likely treats each observation as an independent observation due to the binary loss function in the multi-label classification method.
- *Varying Observation Appearance*: The appearance of observations throughout all inspections could deviate depending on the type of observation, such as roots, which are considered more variable in their appearance than inlets and displaced joints. Hence, the model's ability to generalise across minority types of observations could be negatively impacted.
- *Deviating Inspection Appearance*: Observation appearance variability is tried to be mitigated by maintaining consistent pipe attributes throughout all inspections, letting the model learn features in observations independent of pipe attributes.
- *Non-Stationarity*: Despite efforts to maintain pipe appearance consistency, the visual appearance of the pipes could deviate due to different inspection date times, causing diverse deteriorations.
- *Class-Imbalance*: Data balancing methods are assumed to allow the model to generalise over well-represented minority observations in the dataset. However, the manual resampling factor for each observation can vary for certain experiments, causing the varying observation proportions. This indicates that the model's performance is influenced not just by changes in preprocessing techniques or modifications to the model itself but also by data balancing shifts throughout the experiments.
- *Same Pipe Inspections*: Given that the dataset includes inspections of the same pipes on different inspection dates, there may be a risk of data leakage if inspections from the same pipes are present in both the training and test sets. Particularly, lightning mitigation transformations aimed to standardise the lightning appearance of the inspections may cause inspections of the same pipes to appear more similar.

## 5.2. Test Design

Next, the test design outlines how to train and test the model's performance across certain subsets of the data and evaluate its performance with a combination of five relevant metrics.

### 5.2.1. Training-Validation-Test Splitting Ratio

The dataset is split by stratified sampling, dividing the dataset inspections into training, validation, and test sets using the most effective ratio determined through experiments, which closely aligns with the state-of-the-art splitting ratios. Splitting the dataset by distinct inspections, rather than randomly dividing images, minimises data leakage and ensures the model is tested on entirely unseen inspections and images, reflecting real deployment conditions. To initialise the experiment's data-splitting ratios, state-of-the-art ratios (that do not employ cross-validation) were reviewed to identify optimal ranges for training, testing, and validation that could be assessed during the modelling experiments. For instance, Li et al. [8] used a 70/30 split for the training and test sets while further splitting the test set equally into a test and validation set. Chen et al. [5] increased the training proportion to 75%, leaving 25% for testing. As these two studies obtained the validation set from the division of the test data, Dang et al. [7] allocated 90% of the dataset to training, which was then divided into a 75% training and 25% validation set. These approaches ensure that the same distribution is present across the training, validation, and test data since the images are randomly split. However, due to the lack of an independent test set, this may lead to overfitting and reduce the model's generalizability. Therefore, during the experiments, the dataset's inspections are first divided into training and test sets by around a 70/30 split, followed by an equal split into validation and test sets, allowing balancing techniques to be applied to training data.

To ensure the model is validated and tested on unseen pipes, stratified sampling is used during data splitting to ensure an equal representation of the initial inspection length distribution across the training, validation, and test sets. This is particularly useful since this project deals with an imbalanced dataset where inspections deviate in length, likely causing certain observations to be underrepresented in shorter inspections. Utilising stratified sampling can prevent the validation and test sets from containing only the longest or shortest inspections, thereby enhancing the generalizability and consistency of the performance results to inspections with other lengths. Additionally, Fig. 31 shows t shows that the training and test sets have a similar inspection length distribution as in Fig. 6. In contrast, Fig. 31 shows that random splitting results mainly in inspection lengths around 610 meters in the training set, while the test set contains increased alternating frequency differences. Nevertheless, for stratified sampling, inspections of the same pipe likely have similar lengths, they tend to appear in both sets, risking data leakage. Therefore, leave-one-out cross-validation could ensure that the training and test sets are independent, evaluating the model on truly unseen data and allowing training on even the rarest observations. In conclusion, stratified sampling is utilised, since the minority observations in the training could be upsampled, allowing for enough data for rare observations.

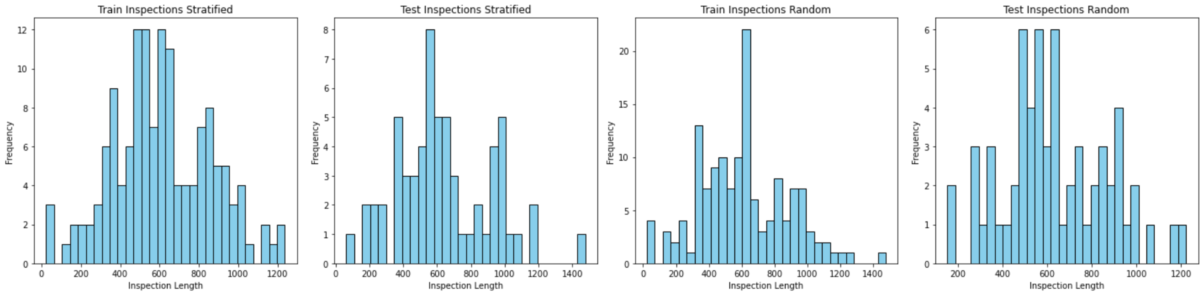


*Fig. 31: Inspection length histogram differences by using stratified sampling compared to random sampling.*

### 5.2.2. Evaluation Metrics

The model's performance is evaluated from multiple perspectives using four selected metrics which account for the imbalance in the dataset. First, recall is a crucial metric in anomaly classification problems [9] [8] as it measures the proportion of true occurrences correctly identified versus those missed (1). Consequently, missing observations can have severe consequences for the predictive maintenance of sewage infrastructure, leading to high repair costs. Furthermore, This metric is particularly relevant for imbalanced data as it focuses on the classification performance of actual observations, excluding the majority class ("empty/normal images"). Recall is integrated into the deep learning framework by utilising the MultilabelRecall function from the torchmetrics.classification module and specifying how many classes are predicted, so the length of the tensor vector, which is, in this case, five, since five observations could be predicted. Hereafter the total tensor predictions are obtained with a probability threshold of 0.5 after training, validation, and testing. Subsequently, the recall is provided for both the total and each observation across these phases.

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

Secondly, the F2 score is selected since missing an observation (FN) is often costlier than an FP, although minimising false positives remains important by maintaining a proficient precision value. In other words, recall is prioritised over precision, but false positives must still be minimised. Mathematically, the Fβ-score is a weighted harmonic mean of precision and recall (2), placing greater emphasis on recall when β > 1. In this project, the β value is set to 2, which weights recall four times more than precision (3). Besides, the F2 score is implemented similarly to the recall value with the MultilabelFBetaScore function.

$$F_\beta = (1 + \beta^2)\frac{Precision * Recall}{\beta^2 * Precision + Recall} \tag{2}$$

$$F_2 = 5\frac{Precision * Recall}{4 * Precision + Recall} \tag{3}$$

Next, the precision-recall (PR) curve is used to select the optimal threshold for predicting values from probability outputs during training and evaluation by plotting precision against recall for different threshold values. Different PR curves can be plotted simultaneously for each observation to determine if separate thresholds are necessary per observation. Besides, the area under the PR curve (AUC-PR) is useful to assess model performance across all thresholds per observation. Additionally, the PR curve emphasises the model performance on the positive class, which is often the minority class in imbalanced datasets and is, therefore, less sensitive to class imbalance than, for instance, the ROC curve. This metric is integrated into the framework using the MultilabelPrecisionRecallCurve function from the torchmetrics.classification module. Afterwards, by generating PR curves for both the training and test (Fig. 32) sets, it could become evident to which observations the trained model overfits.

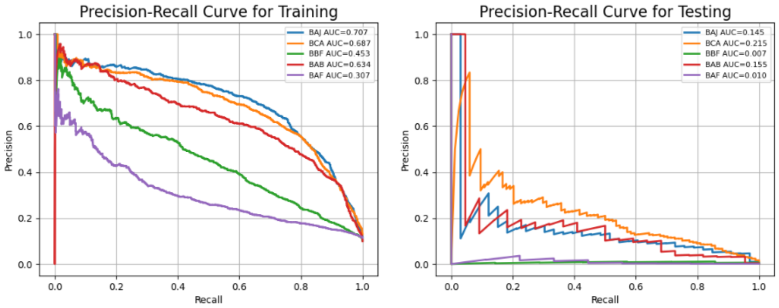


*Fig. 32: Precision-recall curves for having a balanced training set and an imbalanced test set.*

Lastly, the conditional metric of precision at 0.9 recall is used to evaluate the model's ability to identify 90% of actual observations while minimising false positives. Among recall levels of 0.9, 0.95, and 0.99 [9], 0.9 is chosen to balance high recall against a proficient precision performance. This metric is derived from the precision-recall curve by extracting the precision value at a 0.9 recall, indicating a unique threshold for turning probabilities into predictions.

## 5.3. Build Model

The test design and dataset can be integrated into the classification network by incorporating them into custom PyTroch training and evaluation functions with initial hyperparameter settings. The model is set up in five steps by initializing the model and using specific loss, optimizer and learning rate schedular and providing the values of additional hyperparameters. First, the default AlexNet, ResNet50 and RsNet18 can be loaded in with pre-trained weights on ImageNet from the torchvision.models module from Pytorch and the AlexNet pre-trained on sewage images by Xie et al. [7] can be loaded by coding their AlexNet model structure and loading in the corresponding weights as a PTH file. Hereafter, the layers' parameters that require gradients are specified to perform fine-tuning of the last layers. Secondly, a Binary Cross Entropy with Logits Loss function is utilized to treat each observation independently in the multi-labelled data by applying a sigmoid activation and computing binary cross-entropy loss on those probabilities. Especially, this loss function allows cost-sensitive learning by weighting certain classes to penalize misclassifications of infrequent observations. Thirdly, the adamW optimization algorithm is chosen to decouple the weight decay from the learning rate to adjust the learning rate without affecting the weight decay, allowing stable and controllable optimization. This prevents overfitting and weight instability by adding a small penalty to the loss function based on weight sizes. Next, a StepLR learning rate scheduler is used to gradually reduce the learning rate to allow the optimizer to take smaller steps towards a deeper local minimum, enabling more precise fine-tuning of the model weights. Lastly, additional hyperparameter settings are set with a learning rate of 0,001, a batch size of 32, 3 epochs, and an input image size of 224.

## 5.4. Model Assessment

To assess the performance of the built model with varying adjustments, 26 experiments were performed in which pre-processing methods and neural network architectures were interchanged to consider the optimum model modifications. In particular, the experiments were grouped into six topics to consider which settings for the data split ratio, lightning condition methods, temporal mislabeling mitigations, modified loss functions, hyperparameter optimisations, and convolutional neural networks were proficient in raising mainly the model’s recall and F2-score performance. Additionally, the dataset size and resampling multiplication factors were modified throughout the experiments to discover the optimally balanced training set. Finally, the average performance of the recall, F2-score, and the average conditional metrics of the precision at 0.9 over all observations of the model over the validation and test set were considered to evaluate the model results.

A metric visualisation is constructed to clarify the model's performance diviations and facilitate the comparison of the intercorrelated experimental results from diverse evaluation metrics with ease. Fig. 33 illustrates this visualisation, with coloured experiment numbers classified into groups and annotated accordingly by encircled, highlighting the most prominent or extreme results. Generally, the test set shows lower recall and F2-score values compared to the validation set. Consequently, the metric values for each metric belonging to the test and validation set are averaged, as the performances of these sets show similar behaviour compared to other metrics. Additionally, some experiments lack data points due to early stopping due to significant performance reductions compared to the previous experiment.

Furthermore, the overarching trend of different behaviours trouhgout the visualisation in the test and validation set performance is first discussed before examining detailed comparisons within experimental groups. Especially, the recall and F2-score metrics for the validation and test sets deviate significantly at the beginning and end of experiments compared to those in the middle. This may be clarified due to the model being optimised based on validation set performance, potentially fitting the validation data better than the unseen test data. Alternatively, due to the stratified data split, the validation set might contain slightly easier or more representative samples to learn. Eventually, when the test and validation sets show similar behaviour, it may indicate that the model generalises well to unseen data and that the current combination of the balanced training data distribution and the stratified split of the validation and test sets are optimal. This major change likely corresponds to changes in the resampling factor at the beginning, middle, and end of the experiments, leading to different numbers of upsampled minority classes. Thus, the resampling factors and data balance between 9 and 14 may be more optimal, containing representative observations in both the test and validation sets.

