Automation in Construction 144 (2022) 104614

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Multi-scale hybrid vision transformer and Sinkhorn tokenizer for sewer defect classification

Joakim Bruslund Haurum [a,b,*], Meysam Madadi [c], Sergio Escalera [a,b,c,d], Thomas B. Moeslund [a,b]

[a] Visual Analysis and Perception Laboratory, Aalborg University, Denmark
[b] Pioneer Centre for Artificial Intelligence, Denmark
[c] Computer Vision Center, Autonomous University of Barcelona, Spain
[d] Department of Mathematics and Informatics, Universitat de Barcelona, Spain

ARTICLE INFO

Keywords:
Sewer Defect Classification
Vision Transformers
Sinkhorn-Knopp
Convolutional Neural Networks
Closed-Circuit Television
Sewer Inspection

ABSTRACT

A crucial part of image classification consists of capturing non-local spatial semantics of image content. This paper describes the multi-scale hybrid vision transformer (MSHViT), an extension of the classical convolutional neural network (CNN) backbone, for multi-label sewer defect classification. To better model spatial semantics in the images, features are aggregated at different scales non-locally through the use of a lightweight vision transformer, and a smaller set of tokens was produced through a novel Sinkhorn clustering-based tokenizer using distinct cluster centers. The proposed MSHViT and Sinkhorn tokenizer were evaluated on the Sewer-ML multi-label sewer defect classification dataset, showing consistent performance improvements of up to 2.53 percentage points.

1. Introduction

The sewerage infrastructure is one of a few critical infrastructures in modern society. If the infrastructure does not function properly, it can lead to dramatic environmental damage and pose a risk to the public health [1]. Therefore, the sewer pipes require regular inspections in order to determine when a pipe has to be replaced or rehabilitated. However, with more than 1.2 million kilometers of public sewerage infrastructure in just the U.S. [1], this becomes an unimaginable task to perform manually on a regular basis, as each inspection has to be performed by a professional sewer inspector. Therefore, the task of automating the sewer inspection process has been researched for more than three decades, through the development and application of sensors and computer vision algorithms [2–5].

Since its adoption in 2017, the Convolutional Neural Network (CNN) has been the method of choice within the automated sewer inspection domain [2]. A key component of the CNN is the convolutional layers, which efficiently model local spatial semantics within the image. However, for tasks such as multi-label image classification, object detection, and object segmentation, it is essential to model non-local spatial semantics [7]. For example, a displaced joint and intruding roots could be simultaneously in an image but in opposite corners. This represents a case where multi-scale non-local spatial semantics are helpful, as knowing the presence of the displaced joint is a strong signal for inferring the presence of the roots.

Two different approaches have been adopted for vision tasks – either replacing convolutions within the CNN with non-local operations [8,7,9,10] or appending CNNs with non-local operations [11–15], denoted Hybrid Vision Transformer (HViT)-like methods in this paper. However, none of these methods explicitly model non-local spatial semantics across scales for image classification, even though it is used as a common approach in object detection and segmentation. We therefore propose the Multi-Scale Hybrid Vision Transformer (MSHViT), where a Vision Transformer (ViT) [13] is appended at different stages of a CNN backbone for non-local aggregation of features and cross-scale propagation of features. We also introduce the Sinkhorn tokenizer, a clustering-based tokenizer to replace the simple patch based tokenizer in ViTs and act as another source of non-local spatial semantics. Furthermore, we demonstrate that the Sinkhorn tokenizer successfully cluster the CNN features, which are expected to have a high amount of redundant information due to successively applying overlapping convolutional filters and pooling layers. We find that introducing these multi-scale and non-local spatial semantics operations leads to a relative improvement compared to using just the CNN backbone.

In this work we focus on the challenging task of multi-label sewer defect classification, which has been shown by Haurum and Moeslund to

* Corresponding author at: Visual Analysis and Perception Laboratory, Aalborg University, Denmark.
E-mail addresses: joha@create.aau.dk (J.B. Haurum), mmadadi@cvc.uab.es (M. Madadi), sergio@maia.ub.es (S. Escalera), tbm@create.aau.dk (T.B. Moeslund).

https://doi.org/10.1016/j.autcon.2022.104614
Received 25 April 2022; Received in revised form 26 September 2022; Accepted 3 October 2022
Available online 20 October 2022

be an unsolved problem [2,16], highlighting the difficulties of distinguishing visually similar defect classes and poor classification rates of sewer defects with the highest economic impact. Furthermore, improving sewer defect classification performance is crucial for advancing sewer defect detection and segmentation, as such models build upon pre-trained classification models [17].