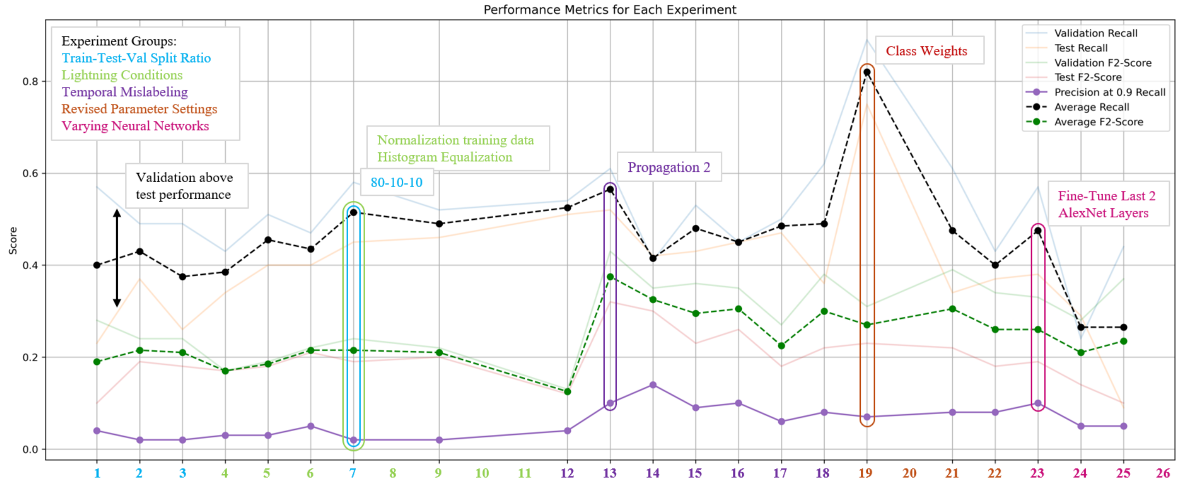


*Fig. 33: Visualisation of the evaluation metric results for the 26 experiments of the model.*

### 5.4.1. Train-Test-Validation Split Ratio

The first experiment group assesses the optimal ratio for splitting the inspection data into training, validation, and test sets based on model performance. The results of Fig. 33 are evaluated using data from experiments 1, 2, 3, and 7 from Table 7. Experiments 1 to 3 reveal that an 85%-7.5%-7.5% ratio yields maximum performance for average recall and F2-score, despite lower average precision at 0.9 recall. Nevertheless, this ratio leads to larger validation and test sizes and a decreasing training set. This phenomenon can be explained by the fact that in the initial training set of 108332 images, most normal images are downsampled with upsampled minority classes, resulting in a total balanced training set of 12016 images. This occurs consistently across all experiments, indicating issues with data splitting as the final sizes of the training, validation, and test sets no longer maintain the initial ratio.

Therefore, for the experiments after experiment 3, a 90%-5%-5% ratio was chosen to reduce validation and test sizes from 9509 to 6023. However, smaller validation and test sets can introduce less representative observations, impacting the reliability of performance metrics. This was evident as BBF and BAF observations often had a recall or F2-score of 0 in the test and validation sets, likely due to underrepresentation during the data split. Therefore, experiment 7 used an 80%-10%-10% ratio with larger validation and test ratios, achieving increased recall performance while maintaining similar F2-score levels as experiments 1-3. Notably, the validation and test set sizes exceeded the training set size, which can be acceptable if the ratio of split inspections is maintained before resampling, ensuring that the observations are similarly split over the training, validation and test set with a specified ratio.

| Exp. Group | Exp. | Training, Val, Test Ratio | Train, Val, Test Size | Resampled Training Data Proportion: Empty, BAJ, BAF, BCA, BBF, BAB | Early Stop |
|---|---|---|---|---|---|
| Ratio | 1 | 95% 2.5% 2.5% | 13083, 1628, 1606 | 6688, 1446, 1350, 1167, 786, 587 | |
| | 2 | 85% 7.5% 7.5% | 12016, 9509, 9801 | 6688, 1446, 1350, 1167, 786, 587 | |
| | 3 | 90% 5% 5% ↓ | 12443, 6023, 6740 | 6688, 1446, 1350, 1167, 786, 587 | |
| Lightning | 4 | Normalization Dataloader | 12443, 6023, 6740 | 6688, 1446, 1350, 1167, 786, 587 | |
| | 5 | Normalization Dataloader, Histogram Equalization | 12443, 6023, 6740 | 6688, 1446, 1350, 1167, 786, 587 | |
| | 6 | Normalization Dataset, Histogram Equalization↓ | 12443, 6023, 6740 | 6688, 1446, 1350, 1167, 786, 587 | |
| Ratio | 7 | 80% 10% 10%↓ | 11578, 12650, 13219 | 6688, 1446, 1350, 1167, 786, 587 | |
| Lightning | 8 | CLAHE X | 11578, 12650, 13219 | 6688, 1446, 1350, 1167, 786, 587 | yes |
| | 9 | Gamma Correction 1.5 X | 11578, 12650, 13219 | 6688, 1446, 1350, 1167, 786, 587 | |

| | | | | | |
|---|---|---|---|---|---|
| | 10 | Gamma Correction 1.1<br>White Correction<br>Shadow Remover X | 11578, 12650, 13219 | 6688, 1446, 1350, 1167, 786, 587 | yes |
| | 11 | Gamma Correction 1.1 X<br>White Correction X | 11578, 12650, 13219 | 6688, 1446, 1350, 1167, 786, 587 | yes |

*Table 7: Experimental setup for experiments 1-11. Green arrows indicate the setting is used for further experiments. Red crosses indicate the setting is not pursued.*

### 5.4.2. Lightning Conditions

The impact of various lighting condition methods to mitigate poor illumination and mean brightness deviations showed that normalization and histogram equalization optimally improved the model's performance. The results from Fig. 33 related to this experimental group is evaluated by first discussing the results from experiments 4 through 6 involving normalizations and histogram equalization, followed by additional considered augmentation methods in experiments 8 through 11 as seen in Table 7.

#### 5.4.2.1. Normalization and Global Histogram Equalization

First, the process of evaluating the impacts of normalization and histogram equalization methods is performed. In Experiment 4, normalization with values based on the standard distribution and variance over the data loader showed moderate improvement in recall and precision at 0.9 recall. Subsequently, it was considered that normalization alone may not be sufficient for significant image enhancement. Therefore, Experiment 5 incorporated histogram equalization to address the brightness disparity between inspections from 2009 and those from 2020 onwards. This approach increased the recall and F2-score, suggesting that combining normalization with histogram equalization enhances model performance by reducing extreme contrasts and increasing observation distinguishability. However, in Experiment 4, normalization was applied to already normalized images from the data loader, which may be incorrect. Hence, Experiment 6 changed this by recalculating the standard deviation and variance over the training data's pixel values and then normalized the unnormalized dataset. This adjustment decreased recall but increased the F2-score and precision at 0.9 recall. Therefore, the choice between applying normalization on the training set or the data loader can be considered. For subsequent experiments is chosen to apply normalization over the training set with newly calculated standard deviation and variance values of the new training dataset. This could ensure identical standardized pixel value distributions for future expansions of the training data.

#### 5.4.2.2. Gamma Correction, White Balance Correction and Shadow Removal

Secondly, during experiments 8 through 11 was found that replacing the histogram equalization and utilizing three additional data augmentation methods was not proficient. Notably, the data splitting ratio for these experiments was adjusted to 80% training, 10% validation, and 10% testing, differing from previous lighting condition experiments. First, the outcomes for Experiments 8, 10, and 11 are considered which were absent from Fig. 33 since early stopping was applied when the validation performance after certain epochs drastically declined compared to earlier experiments. Experiment 8 implemented CLAHE to consider whether local contrast enhancement would outperform global histogram equalization in mitigating lighting conditions for finer image details. However, due to early stopping, it was decided to retain global histogram equalization. Besides, Experiments 10 and 11 respectively assessed the impact of shadow removal and white balance correction to remove lighting artefacts that can obscure defects, although, both were similarly excluded due to early stopping by showing a lower validation performance. Nevertheless, Experiment 9 applied a gamma correction of 1.5 to brighten images and enhance defect visibility without overexposure, maintaining the F2-score and precision at 0.9 recall score as Experiment 8, but was ultimately dropped due to decreased recall. In conclusion, global histogram equalization followed by normalization over the training data was the most effective approach to mitigate lighting conditions and enhance the model's learning capability.

### 5.4.3. Temporal Missed Annotations

From conducting seven experiments to asses whether offset and propagation methods could mitigate the temporal mislabeling problem, a propagation of 2 yielded optimal model performance. Before evaluating the experimental results, assumptions, observations and clarifications are made to interpret the result correctly. Throughout these experiments, a splitting ratio of 80% 10% 10% is used with global histogram equalization and training set normalization applied consistently. However, as shown in Fig. 34 and Table 8, varying upsampling and downsampling factors were used for data balancing and the propagation methods increased the datasets, causing experiments to deviate in their final dataset for modelling. Therefore, the maximum size of the dataset loaded into the dataset class was reduced by a factor of 0.3 for Experiment 13 onwards to maintain consistent training, validation, and test sizes. Moreover, Fig. 34 shows a consistent offset in the BCA sample size above the other observation sample sizes. This phenomenon may result from the multi-label nature of the data, where BCA cannot be downsampled below a certain threshold because it co-occurs with other observations in certain images. Furthermore, the varying training set sizes across the experiments likely suggest that different numbers of observations will be present, causing the experiments with higher training sets to be trained on more observation instances. Lastly, Fig. 33 shows that the precision at 0.9 recall plot behaves similarly with the varying training set sizes as in Fig. 34.

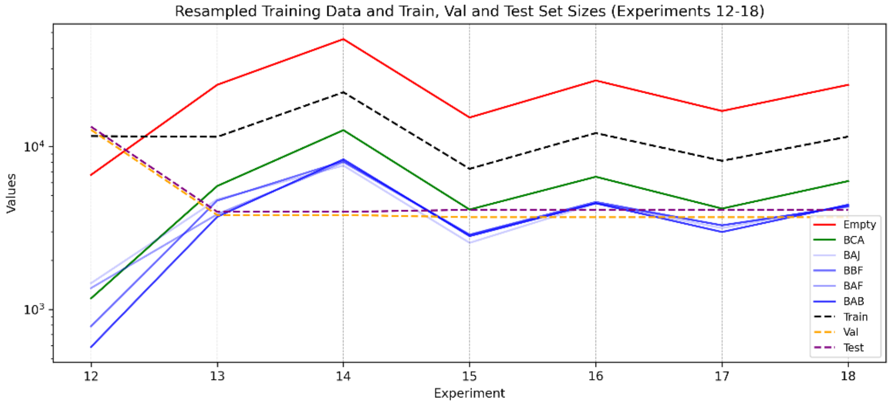


*Fig. 34: Sizes of the observation types in the resampled training data, combined with the sizes of the final training, validation and test sets.*

| Exp. Group | Exp. | Training, Val, Test Ratio | Train, Val, Test Size | Resampled Training Data Proportion: Empty, BCA, BAJ, BBF, BAF, BAB | Early Stop |
|---|---|---|---|---|---|
| Temporal | 12 | Offset 2 | 11578, 12650, 13219 | 6688, 1167, 1446, 786, 1350, 587 | |
| | 13 | Propagation 2<br>SizeFactor = 0.3 ↓ | 11439, 3795, 3965 | 23887,5713, 4729, 4648, 3831, 3697 | |
| | 14 | Propagation 5 | 21504, 3795, 3965 | 45583, 12586, 7631, 8119, 8007, 8331 | |
| | 15 | Propagation 3 | 7286, 3677, 4083 | 15050, 4097, 2559, 2873, 2859, 2814 | |
| | 16 | Propagation 3 | 12078, 3677, 4083 | 25397, 6514, 4457, 4564, 4488, 4478 | |
| | 17 | Propagation 1 | 8160, 3677, 4083 | 16496, 4153, 3153, 3274, 3283, 2983 | |
| | 18 | Propagation 2↓ | 11489, 3677, 4083 | 23887, 6128, 4348, 4291, 4266, 4387 | |

*Table 8: Experimental setup for experiments 12-18. Green arrows indicate the setting is used for further experiments.*

Next, the execution of the seven experiments is explained to gradually conclude why a propagation of 2 seems the most effective method for mitigating temporal missed annotations by achieving an optimal recall and F2-score. First, Experiment 12 offset all labels to the 2nd previous image, resulting in higher recall values but a reduced F2 score. Other offset values did not improve the metrics' performance. In Experiment 13, the propagation method was tested by adding the previous two images relative to the initially labelled image to the training set.  This resulted in increased recall, F2-score, and precision at 0.9 recall, likely due to the greater extent of upsampling, as shown in Fig. 34. To verify if other propagation values would outperform Experiment 13, a propagation of 5 was used in Experiment 14. This reduced recall and F2-score but increased precision at 0.9 recall. In conclusion, the most optimal propagation of two could be chosen which resulted in a recall of 0.51 and an F2-Score of 0.32 for the test set.

To specify the performance of the distinct observations for the three previously discussed experiments, the shape and area under the precision-recall curves for these experiments are assessed. First, Fig. 35 shows that the offset method could be proficient in increasing the precision performance of the BCA, BAB and BAJ. Subsequently, a propagation of 2 reduces the performance of BAB, but increases the performance of BAJ, BBF and BCA. Consequently, higher propagation values solely increase the BCA area under the precision-recall curve which likely caused the model to overfit to connections (BCA) as a greater number of similar images with the connections in sight are used for the training data. The limited ability to correctly identify displaced joints (BAJ) and infiltration (BBF) could be caused by the fact that the BAJ and BBF are less visible from a distance and they could be wrongly classified with a healthy joint. In closing these results could conclude that the offset and diverse propagation settings could be proficient to increase the precision and area under the precision-recall for distinct observations. This likely occurs since certain observations are visible from a further distance, and others not, while certain observations occur in alignment with the pipe and others occur only aligning with the circumference of the pipe.

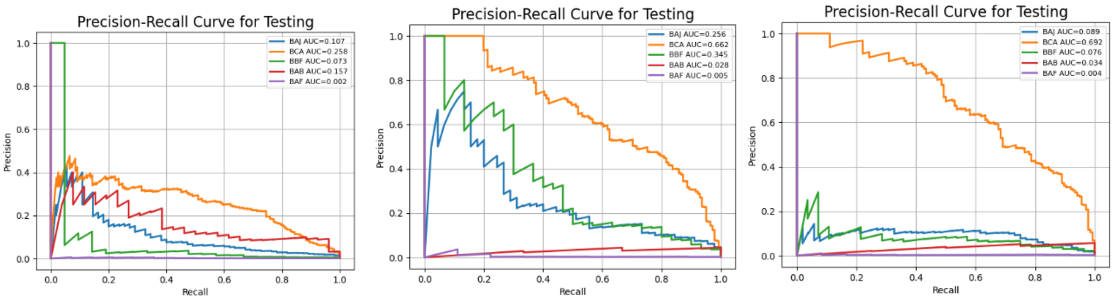


*Fig. 35: Precision-recall curves for an offset of 2 (a), propagation of 2 (a), and propagation of 5 (c).*

Besides, four propagation experiments were conducted to evaluate the model's performance across different propagation values and changed resampled training set distributions. In Experiments 15 and 16, a propagation value of 3 was used, with Experiment 15 having fewer resampled observations than Experiment 16. Fig. 33 shows that Experiment 16 achieved a higher F2-score and precision-at-0.9 recall, but a lower recall value. Therefore Experiment 17 maintained the training balance of Experiment 15 but used a propagation value of 1, resulting in increased recall but significantly reduced F2-score and precision-at-0.9 recall than propagation of 3. Since these three experiments lacked major performance improvements, the propagation value was reverted to 2 with a comparable training dataset balance as Experiment 13. Hence, it was assumed that experiment 18 should perform similarly to experiment 13 again. As the recall of the validation set shows similar behaviour, the test recall value was significantly lower.

The difference between experiments 13 and 18 could be concluded from Fig. 36 as experiment 18 contains a less gradual balance between the observations since the BCA contains more samples compared to the other observations. Therefore, this causes a model to likely perform only well on that solely dominant minority class as Fig. 37 shows that the BCA is the most representable observation in the precision-recall curve of experiment 18. This concludes that maintaining a balance between observations is important to allow other observations to also perform well in the model. Nevertheless, the multi-label nature of the data makes it hard to do this during resampling. Therefore, alternative balancing methods for multi-label classification may be considered. Lastly, the deviated results of experiment 18 may be attributed to factors such as random upsampling since the test set may contain more upsampled observation images with less distinctive features compared to experiment 13.

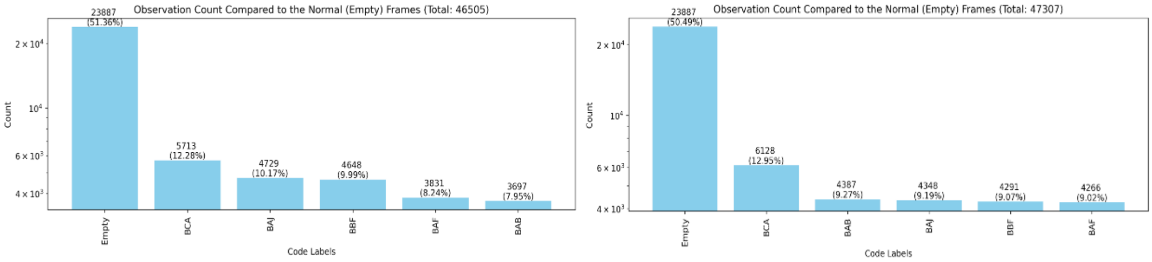


*Fig. 36: Resampled training set distributions in experiment 13 (a) and experiment 18 (b).*

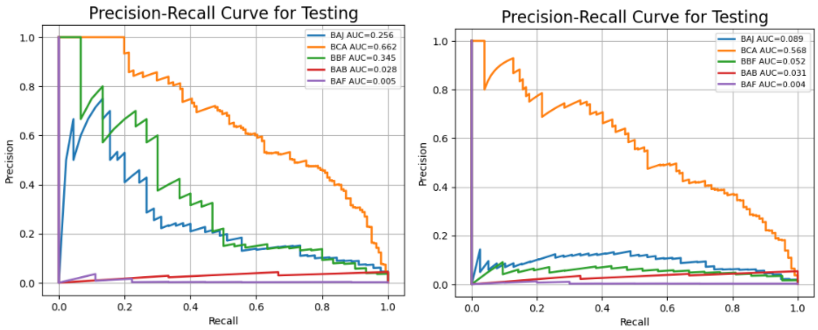


*Fig. 37: Precision-Recall Curve of experiment 13 (a) and experiment 18 (b).*

Lastly, during data preparation was assumed that propagation may cause model overfitting, as larger propagation values could increase the training set with images with visual features similar to the frame on the initial labelling location. Fig. 38 shows that increasing propagation to two does not significantly affect the proximity of the validation loss to the training loss. Even without propagation in (a), the validation loss increased at the 4th epoch while the training loss may approached overfitting. Similarly, with the propagation of two in the second epoch, the validation loss increased while the training loss seemed to decrease. Thus, propagation does not significantly increase the risk of overfitting, as overfitting can occur even without propagation.

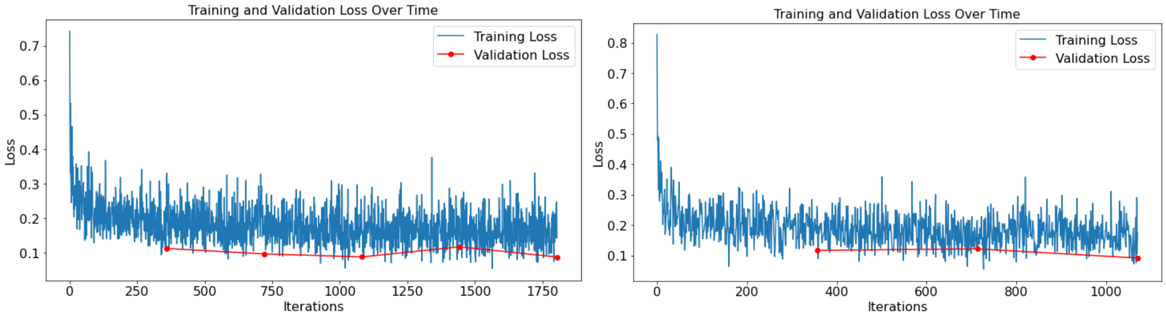


*Fig. 38: Loss plots of the test and validation loss of no propagation in Exp 7 (a) and a propagation of 2 in Exp 13 (b).*

### 5.4.4. Revised Parameter Settings

As previous experiments established the optimal ratio and propagation settings, the model parameters were revised and optimized using hyperparameters and alternative resampling methods through four experiments. Meta-information for each experiment is shown in Table 9. First, in Experiment 19, class weights were applied in the Binary Cross-Entropy Logits Loss function as a cost-sensitive learning method to address the resampling method's difficulty with multi-labeled data. Fig. 33 demonstrates that this experiment yielded the highest extreme values in performance metrics for both validation and test recall, with the test recall reaching 0.75. In this experiment, class weights are applied while the training set is balanced, indicating a combination of resampling and cost-sensitive learning. However, cost-sensitive learning should theoretically improve detection rates for minority classes, while it only benefits the BCA and BAJ classes in the precision-recall curves. Therefore,  further fine-tuning of class weights is necessary to better represent the remaining observation types and ultimately optimize performance since cost-sensitive learning could have more potential than resampling methods.

| Exp. Group | Exp. | Training, Val, Test Ratio | Train, Val, Test Size | Resampled Training Data Proportion: Empty, BCA, BAJ, BBF, BAF, BAB | Early Stop |
|---|---|---|---|---|---|
| Class Weights | 19 | BCE Loss Weights on Balanced Data | 11489, 3677, 4083 | 23887, 6128, 4348, 4291, 4266, 4387 | |
| Loss | 20 | Bootstrapped Cross Entropy Loss | 11489, 3677, 4083 | 23887, 6128, 4348, 4291, 4266, 4387 | yes |
| Hyperparameters | 21 | Learning rate 0.009 | 11489, 3677, 4083 | 23887, 6128, 4348, 4291, 4266, 4387 | |
| | 22 | Batch Size 64 | 11489, 3677, 4083 | 23887, 6128, 4348, 4291, 4266, 4387 | |

*Table 9: Experimental setup for the experiments introducing cost-sensitive learning, loss replacement and optimizing hyperparameters.*

Subsequently, the loss function was replaced by the Bootstrapped Cross Entropy Loss to determine if the model could increase learning the rare observations by focusing on the most challenging examples and preventing the majority class from dominating the learning process. However, early stopping was performed as it was observed that the bootstrapped cross-entropy reached lower metric results than the previous BCE loss usage. This may occurred since the training data was already balanced, likely making the effect of the Bootstrapped Cross Entropy Loss Function less useful. Thus, the Bootstrapped Cross Entropy Loss Function might be more suitable for imbalanced datasets. For the following experiments, the initial BCELogitLoss is used. Lastly, In experiment 21, the learning rate was manually optimized to 0.009. However, this adjustment minimally impacts outcomes compared to the 0.001 learning rate used in experiment 18.  Additionally, experiment 22 increased the batch size to 64 which lowered the recall and F2-Score performance. Therefore, the batch size is reverted to 32, as previous experiment 21 resulted in higher recall and F2 scores.

### 5.4.5. Varying Neural Networks

The performance of a different fine-tuning method and three additional models is assessed to determine their potential compared to the already fine-tuned AlexNet, concluding that the currently used AlexNet provides more proficient performance. The topics of each experiment are described in Table 10. First, Experiment 23 shows from Fig. 34 that fine-tuning the last two layers of a pre-trained AlexNet significantly improves recall, F2-score, and precision at 0.9 recall compared to previous Experiment 22 only. This improvement may suggest that fine-tuning deeper layers helps the model learn specific high-level features from case-specific data. Therefore the number of deeper layers could be continuously fine-tuned until likely a threshold can be reached until the model starts to overfit due to learning too specific features of the training dataset.

Subsequently, in experiments 24 and 25, two ResNet models were fine-tuned and compared to AlexNet to consider other models' potential. It can be found from Fig. 33 that the ResNet50 underperforms on all metrics compared to the previously utilized AlexNet. On the contrary, initially was assumed that ResNet's deeper architecture would perform better during feature extraction. This may be explainable since in this experiment ResNet50 classifies images with anomalies from empty images while simultaneously identifying the types of anomalies. Nonetheless, Haurum and Moeslund […] suggested that ResNets could be more effective for the latter task alone, as they can better distinguish anomaly features from already classified anomaly images. The results of Experiment 24 seem to support this argument, indicating that ResNet50 should be used primarily as a feature extractor between classes rather than for both classification stages. Additionally, ResNet50's lower metric results during the experiment could be because parameters, such as ratios or propagation values, are not yet optimized, unlike those for AlexNet in previous experiments.

Furthermore, experiment 25 yielded extreme results with a fine-tuned ResNet18 as significant deviations exist between the validation and test recall, along with an abnormally high validation F2 score. This could suggest that the ResNet18 performed well on the validation data although poorly on the test set, indicating that the test and validation sets may contain different proportions of similar representable features. Furthermore, since ResNet18 is a shallower network than ResNet50, it should classify images with more complex features less well. Therefore regarding the results, it could be assumed that the observations in the validation set may contain less complex features than the test set. Additionally, ResNet18 required less training time than ResNet50 as it is assumed that the less present residual components demand fewer computational resources. In closing, due to the significant deviation between test and validation recall values, it is unclear if ResNet18 performs better on unseen pipes than AlexNet or the average performance of ResNet50. Nonetheless, an improved version of ResNet18 may still hold potential for future model implementations.

Lastly, experiment 26 implemented a pre-trained AlexNet on sewage pipe images since this network may learned case-specific features already. However, early stopping is performed as the recall and f2-score were 0 for all test set evaluation metrics. Therefore is suggested that the default AlexNet trained on ImageNet only may be more suitable to fine-tune the model to the identical available data. The inefficiency of the network of Xie et al. [7] can be explained by the fact that their robot camera captured images at specific pipe views, differing from this project's constant tubular perspective, likely making their training data unrepresentative for this project. Besides, Xie et al. [7] fine-tuned the entire AlexNet which may have adjusted important lower-level weights to perform well on their data, while our data may needed the initial weights to extract general features.

| Exp. Group | Exp. | Training, Val, Test Ratio | Train, Val, Test Size | Resampled Training Data Proportion: Empty, BCA, BAJ, BBF, BAF, BAB | Early Stop |
|---|---|---|---|---|---|
| Models | 23 | AlexNet, Fine-Tune Last 2 Layers | 11489, 3677, 4083 | 23887, 6128, 4348, 4291, 4266, 4387 | |
| | 24 | ResNet50, Fine-Tune Last Layer | 11489, 3677, 4083 | 23887, 6128, 4348, 4291, 4266, 4387 | |
| | 25 | ResNet18, Fine-Tune Last Layer | 11489, 3677, 4083 | 23887, 6128, 4348, 4291, 4266, 4387 | |
| | 26 | Xie AlexNet, Fine-Tune Last Layer | 11489, 3677, 4083 | 23887, 6128, 4348, 4291, 4266, 4387 | yes |

*Table 10: Experimental setup for the experiments comparing different models and fine-tuning methods.*

# 6. Discussion

## 6.1. Answering the Research Questions

To answer the main research question of how sewage pipe annotation can be automated with image classification to mitigate for inconstant labelling and data imbalance, the modelling results regarding applied data preparation and modelling methods are interpreted by answering the two first subquestions.