Our main contributions are as follows:

- We present the Multi-Scale Hybrid Vision Transformer (MSHViT), a novel multi-scale extension of the Hybrid Vision Transformer model for capturing non-local spatial semantics across scales.
- We present the Sinkhorn tokenizer, a novel clustering-based tokenizer using Sinkhorn distances, which reduces the number of tokens and improves metric performance. We visually verify the cross-scale non-local interactions.
- We demonstrate that the MSHViT model outperforms the baseline CNN approaches and other HViT-like approaches on the Sewer-ML multi-label sewer classification dataset, when only considering the defect classification task.
- We demonstrate the applicability of MSHViT and the Sinkhorn tokenizer across the backbones in the ResNet and TResNet CNN architecture families, and thoroughly investigate the impact of each introduced hyperparameter.

The paper is structured as follows. In Section 2, we review the related works within automated sewer inspections, Vision Transformers, non-local CNN blocks, and tokenizers. In Section 3, we introduce MSHViT and the Sinkhorn tokenizer. In Section 4, we determine the improvement obtained by introducing the MSHViT and Sinkhorn tokenizer and compare to other HViT-like approaches. In Section 5 we conduct an extensive ablation study of the proposed methods. In Section 6 we qualitatively investigate the clustering assignment made by the Sinkhorn tokenizer, and in Section 7 we discuss the limitations and practical use of the proposed method. Finally, in Section 8, we conclude the paper.

2. Related works

In this section we review the literature within the automated sewer inspection domain, as well as recent progress within Vision Transformers, non-local CNN blocks, and tokenization approaches.

2.1. Automated sewer inspections

The automated sewer inspection research field has been active for more than three decades, developing domain-specific computer vision algorithms to handle the unique environment that is the sewerage infrastructure [2]. However, Haruum and Moeslund [2] found that the research field has been hindered by the lack of open source code and data, which in combination with differing evaluation protocols, has made it extremely difficult to compare the proposed methods in the literature and caused the field to lag behind the general computer vision domain. This has been rectified for the classification tasks with the introduction of the public Sewer-ML dataset [16], enabling fair and open comparisons of multi-label classification approaches. Using the Sewer-ML dataset Haurum and Moeslund showed that the sewer defect classification tasks are far from solved, comparing the leading sewer defect classification methods from Kumar et al. [18], Meijer et al. [19], Xie et al. [20], Chen et al. [21], Hassan et al. [22], and Myrans et al. [23]. Concurrent research directions in the sewer defect classification subfield have focused on the usage of StyleGAN-based approaches to increase the effective size of small training dataset [24,25], developing and deploying networks on embedded devices [26,27], and providing defect localization information without explicit localization labels [28,29].

However, the main focus of the field within recent years has been on the defect detection and segmentation tasks [30–32,17,33–36], where no public datasets are available. The field has, however, become more transparent as many have started to directly compare different methods on the same datasets, in an effort to offset the lack of public detection and segmentation datasets [17,36,34]. Recently, the field has also started investigating other parts of the sewer inspection process [30,32,17,37–41], such as Haurum et al. [37] proposing a multi-task classification approach for simultaneously classifying defects, water level, pipe material, and pipe shape, and Wang et al. [30] proposed a framework to accurately determine the severity of defects related to the operation and maintenance of the pipes. The field has also adopted recent trends from the general computer vision field such as self-supervised learning [39], synthetic data generation [25,24,42–44], neural architecture search [45], and usage of the Transformer architecture [17,46], indicating that the automated sewer inspection field is catching up to the general computer vision domain.

2.2. Vision transformers

Transformers were originally developed for Natural Language Processing (NLP) [47]. Dosovitskiy et al. [13] demonstrated how a pure Transformer based architecture, denoted Vision Transformer (ViT), led to competitive performance on several vision classification tasks. The ViT architecture has led to an increased research focus on adapting Transformers for vision tasks [48–58]. A general trend has been introducing components from CNNs into the ViTs, such as limited region of interests and hierarchical representations [53,50,54,55] or extending CNNs with Transformers in a hybrid approach [13,15,48,12]. However, unlike CNNs the ViT only processes the input image on a single scale due to the initial tokenization step and the absence of pooling operations. This problem has been approached in two ways, by introducing either hierarchical representations inspired by classical CNN architecture design [53–56] or multi-scale representations by applying different ViTs sequentially [59] or working on variations of the input in parallel [60,58]. Our proposed method differs fundamentally from the prior work as we introduce multi scale features by combining CNNs and ViTs, instead of adapting a purely ViT-based model.

2.3. Non-local CNN blocks

Combining non-local blocks and operations with classical CNNs have been of great interest as a way of capturing global spatial semantics. The Non-Local Network (NLN) [8] was proposed as an extension of the ResNet architecture family, where non-local aggregation operations were inserted into the last blocks of the architecture. The NLN architecture was extended by Srinivas et al. [7] who introduced the Bottleneck Transformer, where Multi-Head Self-Attention was inserted directly into the ResNet bottleneck blocks. Both of these approaches lead to direct improvements on several vision tasks. Appending CNNs with non-local operations have similarly lead to improvements in image classification as shown by Dai et al. [14] who investigated how to design Hybrid Vision Transformers (HViTs), *i.e.* CNNs appended with a ViT, and in tasks such as object detection with the DETR model [11] and enabling image-caption pair based training [15]. In contrast to the previous application of non-local blocks, we append the CNN at several stages in order to explicitly introduce multi-scale interactions through the proposed MSHViT architecture.

2.4. Tokenizers

An essential part of the Transformer architecture is the choice of how to generate the token embeddings. In NLP several embedding methods have been utilized through the years in order to represent sentences and words [61,62]. However, for image data this has not been the case. Dosovitskiy et al. [13] proposed simply extracting non-overlapping patches of the input image and linearly map the patches to an

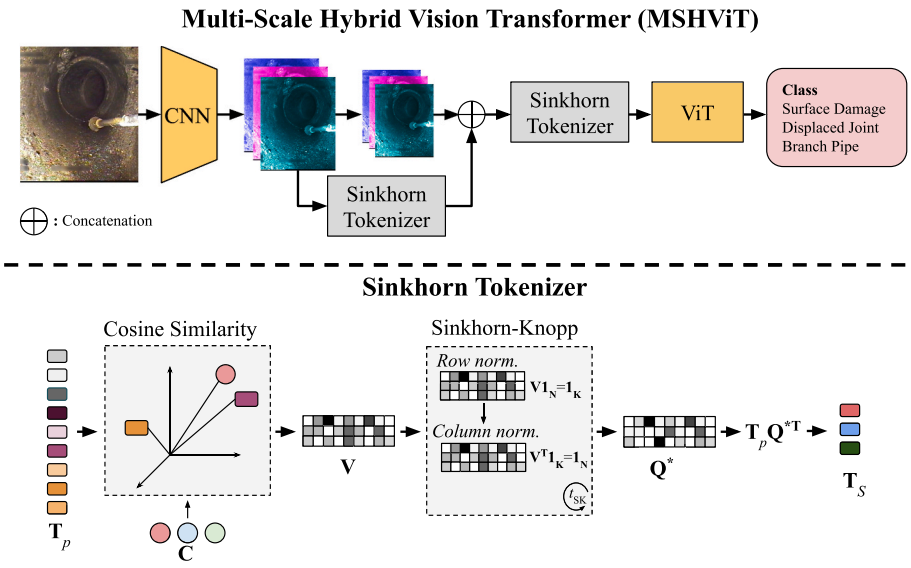


Fig. 1. System overview. (Top) A CNN backbone returns feature maps from a subset of the internal scales in the CNN. The feature maps from each scale are first tokenized and then processed by a weight-shared ViT. The information from previous scales are propagated forward to the next scale, shown in the figure by forwarding the Sinkhorn tokenizer output to the next scale as per Eq. (9). (Bottom) The Sinkhorn tokenizer reduces the number of tokens by first measuring the cosine similarity, $\mathbf{V}$, between all input tokens $\mathbf{T}_p$ and cluster centers $\mathbf{C}$. The Sinkhorn distances [6] are then computed by applying Sinkhorn-Knopp for t_{SK} iterations, resulting in the soft assignment matrix, $\mathbf{Q}^{*\top}$. Using $\mathbf{Q}^{*\top}$ the input features are clustered into the smaller set of tokens, $\mathbf{T}_S$.

embedding space. This approach has since been iterated upon, by instead extracting overlapping patches [57], learning to select the patch size of the conventional patch tokenizer [63], as well as replacing the initial layer of the Transformer with a convolutional stem similar to those found in CNNs [49]. In parallel, different token downsampling approaches have been investigated in order to reduce token redundancy. Goyal et al. [64] and Rao et al. [65] propose score-based token downsampling methods, where each token is assigned a score based on the incoming attention from other tokens or a predictive subnetwork, respectively. In contrast, this work and the concurrent work by Marin et al. [66] propose clustering based approaches for reducing the number of tokens. The method by Marin et al. utilizes a K-means/medoids based approach, whereas our proposed Sinkhorn tokenizer utilizes Sinkhorn distances [6] in order to softly assign the input tokens to a set of cluster centers. All of the prior approaches [64–66] are focused on pure ViT architectures and inserted in between each encoder block progressively decimating the number of tokens present. Comparatively, the proposed Sinkhorn tokenizer is applied on HViTs in order to reduce redundancy in the CNN feature-based tokens.