The first sub-research question investigates which image pre-processing methods can be utilized to mitigate the data imbalance and enhance image qualities to obtain proficient model performance. First, applying histogram equalisation and normalization to the training dataset proficiently mitigates the brightness deviation between the inspections conducted in 2009 compared to those from 2020 onwards, reaching an average recall value of 0.52 and an average F2-score of 0.22. Secondly, utilizing propagation by including two frames before an initial labelled frame and labelling them as the initial frame can improve the average recall and F2-score to 0.56 and 0.38 respectively. However, during the propagation experiments was found that the resampling balance of each observation matters. Especially since maintaining an average balance with higher resampling factors for each observation could be more proficient than strict balancing where due to the multi-labelled data one observation, in this case BCA, is disproportionately increased. Furthermore, combining cost-sensitive learning with resampling by applying weights to the balanced training data could significantly increase the averaged recall to 0.82, although lowering the F2-score to 0.27. Particularly, the weights for each observation are calculated from the imbalanced training set and applied to the balanced training.

The second sub-research question explores which pre-trained image classification neural networks can be fine-tuned to mainly increase the recall performance compared to the F2-score. Among the three evaluated models and their variations, fine-tuning the default weights of the last CNN-based AlexNet layer yielded the highest recall and F2 scores during propagation and when implementing class weights. Additionally, fine-tuning the last two layers of AlexNet improved recall and F2-score to 0.48 and 0.26 compared to the ResNet implementations where the highest recall and F2-score performance for the ResNet18 yielded 0.27 and 0.24 respectively. Between the ResNet50 and ResNet18 implementations can be found that the latter network trains faster due to its shallow layers, although ResNet50 provides more consistent validation and test recall scores. Utilising a pre-trained AlexNet on sewage images proved less optimal performance since it was likely not trained on similar contextual images as those of the sewage pipes in the Netherlands using this project's data. In conclusion, classic pre-trained CNN models can be preferred over ResNet models to increase the recall and F2 score for multi-label classification.

The final sub-research question aims to evaluate the classifier's objectivity compared to manual classification to determine its bias towards certain observations. This subquestion remains unanswered as this thesis has not yet deployed the model to assess its objectivity compared to human inspectors. However, evaluation experiments could assess the classifier's consistency by providing identical inspections to both the model and various human inspectors to analyse interpretation deviations between the inspector and the model. This could reveal systematic biases among human inspectors, such as tendencies to over- or under-report certain observation types. Statistical tests could then determine significant differences between the model's and human inspectors' performance. Furthermore, the inspector's results would be relative to each other and relative to the results of the model, as inspection annotation will always lack an exact ground truth since each inspector interprets observations differently during an inspection. Therefore, a democratic average of different inspectors' annotations could be taken while weighting each annotation based on the inspector's experience and experienced fatigue during labelling. Eventually, this weighted average could serve as an approximated ground truth for prediction comparison with the model.

## 6.2. Limitations

Four modelling factors potentially affecting the results are discussed regarding data quality issues, resampling variations in the data balance, the underperformance of certain observations, and the application of class weights.

First, despite data quality enhancement, erroneous images showing the manhole shaft at the beginning and end of certain inspections remained in the training data. This issue likely arose since the exclusion threshold for the front and back images was based on a qualitative assessment of only 10 inspections. Unassessed inspections might have more extreme image indexes of front and back frames at the beginning and end of the inspections where the manhole appears earlier or later than the applied threshold.

Besides, two limitations during the propagation methods occurred regarding the training set balancing First, fluctuations of the training set size due to varying resampling factors throughout the propagation experiments likely affected the results of the various propagation methods as a controlled assessment of these techniques was limited. Second, an imbalance between observation types accrued as the largest minority group (BCA) could no longer be downsampling while resampling factors for the other observations increased. This phenomenon is evident between the results of experiments 13 and 18 where a propagation of two 2 was applied, although the performance of experiment 18 decreased since the BCA observation was dominant. Hence, this biases the model towards finding BCA's over other observations. In conclusion, the combination of how you resampled the observations is a big factor and can be harder during multi-labelled data.

Deviations in average recall and F2-scores may result from three factors regarding the data splitting of the inspections over the training, test and validation set and the reducing the training dataset size. Firstly, certain experiments showed underrepresentation of the BAF, BAB and BBF when smaller training sets were loaded from choosing lower upsampling factors Secondly, downscaling the dataset in the data class loader by limiting the input dataset to a fixed size could have excluded the rare observations. Particularly, the dataset size was reduced in a non-random effort by only including a specified portion of the head of the loaded data frame. This likely excluded observations appearing later in the balanced training data frame and inspections with more diverse observations could have been omitted as only those with larger frame counts could have been included in the head of the data frame. Thirdly, stratified sampling aimed to maintain the distribution of inspection frame counts across the training, validation, and test sets. Nonetheless, it did not preserve the observation distribution within these sets which may have supported the exclusion of previously mentioned minority observations.

Class weights appeared to be a promising approach for addressing the class imbalance, although their implementation involves three primary limitations related to class weight calculation and training loss function behaviour. Firstly, class weights are calculated from the initial imbalanced training set but applied to a balanced dataset. This approach aims to improve recall performance, as using an imbalanced set with class weights would lack enough rare observations for effective model learning. Additionally, the training loss function plot during class-weight implementation showed greater loss variations per iteration compared to other experiments, suggesting that class weights may contribute to the instability of the model.

## 6.3. Future Work

Based on the limitations various alternative methods could be recommendations regarding the increase of data diversity, balancing methods, data splitting, hyperparameter tuning and selecting sequential prone networks.

### 6.3.1. Data Diversity

Five data adjustments could enhance the dataset's diversity by increasing its data sources, including additional types and continuous observations, and using unwrapping or higher image resolution. First, expanding the dataset to include annotated sewage data from other municipalities could ensure broader applicability of the system through differently located pipes, as relying solely on footage from Apeldoorn may not generalize to the entire Netherlands. Second, excluding PVC and HDPE pipes could omit relevant observations. Hence, while including those two pipe materials, the concrete pipes could be undersampled or the PVC and HDPE pipes upsampled to improve dataset variability. Meanwhile, this could allow for additional observations to contain enough data to be included in the final dataset. Third, implementing continuous annotations with severity levels could make the labelled frames more comprehensive, capturing both continuous and non-continuous observations. Next, unwrapping and stitching front- and back-facing images would provide a full-frame detailed view of each feature. In regards to the temporal labelling location issue, a certain number of frames around the unwrapped frame could be excluded. Finally, higher image resolutions could be implemented to capture finer details of less distinctive observations although that could be computationally demanding. Given that a portion of the input images coming from a fisheye lens are black around the circular circumference of the image, experimenting with circular convolutional neural networks could be beneficial.

### 6.3.2. Data Balancing Methods

Besides currently implemented balancing methods, the resampling and cost-sensitive learning could be expended with other balancing techniques such as ensemble learning. First, with synthetic resampling methods such as Borderline-SMOTE, ADASYN, and k-nearest neighbors with Tomek links or RUS could be experimented with. However, our experiments indicate that cost-sensitive learning, which assigns weights to the minority class, outperforms resampling as the literature stated. This approach enables the model to learn the unique characteristics of the minority class without bias towards the majority class. Furthermore, optimizing the class-weight calculations to assign higher costs to the minority class could be beneficial. This strategy can be combined by assigning higher costs to distinct minority class instances in annotated images to tackle the temporal labeling problem. Besides, ensemble learning boosters often surpass resampling and cost-sensitive learning by implementing techniques such as the XGBoost or TLUSboost. Lastly, since different observations often behave differently in the precision-recall curve, distinct methods can be applied to specific observations rather than the entire dataset, such as resampling or propagation, This targeted approach can improve the performance of each distinct observation based on its unique visual characteristics.

### 6.3.3. Data Splitting

To ensure enough observations are present in the validation and test sets for robust model generalization and reliable result calculation, three methods could be implemented to split the data in different ratios using stratified sampling or applying cross-validation. First, splitting ratios for a higher validation and test set could include more observations in the test and validation set to allow for less distinguishable observations to have a chance to be more representative. Secondly, the problem was experienced where the validation and test set increased drastically in splitting ratio compared to the downsampled training set. Currently, stratified sampling was used for the division of the inspection by preserving the frame count proportion of the whole distribution. Hence, it would also be important for the test and validation sets to undergo stratified sampling to maintain a similar distribution of the observation presence. Nonetheless, to ensure all observations are used for validation, testing, and training, k-fold cross-validation can be implemented. The dataset is divided into k subsets, and the model is trained k times, each time using a different subset as the validation set while the remaining k-1 subsets serve as the training set. This method ensures the model's performance is not dependent on a specific data split and prevents data leakage by iteratively training on different folds and testing on the remaining fold until all subsets have been utilized.

### 6.3.4. Hyperparameter Tuning

Moreover, training the model for an increased number of epochs can allow the model to learn from the data for a longer time. However, this approach may lead to overfitting, which can be mitigated through at least three methods regarding early stopping, regularization, and dropout. First, early stopping could be inserted when the performance on the validation set declines. Secondly, regularization techniques such as L2 regularization to the loss function could be applied to penalise large weights and encourage the model to find simpler patterns. Thirdly, the dropout rate can be experimented with in a pre-trained PyTorch network to allow the model to learn different data patterns. In closing, these and other hyperparameters can be fine-tuned using automated tuning methods such as Optuna or RayTune with pruning to terminate unpromising training cycles and efficiently find optimal parameters. Each hyperparameter can be ranked to identify those with the most potential for model improvement by being adjusted. Furthermore, Explainable AI (XAI) can be utilized by plotting activation weights to understand the patterns recognized by the neural network in specific observations to assess whether the identified feature maps correspond to actual features from the observations.

### 6.3.5. Temporal Dependent Models

Three types of multi-label classification networks may address the sequential nature of the data and mitigate temporal mislabeling of frames around a labelled image. Although CNNs are primarily used in sewage pipe research, other models capable of understanding temporal observations should be introduced. First, Recurrent Neural Networks (RNNs), including Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs), could be implemented to capture long-range dependencies in sequences, despite potential vanishing gradient issues. Second, Temporal Convolutional Networks (TCNs) offer an alternative by using dilated convolutions to capture long-range dependencies in sequential data. Third, Vision Transformers (ViTs) can be employed to consider long-range dependencies across an entire image sequence, although they require significant computational resources and substantial training data.

# 7. Conclusion

This thesis determined how to train a robust multi-label anomaly classifier against labelling inconsistencies to automate sewage inspection. Experimental results demonstrated that fine-tuning a pre-trained convolutional neural network (CNN) with default weights, combined with normalization, histogram equalization for image enhancement and balancing the training data through resampling with applied class weights, showed the most promising results based on the client-provided dataset.

The data preparation followed a specific order due to dependencies in the propagation and offset methods, requiring the merging of equal numbers of front and back images. Therefore, rows containing both front and back images were filtered, followed by the exclusion of diverse start, end, and blue images, ultimately retaining only the front images.

Initial experiments focused on the data splitting ratio, assuming smaller validation and test set ratios would contain enough observations for proficient result calculations. However, observations remained underrepresented despite upsampling to the same count as other observations. Additionally, smaller training ratios unexpectedly resulted in validation or test sets exceeding training images due to post-split downsampling of empty images.

Subsequent experiments evaluated image enhancement techniques to address color and brightness deviations. While brightness and shadow mitigation techniques, along with CLAHE (an improved histogram equalization), were anticipated to allow the model to distinguish finer areas of the observation features, only normalization and global histogram equalization proved effective.

Furthermore, the combination of class weights and upsampling was hypothesized to improve model performance, which was confirmed by the experiments. However, this study employed both sampling and cost-sensitive learning by applying class weights from the imbalanced training set to the loss functions during the training over the balanced set.

It was chosen to focus on multi-label classification as the first modelling implementation since the data was already labelled in a multi-label manner. Besides, regarding the modelling, it was expected that convolutional neural networks would underperform compared to deeper residual neural networks as was assumed that ResNets would learn more complex features. However, the AlexNet CNN increased the recall and F2-score compared to the ResNets. Still, this was assumed based on the literature since AlexNet was proficient for binary normal and defect classification only, while the ResNet could better be used only during in-between observation classification.

As the steps of the CRISP-DM method can be flexible, this thesis combined data understanding and data preparation since the discovery of a correlation would be a direct factor for data filtering. Furthermore, past these two parts, only modelling is mainly used, and the evaluation phase is partly defined as the discussion chapter since real-life deployment assessment of the model is not of interest to this thesis. Additionally, the business understanding is incorporated into the extensive introduction to allow all literature discoveries to already be mentioned in a structured manner in the introduction, including the motivation and problem for this project based on an interview with the client.

Future research should include additional pipe attributes and continuous severity levels, using XGBoost or TLUSboosters for ensemble learning to balance data and applying k-fold cross-validation and optimize hyperparameters to train sequential data-dependent models, such as RNN, LSTM, TCN, or ViTs.