3. Methodology

In this section we first review the Vision Transformer and its hybrid variant originally proposed by Dosovitskiy et al. [13]. Then we present our novel clustering-based Sinkhorn tokenizer, designed to reduce the number of redundant tokens in ViTs. Lastly, we present our MSHViT architecture, designed to non-locally combine CNN features at the ith scale and progressively combine features across scales, as illustrated in Fig. 1. An overview of the introduced symbols and notations can be found in Appendix A.

3.1. Vision transformers

The Vision Transformer [13] demonstrated that the original Transformer architecture [47] can be used with little modifications for image classification, and without the image-related inductive biases found in CNNs.

3.1.1. Tokenization

The Transformer takes a series of 1D token embeddings as input, and process the series in parallel. In order to convert image data to a series of 1D tokens the input image $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$ is convolved with D different $P \times P$ kernels with a stride of P and flattend to a 1D vector per patch, producing $N = HW/P^2$ linearly embedded tokens $\mathbf{T}_p \in \mathbb{R}^{D\times N}$.

Furthermore, a special class (CLS) token $\mathbf{x}_{\text{CLS}} \in \mathbb{R}^D$ is appended to $\mathbf{T}_p$. The CLS token is randomly initialized and used to generate an image-level feature representation. In order to encode a spatial ordering into the tokens a learnable positional embedding $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{D\times N+1}$ is added, leading to the final token representations:

$$\mathbf{Z}_0 = [\mathbf{x}_{\text{CLS}} \parallel \mathbf{T}_p] + \mathbf{E}_{\text{pos}}, \tag{1}$$

where $\parallel$ denotes concatenation.

3.1.2. ViT model

The Transformer consists of L stacked encoder blocks, each consisting of a token-aggregation step, such as Multi-Head Self-Attention (MHSA), followed by an inverted bottleneck projecting each token into an intermediate $\mathbb{R}^{D\cdot r}$ space, where r is an adjustable hyperparameter, followed by a down projection to the D-dimensional feature space. Layer normalization (LN) [67] is applied before both actions and residual connections are inserted around each action. The final feature representation is the CLS token after L blocks and a final layer normalization step, $\mathbf{y} = \text{LN}(\mathbf{Z}_{L,0})$.

3.1.3. Hybrid ViT

Unlike CNNs, ViTs have very little image-specific inductive biases [13]. Therefore, ViTs often require large amount of training data in order to learn relevant relations, which are encoded directly into CNN architectures. However, this lack of inductive biases similarly allows ViTs to learn relations within images, which are not viable with CNNs, such as capturing non-local spatial semantics. The HViT aims at combining these two architectures, by first using a CNN to encode local features, and then compute non-local spatial semantics using a ViT. This is realized by extracting the tokens $\mathbf{T}_p$ from a CNN feature map with a kernel size $P = 1$, typically at the last feature map before the commonly used global pooling step. This is in contrast to the ViT model where the tokens are extracted directly from the input image $\mathbf{X}$.

3.2. Sinkhorn tokenizer

The original ViTs generate the token representations of the image through a non-overlapping patch based method [13]. Several methods have been proposed to improve the tokenizer either by reducing the stride of the convolutional layer such that the patches overlap [57], or instead use a convolutional stem which aggressively downsamples the

spatial dimensions of the input [49]. However, these methods do not consider the redundancy of features stemming from encoding similar patches in the image and therefore lead to disproportionately representing these in the generated tokens. While this may be implicitly handled by the attention mechanisms in the ViT, it introduces an unnecessary processing overhead and requires the model to learn these relations.