Returning to the problem statement, the robustness of the model against inconsistencies is targeted where this research assumes that the proficient methods found make the model robust against inconsistencies as the training data consists of diverse annotated inspections. However, it is not considered yet if the inspection in the project dataset is annotated by multiple inspectors, as this would determine likely the bias of the model itself. Still, this project showed from its experiment how to not only be robust against inconsistency, but mainly against the imbalanced nature of the data. Where this project can support the literature that cost-sensitive learning indeed could show more prominent results than only implementing resampling. Additionally, this research tackles two new methods not yet found in the literature by using label offset and propagation to deal with the temporal mislabeling localization problem where from the experiments the propagation was found to be proficient. Therefore future research could extend on the propagation method. Although it could be thought of to unwrap and stitch the images for multi-label classification as it has only been used for semantic pixel labelling yet.

# 8. Acknowledgements

During this process, I explored a new field and acquired new skills through continuous discovery. I am especially grateful to Bram, who maintained a critical tone toward my findings, prompting me to reconsider and revise my initial assumptions. I also want to thank Faizan for emphasizing the importance of endurance throughout this project. Their input encouraged me to reexamine the framework and question why certain data was incorrect, leading to iterative adjustments. Besides, I would like to thank Hajo for his help by providing the data and answering data explorative qustions. Additionally, I would like to thank my family for supporting me trough the struggles with understanding certain complex concepts, and to my friends for their empathy and understanding throughout this journey.

References

[1] T. Delft, "Zinvol Rioolonderhoud," TU Delft, 2019. [Online]. Available: https://www.tudelft.nl/citg/onderzoek/stories-of-science/zinvol-rioolonderhoud. [Accessed 27 6 2024].

[2] S. S. Kumar, M. Wang, D. M. Abraham, M. R. Jahanshahi, T. Iseley and J. C. Cheng, "Deep learning–based automated detection of sewer defects in CCTV videos," *Journal of Computing in Civil Engineering,* vol. 34, no. 1, p. 04019047, 2020.

[3] S. Rioned, "Geld - Riool en Raad," Rioned, 23 3 2022. [Online]. Available: https://www.rioolenraad.nl/benodigdheden/geld/. [Accessed 27 6 2024].

[4] J. B. Haurum and T. B. Moeslund, "A survey on image-based automation of CCTV and SSET sewer inspections," *Automation in Construction,* vol. 111, p. 103061, 2020.

[5] J. B. Haurum and T. B. Moeslund, "Sewer-ML: A multi-label sewer defect classification dataset and benchmark," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* Nashville, 2021.

[6] J. Kunzel, T. Werner, P. Eisert and J. Waschnewski, "Automatic Analysis of Sewer Pipes Based on Unrolled Monocular Fisheye Images," *2018 IEEE winter conference on applications of computer vision (WACV),* pp. 2019-2027, 2018.

[7] Q. Xie, D. Li, J. Xu, Z. Yu and J. Wang, "Automatic Detection and Classification of Sewer Defects via Hierarchical Deep Learning," *IEEE Transactions on Automation Science and Engineering,* vol. 16, no. 4, pp. 1836-1847, 2019.

[8] D. Li, A. Cong and S. Guo, "Sewer damage detection from imbalanced CCTV inspection data using deep," *Automation in Construction* , vol. 101, pp. 199-208, 2019.

[9] D. Meijer, L. Scholten, F. Clemens and A. Knobbe, "A defect classification methodology for sewer image sets with convolutional neural networks," *Automation in Construction,* vol. 104, pp. 281-298, 2019.

[10] J. Zhang, X. Liu, X. Zhang, Z. Xi and S. Wang, "Automatic Detection Method of Sewer Pipe Defects Using Deep Learning Techniques," *Applied Sciences,* vol. 13, no. 7, p. 4589, 2023.

[11] S. Hassan, L. Dang, I. Mehmood, S. Im, C. Choi, J. Kang, Y. Park and H. Moon, "Underground sewer pipe condition assessment based on convolutional neural networks," *Automation in Construction,* vol. 106, p. 102849, 2019.

[12] L. Sun, J. Zhu, J. Tan, X. Li, R. Li, H. Deng, X. Zhang, B. Liu and X. Zhu, "Deep learning-assisted automated sewage pipe defect detection for urban water environment management," *Science of The Total Environment,* vol. 882, p. 163562, 2023.

[13] J. Cheng and M. Wang, "Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques," *Automation in Construction,* vol. 95, pp. 155-171, 2018.

[14] X. Yin, Y. Chen, A. Bouferguene, H. Zaman, M. Al-Hussein and L. & Kurach, "A deep learning-based framework for an automated defect detection system for sewer pipes," *Automation in construction,* vol. 109, p. 102967, 2020.

[15] D. Li, Q. Xie, Z. Yu, Q. Wu, J. Zhou and J. Wang, "Sewer pipe defect detection via deep learning with local and global feature fusion," *Automation in Construction,* vol. 129, p. 103823, 2021.

[16] M. L. Dang, S. Kyeong, Y. Li, H. Wang and T. N. Nguyen, "Deep learning-based sewer defect classification for highly imbalanced dataset," *Computers & Industrial Engineering,* vol. 161, p. 107630, 2021.

[17] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on knowledge and data engineering,* vol. 21, no. 9, pp. 1263-1284, 2009.

[18] S. Kumar, S. Biswas and D. Devi, "TLUSBoost algorithm: a boosting solution for class imbalance problem," *Soft Comput,* vol. 23, pp. 10755-10767, 2019.

[19] K. Chen, H. Hu and L. H. C. Chen, "An Intelligent Sewer Defect Detection Method Based on Convolutional Neural Network," in *2018 IEEE International Conference on Information and Automation (ICIA),* Wuyishan, 2018.

[20] D. Ma, J. Liu, H. Fang, N. Wang and C. Zhang, "A Multi-defect detection system for sewer pipelines based on StyleGAN-SDM and fusion CNN," *Consruction and Building Materials,* vol. 312, p. 125385, 2021.

# 9. Appendix

## 9.1. Appendix A

This appendix contains further descriptions of the data sources. Of which the following Fig A.1 explains the directory structure of the inspection image folders.

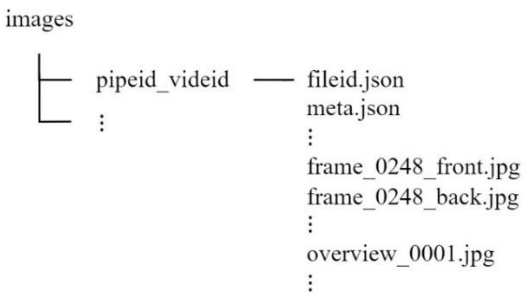


*Fig A.1: Directory structure of the image data consisting of various inspection folders.*

In the following Fig. A.2, the *file3* takes only the first three letters of the fileid, which may be promising in reducing data retrieval time while forming the custom dataset.

```
"fileid" : videoid
"file3" : abbreviated videoid
"assetid" : pipeid
"datetime" : inspection date
```

```
{ "fileid": "e7c73c45b47f40d58a3c396edcbbdd56",
  "file3": "e7c",
  "assetid": "02febe3dc4894e98abd5381501fbc4f9",
  "datetime": "2020-02-12 00:00:00" }
```

*Fig. A.2: a) Explained keys of the fileid.json file, b) Example of keys with values of an existing inspection.*

The meta.json file contains relevant detailed inspection information about the frame resolution, and camera settings, allowing allocating labelling locations and advanced image processing of which the key attributes are discussed in Fig. 3 in millimetres (the untranslated and complete meta.json content is available in Appendix …). First, the count and pixel resolution of captured frames and overview images are provided, with *frameCount* including capture times for both front and back images, therefore sharing the same image index. Secondly, *pipe* details, such as its identifier and diameter in millimetres are noted. Besides, for the *startPosition*, positive values showing a later start and negative values an earlier one. Lastly, the *stepDistance* marks the incremental distance between images in centimeters.

Most inspections lack the additional *props* (properties) section, which offers useful camera information for image processing. The *center* coordinates specify the pixel-wise location of the pipe centre in the initial front and back images, which may vary due to robot positioning. *DN* (Diameter Nominal) and specifies the pipe's nominal diameter in millimetres. These last two attributes may be interesting for potential image unwrapping. The *Distance* refers to the *stepDistance*, but in millimeters. Moreover, similar to *startPosition*, *StartPos* indicates the start of the capturing distance in millimetres from the pipe's starting point, with *StartImage* denoting the reference starting image index. The *TotalFrames* matches the *frameCount*. Besides, *Deviation_BrightnessCompensation* and *Deviation_HD* as imaging options, indicate respectively that brightness and high-definition processing are not applied. Lastly, the *PipeWidth* matches *pipe.diameter* in millimetres, and *PipeInspDate* corresponds to the *datetime* of the file.json file.

```
"frameCount":250, "frameResolution":{"x":1040,"y":1040},
"overviewCount":13, "overviewResolution":{"x":320,"y":503},
"pipe":{"name":"520-46","diameter":500},
"startPosition":110, "stepDistance":5,
"props":[...
        "center_y1=516","center_x1=520","center_y2=514","center_x2=520",
        …
        "DN=500",
        "Distance=50",
        …
        "StartPos=1100","StartImage=22",
        "TotalFrames=250",
        …
        "Deviation_BrightnessCompensation=0","Deviation_HD=0",
        …
        "PipeWidth=500",
        …
        "PipeInspDate=12-2-2020 16:04:30",
```

*Fig. A.3: Example of the filtered relevant content in the meta.json file*

Full content meta.json file in the original German language:

```
"frameCount":250, "frameResolution":{"x":1040,"y":1040},
"overviewCount":13, "overviewResolution":{"x":320,"y":503},
"pipe":{"name":"520-46","diameter":500},
"startPosition":110,
"stepDistance":5,
"props":[
        "Panoramo.ips","41","[Init]","Kennung=2","InfoText=520-46",
        "zentrum_y1=516","zentrum_x1=520","zentrum_y2=514","zentrum_x2=520",
        "OWinkel_1=517","OWinkel_2=509",
        "Profil=0",
        "DN=500",
        "BAbstand=50","SPixel=16",
        "StartPos=1100","StartBild=22",
        "AnzahlBilder=250",
        "IPF_filesize=101326",
        "ColorPlus=1","Abw_Helligkeitausgleich=0","ABW_HD=0",
        "[Main]","HaltungsNr=520-46",
        "HaltungsBreite=500","HaltungsInspNr=109","HaltungsInspDatum=12-2-2020
        16:04:30","HaltungsInspRichtung=gegen",v
        "LeitName=","LeitungsBreite=","LeitungsInspNr=","LeitungsInspDatum=","LeitungsInspRichtu
        ng=","SchachtNr=18000","Pfad_Filme=D:\\IKASDATA\\IKAS32
        Films\\","Pfad_Profiler=D:\\IKASDATA\\IKAS32ProfilerFilms\\","Projekt=39520108A","Datu
        m=","SchachtLaenge=","SchachtBreite=","SchachtInspNr=","SchachtInspDatum=","HaltungsHo
        ehe=500"
        ]
```

*Fig. A.4: Example of original relevant content in the meta.json file.*

The following table shows the additional attributes of the pipe data.

| Attribute | Importance | Data Type | Value |
|---|---|---|---|
| pipeid | x | textual | pipe Uuid, eg. 00094c07091a4c689781f6c0bf9f31f2 |
| pipesystem | | Categorical | gravitySewer, mechanicSewer |

| | | | |
|---|---|---|---|
| pipecontent | x | Categorical | MixedSewer, WasteSewer, RainSewer, Unkown |
| pipefunction | | Categorical | Transport, Infiltration, Culvert, Unkown, Connection, Drain |
| pipeshape | x | Categorical | Round, Unkown, Ovoid, Square |
| pipefoundation | | Categorical | OnSteel, NaN |
| width | | Numerical | Pipewidth in mm |
| height | | Numerical | Pipeheight in mm |
| pipelength | | Numerical | Length in m |
| construcion_date | | Date/Time | Format: 01/01/1974 |
| material | x | Categorical | Normalized material of the pipe: Concrete, PVC, Other, PE, Unkown, HDPE, VitrifiedClay, ReinforcedConcrete, Steel |
| material_raw | | Categorical | Original (unnormalized) material of the pipe: beton ongewapend, PVC, overig, PE, beton infiltratie, onbekend, PVC infiltratie, HPE, gres, beton gewapend, Z, staal |

*Table A.1:Pipe dataset and its attributes, their importance, datatypes, and explanations or examples.*

Additional attributes of the annotation data are discussed which may be of future interest to expand the labeling information. The *position* and *video_time indicate* where in the pipe and when an observation is made during the inspection. Additionally, *end_position* and *video_end_time* can be filled or left empty, denoting if an observation extends over a longer distance and when it ends in the pipe. The *reference* attribute indicates whether the inspection started at the official start or end of the pipe. The *sequence* attribute cumulates the number of observations in one inspection. In particular, the sequence number is set to zero for observations in the first frame, often including observations spanning certain distances in the pipe. Moreover, various *characteristics* and their distinct or ranging *quantizations* of each *code* can be recorded. In addition, the *clockface* reference coordinates localize the radial surface where the observation occurred, specifying a starting and ending degree in a clockwise manner. Lastly, the *damage class* denotes the severity of the detected defect or observation.

| Attribute | Importance | Data Type | Value |
|---|---|---|---|
| pipeid | x | textual | pipe uuid |
| inspectionid | x | textual | inspection uuid |
| datetime | x | date/time | inspection date, 20/08/2010 |
| position | x | numerical | in-pipe position in m |
| end_position | x | numerical | end position in pipe in m |
| reference | | categorical | starting reference, FromStartPit, Unkown, FromEndPit |
| seq | | numerical | sequence number |
| code | x | categorical | damage code, eg. BCA |
| kar1 | | categorical | damage characterization 1, A, B, etc. |
| kar2 | | categorical | damage characterization 2 |
| kar3 | | categorical | damage characterization 3 |
| kwan1 | | numerical | damage quantification 1 |
| kwan2 | | numerical | damage quantification 2 |
| clock1 | | numerical | start position on circumference of pipe, degrees |
| clock2 | | numerical | end position on circumference of pipe, degrees |
| damage_class | | categorical | damage classification: 2, 3, 4 or 5 |
| video_time | | numerical | timestamp in video in ms |
| video_end_time | | numerical | end timestamp in video in ms |

*Table A.2: Annotation dataset and its attributes, their importance, datatypes, and explanations or examples.*

Based on the relevant attributes of the pipe and annotation files, an entity-relationship diagram (see Fig. A.3) is constructed to illustrate the relationships among pipes, annotations, image data, and metadata entities. This diagram assists in visualizing the integration of these components for the ease of constructing a combined dataset and performing an exploratory analysis. A one-to-many relationship exists between pipes and inspection datasets, linked by the primary key pipeid, indicating that a single pipe can relate to multiple inspections and videos. Furthermore, each inspection contains a unique set of images identified by a unique videoid. Consequently, a folder with a unique videoid contains multiple frames, each with a different frame number. Besides, regarding the inspection metadata, the fileid.json file contains a related inspection pipeid as assetid. Furthermore, its datetime may not be unique within the annotation file, as multiple inspections can occur on the same day. Additionally, the start position in meta.json is unique for each inspection, though multiple inspections may share the same reference start position. This start position links the frame number to the locations of identified observations within the inspection dataset.

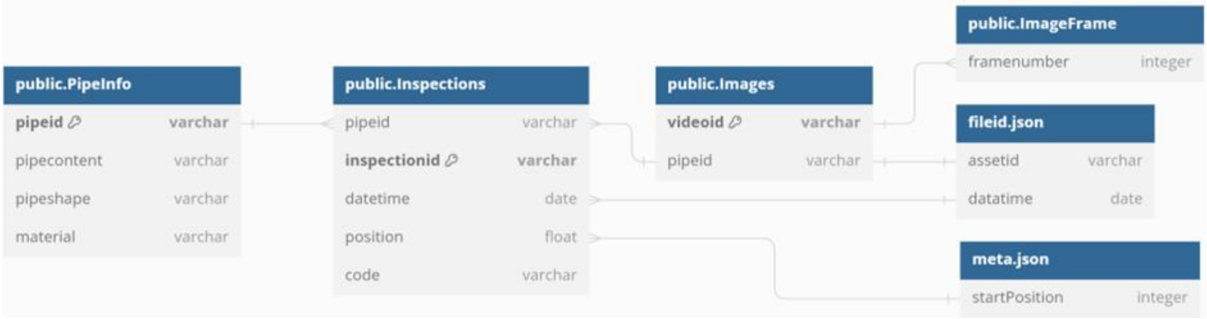
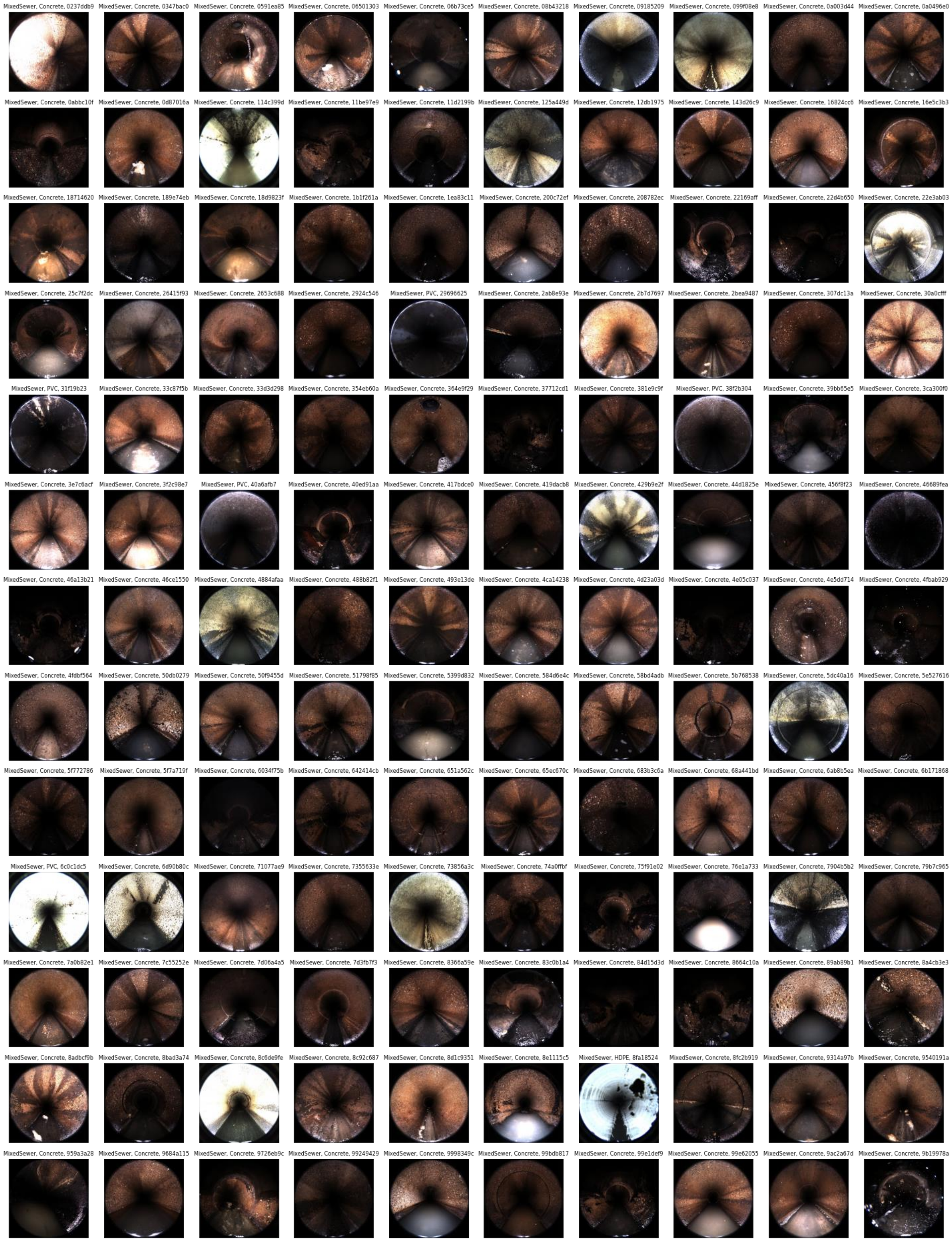
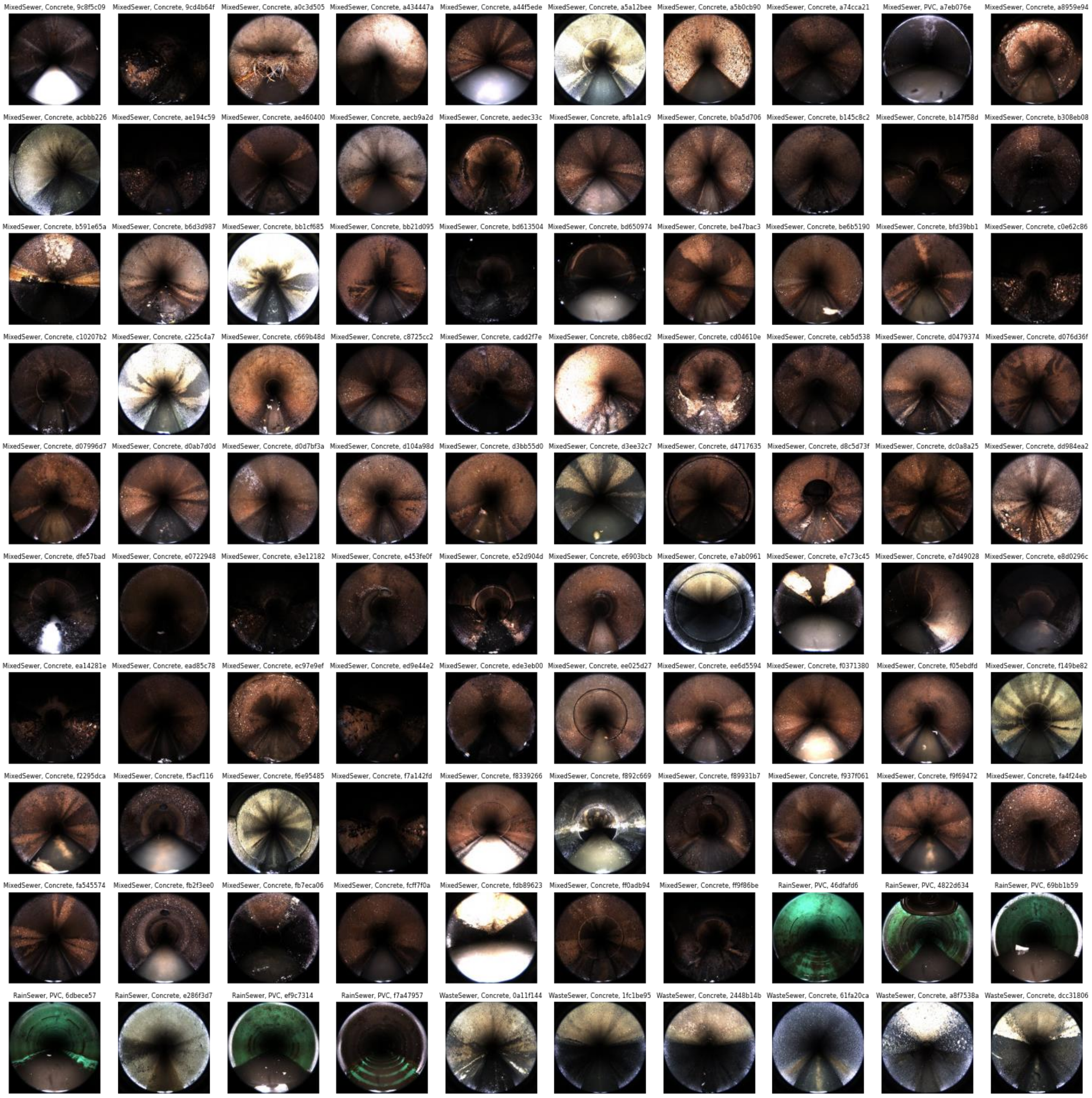


*Fig. A.3 Entity-relationship diagram of the pipe, annotation, and image data.*

## 9.2. Appendix B

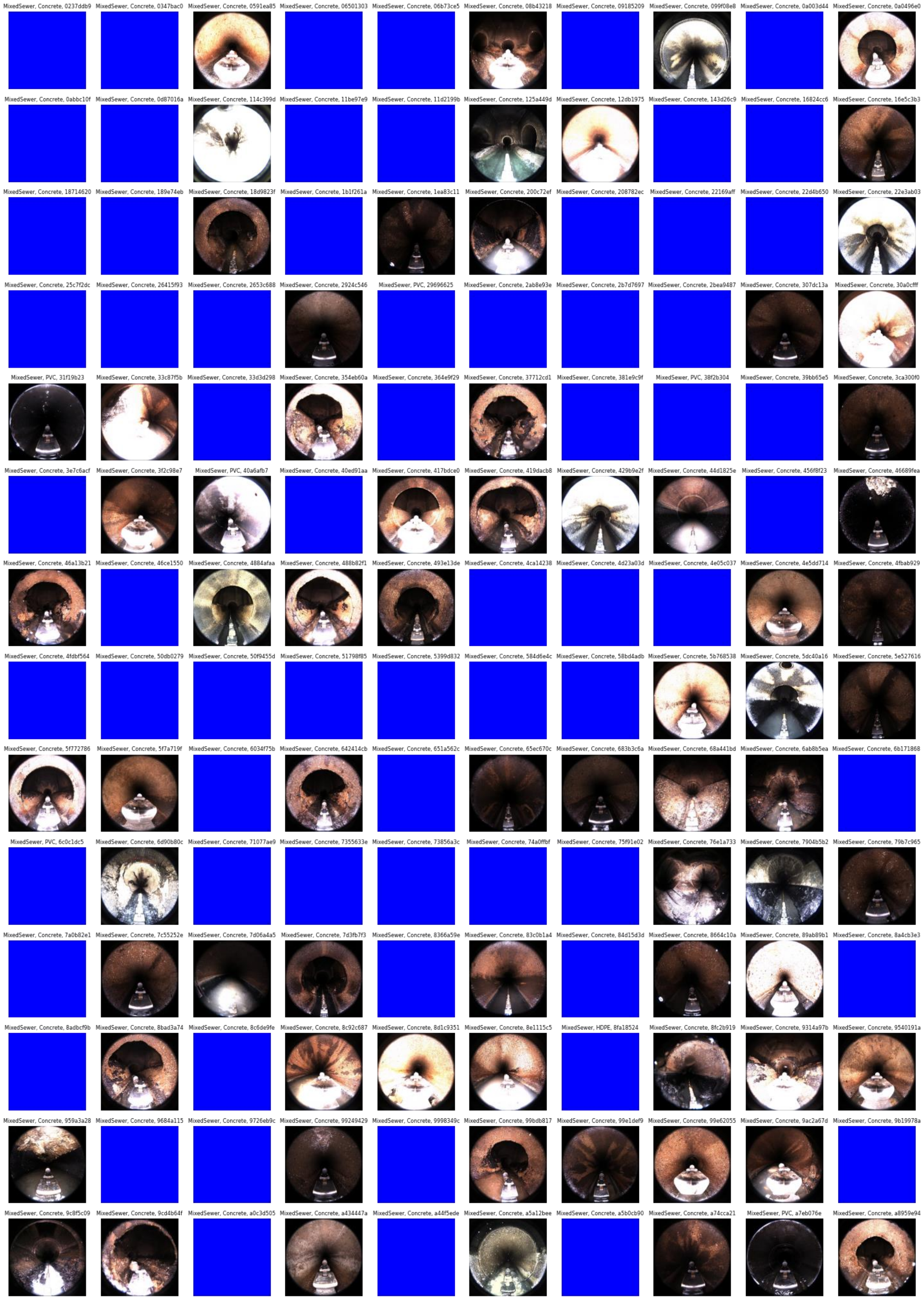
8th Front Frame for each videoid while annotating the pipe content and pipe material.

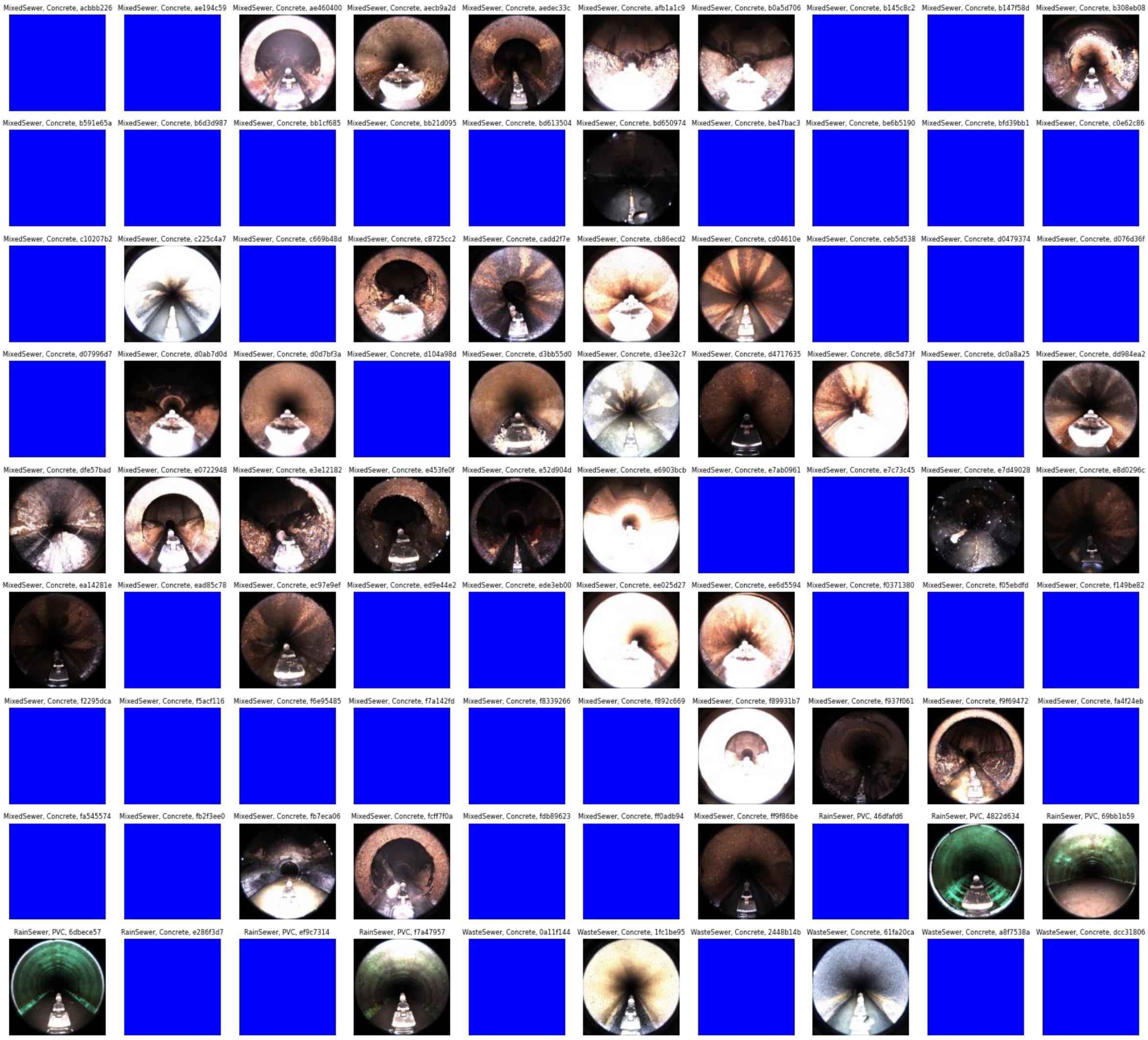

MixedSewer, Concrete, 0237ddb9
MixedSewer, Concrete, 0347bac0
MixedSewer, Concrete, 0591ea85
MixedSewer, Concrete, 06501303
MixedSewer, Concrete, 06b73ce5
MixedSewer, Concrete, 08b43218
MixedSewer, Concrete, 09185209
MixedSewer, Concrete, 099f08e8
MixedSewer, Concrete, 0a003d44
MixedSewer, Concrete, 0a0496e0
MixedSewer, Concrete, 0abbc10f
MixedSewer, Concrete, 0d87016a
MixedSewer, Concrete, 114c399d
MixedSewer, Concrete, 11be97e9
MixedSewer, Concrete, 11d2199b
MixedSewer, Concrete, 125a449d
MixedSewer, Concrete, 12db1975
MixedSewer, Concrete, 143d26c9
MixedSewer, Concrete, 16824cc6
MixedSewer, Concrete, 16e5c3b3
MixedSewer, Concrete, 18714620
MixedSewer, Concrete, 189e74eb
MixedSewer, Concrete, 18d9823f
MixedSewer, Concrete, 1b1f261a
MixedSewer, Concrete, 1ea83c11
MixedSewer, Concrete, 200c72ef
MixedSewer, Concrete, 208782ec
MixedSewer, Concrete, 22169aff
MixedSewer, Concrete, 22d4b650
MixedSewer, Concrete, 22e3ab03
MixedSewer, Concrete, 25c7f2dc
MixedSewer, Concrete, 26415f93
MixedSewer, Concrete, 2653c688
MixedSewer, Concrete, 2924c546
MixedSewer, PVC, 29696625
MixedSewer, Concrete, 2ab8e93e
MixedSewer, Concrete, 2b7d7697
MixedSewer, Concrete, 2bea9487
MixedSewer, Concrete, 307dc13a
MixedSewer, Concrete, 30a0cfff
MixedSewer, PVC, 31f19b23
MixedSewer, Concrete, 33c87f5b
MixedSewer, Concrete, 33d3d298
MixedSewer, Concrete, 354eb60a
MixedSewer, Concrete, 364e9f29
MixedSewer, Concrete, 37712cd1
MixedSewer, Concrete, 381e9c9f
MixedSewer, PVC, 38f2b304
MixedSewer, Concrete, 39bb65e5
MixedSewer, Concrete, 3ca300f0
MixedSewer, Concrete, 3e7c6acf
MixedSewer, Concrete, 3f2c98e7
MixedSewer, PVC, 40a6afb7
MixedSewer, Concrete, 40ed91aa
MixedSewer, Concrete, 417bdce0
MixedSewer, Concrete, 419dacb8
MixedSewer, Concrete, 429b9e2f
MixedSewer, Concrete, 44d1825e
MixedSewer, Concrete, 456f8f23
MixedSewer, Concrete, 46689fea
MixedSewer, Concrete, 46a13b21
MixedSewer, Concrete, 46ce1550
MixedSewer, Concrete, 4884afaa
MixedSewer, Concrete, 488b82f1
MixedSewer, Concrete, 493e13de
MixedSewer, Concrete, 4ca14238
MixedSewer, Concrete, 4d23a03d
MixedSewer, Concrete, 4e05c037
MixedSewer, Concrete, 4e5dd714
MixedSewer, Concrete, 4fbab929
MixedSewer, Concrete, 4fdbf564
MixedSewer, Concrete, 50db0279
MixedSewer, Concrete, 50f9455d
MixedSewer, Concrete, 51798f85
MixedSewer, Concrete, 5399d832
MixedSewer, Concrete, 584d6e4c
MixedSewer, Concrete, 58bd4adb
MixedSewer, Concrete, 5b768538
MixedSewer, Concrete, 5dc40a16
MixedSewer, Concrete, 5e527616
MixedSewer, Concrete, 5f772786
MixedSewer, Concrete, 5f7a719f
MixedSewer, Concrete, 6034f75b
MixedSewer, Concrete, 642414cb
MixedSewer, Concrete, 651a562c
MixedSewer, Concrete, 65ec670c
MixedSewer, Concrete, 683b3c6a
MixedSewer, Concrete, 68a441bd
MixedSewer, Concrete, 6ab8b5ea
MixedSewer, Concrete, 6b171868
MixedSewer, PVC, 6c0c1dc5
MixedSewer, Concrete, 6d90b80c
MixedSewer, Concrete, 71077ae9
MixedSewer, Concrete, 7355633e
MixedSewer, Concrete, 73856a3c
MixedSewer, Concrete, 74a0ffbf
MixedSewer, Concrete, 75f91e02
MixedSewer, Concrete, 76e1a733
MixedSewer, Concrete, 7904b5b2
MixedSewer, Concrete, 79b7c965
MixedSewer, Concrete, 7a0b82e1
MixedSewer, Concrete, 7c55252e
MixedSewer, Concrete, 7d06a4a5
MixedSewer, Concrete, 7d3fb7f3
MixedSewer, Concrete, 8366a59e
MixedSewer, Concrete, 83c0b1a4
MixedSewer, Concrete, 84d15d3d
MixedSewer, Concrete, 8664c10a
MixedSewer, Concrete, 89ab89b1
MixedSewer, Concrete, 8a4cb3e3
MixedSewer, Concrete, 8adbcf9b
MixedSewer, Concrete, 8bad3a74
MixedSewer, Concrete, 8c6de9fe
MixedSewer, Concrete, 8c92c687
MixedSewer, Concrete, 8d1c9351
MixedSewer, Concrete, 8e1115c5
MixedSewer, HDPE, 8fa18524
MixedSewer, Concrete, 8fc2b919
MixedSewer, Concrete, 9314a97b
MixedSewer, Concrete, 9540191a
MixedSewer, Concrete, 959a3a28
MixedSewer, Concrete, 9684a115
MixedSewer, Concrete, 9726eb9c
MixedSewer, Concrete, 99249429
MixedSewer, Concrete, 9998349c
MixedSewer, Concrete, 99bdb817
MixedSewer, Concrete, 99e1def9
MixedSewer, Concrete, 99e62055
MixedSewer, Concrete, 9ac2a67d
MixedSewer, Concrete, 9b19978a


MixedSewer, Concrete, 9c8f5c09
MixedSewer, Concrete, 9cd4b64f
MixedSewer, Concrete, a0c3d505
MixedSewer, Concrete, a434447a
MixedSewer, Concrete, a44f5ede
MixedSewer, Concrete, a5a12bee
MixedSewer, Concrete, a5b0cb90
MixedSewer, Concrete, a74cca21
MixedSewer, PVC, a7eb076e
MixedSewer, Concrete, a8959e94
MixedSewer, Concrete, acbbb226
MixedSewer, Concrete, ae194c59
MixedSewer, Concrete, ae460400
MixedSewer, Concrete, aecb9a2d
MixedSewer, Concrete, aedec33c
MixedSewer, Concrete, afb1a1c9
MixedSewer, Concrete, b0a5d706
MixedSewer, Concrete, b145c8c2
MixedSewer, Concrete, b147f58d
MixedSewer, Concrete, b308eb08
MixedSewer, Concrete, b591e65a
MixedSewer, Concrete, b6d3d987
MixedSewer, Concrete, bb1cf685
MixedSewer, Concrete, bb21d095
MixedSewer, Concrete, bd613504
MixedSewer, Concrete, bd650974
MixedSewer, Concrete, be47bac3
MixedSewer, Concrete, be6b5190
MixedSewer, Concrete, bfd39bb1
MixedSewer, Concrete, c0e62c86
MixedSewer, Concrete, c10207b2
MixedSewer, Concrete, c225c4a7
MixedSewer, Concrete, c669b48d
MixedSewer, Concrete, c8725cc2
MixedSewer, Concrete, cadd2f7e
MixedSewer, Concrete, cb86ecd2
MixedSewer, Concrete, cd04610e
MixedSewer, Concrete, ceb5d538
MixedSewer, Concrete, d0479374
MixedSewer, Concrete, d076d36f
MixedSewer, Concrete, d07996d7
MixedSewer, Concrete, d0ab7d0d
MixedSewer, Concrete, d0d7bf3a
MixedSewer, Concrete, d104a98d
MixedSewer, Concrete, d3bb55d0
MixedSewer, Concrete, d3ee32c7
MixedSewer, Concrete, d4717635
MixedSewer, Concrete, d8c5d73f
MixedSewer, Concrete, dc0a8a25
MixedSewer, Concrete, dd984ea2
MixedSewer, Concrete, dfe57bad
MixedSewer, Concrete, e0722948
MixedSewer, Concrete, e3e12182
MixedSewer, Concrete, e453fe0f
MixedSewer, Concrete, e52d904d
MixedSewer, Concrete, e6903bcb
MixedSewer, Concrete, e7ab0961
MixedSewer, Concrete, e7c73c45
MixedSewer, Concrete, e7d49028
MixedSewer, Concrete, e8d0296c
MixedSewer, Concrete, ea14281e
MixedSewer, Concrete, ead85c78
MixedSewer, Concrete, ec97e9ef
MixedSewer, Concrete, ed9e44e2
MixedSewer, Concrete, ede3eb00
MixedSewer, Concrete, ee025d27
MixedSewer, Concrete, ee6d5594
MixedSewer, Concrete, f0371380
MixedSewer, Concrete, f05ebdfd
MixedSewer, Concrete, f149be82
MixedSewer, Concrete, f2295dca
MixedSewer, Concrete, f5acf116
MixedSewer, Concrete, f6e95485
MixedSewer, Concrete, f7a142fd
MixedSewer, Concrete, f8339266
MixedSewer, Concrete, f892c669
MixedSewer, Concrete, f89931b7
MixedSewer, Concrete, f937f061
MixedSewer, Concrete, f9f69472
MixedSewer, Concrete, fa4f24eb
MixedSewer, Concrete, fa545574
MixedSewer, Concrete, fb2f3ee0
MixedSewer, Concrete, fb7eca06
MixedSewer, Concrete, fcff7f0a
MixedSewer, Concrete, fdb89623
MixedSewer, Concrete, ff0adb94
MixedSewer, Concrete, ff9f86be
RainSewer, PVC, 46dfafd6
RainSewer, PVC, 4822d634
RainSewer, PVC, 69bb1b59
RainSewer, PVC, 6dbece57
RainSewer, Concrete, e286f3d7
RainSewer, PVC, ef9c7314
RainSewer, PVC, f7a47957
WasteSewer, Concrete, 0a11f144
WasteSewer, Concrete, 1fc1be95
WasteSewer, Concrete, 2448b14b
WasteSewer, Concrete, 61fa20ca
WasteSewer, Concrete, a8f7538a
WasteSewer, Concrete, dcc31806

## 9.3. Appendix C

Last 13th back frame for each videoid while annotating the pipe content and pipe material. At this timestamp, the first images tend to obtain a full black color, due to ending the inspection or an error.


MixedSewer, Concrete, 0237ddb9
MixedSewer, Concrete, 0347bac0
MixedSewer, Concrete, 0591ea85
MixedSewer, Concrete, 06501303
MixedSewer, Concrete, 06b73ce5
MixedSewer, Concrete, 08b43218
MixedSewer, Concrete, 09185209
MixedSewer, Concrete, 099f08e8
MixedSewer, Concrete, 0a003d44
MixedSewer, Concrete, 0a0496e0
MixedSewer, Concrete, 0abbc10f
MixedSewer, Concrete, 0d87016a
MixedSewer, Concrete, 114c399d
MixedSewer, Concrete, 11be97e9
MixedSewer, Concrete, 11d2199b
MixedSewer, Concrete, 125a449d
MixedSewer, Concrete, 12db1975
MixedSewer, Concrete, 143d26c9
MixedSewer, Concrete, 16824cc6
MixedSewer, Concrete, 16e5c3b3
MixedSewer, Concrete, 18714620
MixedSewer, Concrete, 189e74eb
MixedSewer, Concrete, 18d9823f
MixedSewer, Concrete, 1b1f261a
MixedSewer, Concrete, 1ea83c11
MixedSewer, Concrete, 200c72ef
MixedSewer, Concrete, 208782ec
MixedSewer, Concrete, 22169aff
MixedSewer, Concrete, 22d4b650
MixedSewer, Concrete, 22e3ab03
MixedSewer, Concrete, 25c7f2dc
MixedSewer, Concrete, 26415f93
MixedSewer, Concrete, 2653c688
MixedSewer, Concrete, 2924c546
MixedSewer, PVC, 29696625
MixedSewer, Concrete, 2ab8e93e
MixedSewer, Concrete, 2b7d7697
MixedSewer, Concrete, 2bea9487
MixedSewer, Concrete, 307dc13a
MixedSewer, Concrete, 30a0cfff
MixedSewer, PVC, 31f19b23
MixedSewer, Concrete, 33c87f5b
MixedSewer, Concrete, 33d3d298
MixedSewer, Concrete, 354eb60a
MixedSewer, Concrete, 364e9f29
MixedSewer, Concrete, 37712cd1
MixedSewer, Concrete, 381e9c9f
MixedSewer, PVC, 38f2b304
MixedSewer, Concrete, 39bb65e5
MixedSewer, Concrete, 3ca300f0
MixedSewer, Concrete, 3e7c6acf
MixedSewer, Concrete, 3f2c98e7
MixedSewer, PVC, 40a6afb7
MixedSewer, Concrete, 40ed91aa
MixedSewer, Concrete, 417bdce0
MixedSewer, Concrete, 419dacb8
MixedSewer, Concrete, 429b9e2f
MixedSewer, Concrete, 44d1825e
MixedSewer, Concrete, 456f8f23
MixedSewer, Concrete, 46689fea
MixedSewer, Concrete, 46a13b21
MixedSewer, Concrete, 46ce1550
MixedSewer, Concrete, 4884afaa
MixedSewer, Concrete, 488b82f1
MixedSewer, Concrete, 493e13de
MixedSewer, Concrete, 4ca14238
MixedSewer, Concrete, 4d23a03d
MixedSewer, Concrete, 4e05c037
MixedSewer, Concrete, 4e5dd714
MixedSewer, Concrete, 4fbab929
MixedSewer, Concrete, 4fdbf564
MixedSewer, Concrete, 50db0279
MixedSewer, Concrete, 50f9455d
MixedSewer, Concrete, 51798f85
MixedSewer, Concrete, 5399d832
MixedSewer, Concrete, 584d6e4c
MixedSewer, Concrete, 58bd4adb
MixedSewer, Concrete, 5b768538
MixedSewer, Concrete, 5dc40a16
MixedSewer, Concrete, 5e527616
MixedSewer, Concrete, 5f772786
MixedSewer, Concrete, 5f7a719f
MixedSewer, Concrete, 6034f75b
MixedSewer, Concrete, 642414cb
MixedSewer, Concrete, 651a562c
MixedSewer, Concrete, 65ec670c
MixedSewer, Concrete, 683b3c6a
MixedSewer, Concrete, 68a441bd
MixedSewer, Concrete, 6ab8b5ea
MixedSewer, Concrete, 6b171868
MixedSewer, PVC, 6c0c1dc5
MixedSewer, Concrete, 6d90b80c
MixedSewer, Concrete, 71077ae9
MixedSewer, Concrete, 7355633e
MixedSewer, Concrete, 73856a3c
MixedSewer, Concrete, 74a0ffbf
MixedSewer, Concrete, 75f91e02
MixedSewer, Concrete, 76e1a733
MixedSewer, Concrete, 7904b5b2
MixedSewer, Concrete, 79b7c965
MixedSewer, Concrete, 7a0b82e1
MixedSewer, Concrete, 7c55252e
MixedSewer, Concrete, 7d06a4a5
MixedSewer, Concrete, 7d3fb7f3
MixedSewer, Concrete, 8366a59e
MixedSewer, Concrete, 83c0b1a4
MixedSewer, Concrete, 84d15d3d
MixedSewer, Concrete, 8664c10a
MixedSewer, Concrete, 89ab89b1
MixedSewer, Concrete, 8a4cb3e3
MixedSewer, Concrete, 8adbcf9b
MixedSewer, Concrete, 8bad3a74
MixedSewer, Concrete, 8c6de9fe
MixedSewer, Concrete, 8c92c687
MixedSewer, Concrete, 8d1c9351
MixedSewer, Concrete, 8e1115c5
MixedSewer, HDPE, 8fa18524
MixedSewer, Concrete, 8fc2b919
MixedSewer, Concrete, 9314a97b
MixedSewer, Concrete, 9540191a
MixedSewer, Concrete, 959a3a28
MixedSewer, Concrete, 9684a115
MixedSewer, Concrete, 9726eb9c
MixedSewer, Concrete, 99249429
MixedSewer, Concrete, 9998349c
MixedSewer, Concrete, 99bdb817
MixedSewer, Concrete, 99e1def9
MixedSewer, Concrete, 99e62055
MixedSewer, Concrete, 9ac2a67d
MixedSewer, Concrete, 9b19978a
MixedSewer, Concrete, 9c8f5c09
MixedSewer, Concrete, 9cd4b64f
MixedSewer, Concrete, a0c3d505
MixedSewer, Concrete, a434447a
MixedSewer, Concrete, a44f5ede
MixedSewer, Concrete, a5a12bee
MixedSewer, Concrete, a5b0cb90
MixedSewer, Concrete, a74cca21
MixedSewer, PVC, a7eb076e
MixedSewer, Concrete, a8959e94


MixedSewer, Concrete, acbbb226
MixedSewer, Concrete, ae194c59
MixedSewer, Concrete, ae460400
MixedSewer, Concrete, aecb9a2d
MixedSewer, Concrete, aedec33c
MixedSewer, Concrete, afb1a1c9
MixedSewer, Concrete, b0a5d706
MixedSewer, Concrete, b145c8c2
MixedSewer, Concrete, b147f58d
MixedSewer, Concrete, b308eb08
MixedSewer, Concrete, b591e65a
MixedSewer, Concrete, b6d3d987
MixedSewer, Concrete, bb1cf685
MixedSewer, Concrete, bb21d095
MixedSewer, Concrete, bd613504
MixedSewer, Concrete, bd650974
MixedSewer, Concrete, be47bac3
MixedSewer, Concrete, be6b5190
MixedSewer, Concrete, bfd39bb1
MixedSewer, Concrete, c0e62c86
MixedSewer, Concrete, c10207b2
MixedSewer, Concrete, c225c4a7
MixedSewer, Concrete, c669b48d
MixedSewer, Concrete, c8725cc2
MixedSewer, Concrete, cadd2f7e
MixedSewer, Concrete, cb86ecd2
MixedSewer, Concrete, cd04610e
MixedSewer, Concrete, ceb5d538
MixedSewer, Concrete, d0479374
MixedSewer, Concrete, d076d36f
MixedSewer, Concrete, d07996d7
MixedSewer, Concrete, d0ab7d0d
MixedSewer, Concrete, d0d7bf3a
MixedSewer, Concrete, d104a98d
MixedSewer, Concrete, d3bb55d0
MixedSewer, Concrete, d3ee32c7
MixedSewer, Concrete, d4717635
MixedSewer, Concrete, d8c5d73f
MixedSewer, Concrete, dc0a8a25
MixedSewer, Concrete, dd984ea2
MixedSewer, Concrete, dfe57bad
MixedSewer, Concrete, e0722948
MixedSewer, Concrete, e3e12182
MixedSewer, Concrete, e453fe0f
MixedSewer, Concrete, e52d904d
MixedSewer, Concrete, e6903bcb
MixedSewer, Concrete, e7ab0961
MixedSewer, Concrete, e7c73c45
MixedSewer, Concrete, e7d49028
MixedSewer, Concrete, e8d0296c
MixedSewer, Concrete, ea14281e
MixedSewer, Concrete, ead85c78
MixedSewer, Concrete, ec97e9ef
MixedSewer, Concrete, ed9e44e2
MixedSewer, Concrete, ede3eb00
MixedSewer, Concrete, ee025d27
MixedSewer, Concrete, ee6d5594
MixedSewer, Concrete, f0371380
MixedSewer, Concrete, f05ebdfd
MixedSewer, Concrete, f149be82
MixedSewer, Concrete, f2295dca
MixedSewer, Concrete, f5acf116
MixedSewer, Concrete, f6e95485
MixedSewer, Concrete, f7a142fd
MixedSewer, Concrete, f8339266
MixedSewer, Concrete, f892c669
MixedSewer, Concrete, f89931b7
MixedSewer, Concrete, f937f061
MixedSewer, Concrete, f9f69472
MixedSewer, Concrete, fa4f24eb
MixedSewer, Concrete, fa545574
MixedSewer, Concrete, fb2f3ee0
MixedSewer, Concrete, fb7eca06
MixedSewer, Concrete, fcff7f0a
MixedSewer, Concrete, fdb89623
MixedSewer, Concrete, ff0adb94
MixedSewer, Concrete, ff9f86be
RainSewer, PVC, 46dfafd6
RainSewer, PVC, 4822d634
RainSewer, PVC, 69bb1b59
RainSewer, PVC, 6dbece57
RainSewer, Concrete, e286f3d7
RainSewer, PVC, ef9c7314
RainSewer, PVC, f7a47957
WasteSewer, Concrete, 0a11f144
WasteSewer, Concrete, 1fc1be95
WasteSewer, Concrete, 2448b14b
WasteSewer, Concrete, 61fa20ca
WasteSewer, Concrete, a8f7538a
WasteSewer, Concrete, dcc31806