To deal with the redundant features we introduce a clustering-based tokenizer using Sinkhorn distances [6], inspired by clustering-based self-supervised learning [68,69]. The approach builds upon the original patch tokenizer with $P = 1$. The N patch tokens $\mathbf{T}_p$ are compared to K cluster centers $\mathbf{C} \in \mathbb{R}^{D\times K}$ which are initialized from a D-dimensional Normal distribution with zero-mean and unit variance. We assume both $\mathbf{T}_p$ and $\mathbf{C}$ are ℓ_2 normalized and measure similarity using the cosine similarity $\mathbf{V} = \mathbf{C}^\top \mathbf{T}_p \in \mathbb{R}^{K\times N}$. Based on the similarity scores $\mathbf{V}$ we compute the soft assignment matrix $\mathbf{Q} \in \mathbb{R}_+^{K\times N}$, which belongs to the set of valid assignment matrices Q, such that the similarity between the cluster centers and features is maximized:

$$\max_{\mathbf{Q}\in Q} \operatorname{Tr}(\mathbf{Q}^\top \mathbf{V}) + \epsilon H(\mathbf{Q}), \tag{2}$$

where H is the matrix entropy function and ϵ controls the weighting of the entropy loss and thereby the smoothness of the assignment scores.

Similar to [68,69] we constrain $\mathbf{Q}$ to be in the transportation polytope under an equipartition constraint of the input and cluster centers *i.e.* the features should on average be uniformly assigned to the cluster centers. However, instead of applying the constraint on the full dataset [68] or mini-batches [69], we apply the constraint on the N features from a single input, see Eq. (3). We apply the constraint on the N features such that there is no cross-information between input images, enabling single image evaluation.

$$Q = \{\mathbf{Q} \in \mathbb{R}_+^{K\times N} | \mathbf{Q}\mathbf{1}_N = \frac{1}{K}\mathbf{1}_K, \mathbf{Q}^\top \mathbf{1}_K = \frac{1}{N}\mathbf{1}_N\}, \tag{3}$$

where $\mathbf{1}_K$ and $\mathbf{1}_N$ are K and N-dimensional vectors filled with ones, respectively.

The solution to Eq. (2) can then be formulated as follows:

$$\mathbf{Q}^* = \operatorname{diag}(\mathbf{u}) \exp\left(\frac{\mathbf{V}}{\epsilon}\right) \operatorname{diag}(\mathbf{v}) \in \mathbb{R}_+^{K\times N}, \tag{4}$$

where the renormalization vectors $\mathbf{u}$ and $\mathbf{v}$ are computed using the iterative Sinkhorn-Knopp algorithm [6] through t_{SK} iterations.

Using the soft assignments between input features $\mathbf{T}_p$ and cluster centers $\mathbf{C}$ stored in $\mathbf{Q}^*$ we transform the input features into K new tokens:

$$\mathbf{T}_S = \mathbf{T}_p \mathbf{Q}^{*\top} \in \mathbb{R}^{D\times K} \tag{5}$$

3.3. Multi-scale hybrid vision transformers

Based on prior work on combining non-local operations with classical CNNs, such as HViTs, we propose the Multi-Scale Hybrid Vision Transformer. Whereas the original HViT simply extends the backbone CNN with a ViT, we propose applying ViTs at different scales of the backbone CNN. Furthermore, we also introduce cross-scale connections between the ViTs in order to encode non-local spatial semantics in the image at different scales, see Fig. 1.

CNNs such as ResNets [70] and Inception networks [71,72] have a set of natural scales within them due to the periodic pooling operations. The representative feature map of each scale is defined to be the last feature map before each pooling operation and denoted $\mathbf{X}^i$ for the ith scale. At every scale each feature in $\mathbf{X}^i$ is linearly embedded into a common D-dimensional space as tokens $\mathbf{T}_p^i$. These tokens are processed using a tokenization function ψ^i, representing either the Sinkhorn tokenizer (Eq. (5)) or an identity function for the standard patch tokenizer, with the output denoted $\mathbf{T}^i$. The tokens can then be processed by a scale-specific ViT of depth L, denoted as ϕ^i, producing the scale features:

$$\mathbf{Z}_L^i = \phi^i(\mathbf{T}^i) \tag{6}$$

3.3.1. Cross-scale connections

In order to share information between different scales, we introduce cross-scale connections. For scale $i > 1$, all previous scale features, or a subset of the features, are included, denoted $\mathbf{S}^i$, in addition to the ith scale features $\mathbf{T}_p^i$, see Eq. (7).

$$\mathbf{T}^i = \psi^i(\mathbf{T}_p^i \parallel \mathbf{S}^i) \tag{7}$$

This cross-scale connection can occur using features from three different stages: the linearly embedded CNN features $\mathbf{T}_p$, see Eq. (8), the Sinkhorn tokens $\mathbf{T}_S$, see Eq. (9), or the final token embeddings $\mathbf{Z}_L$, see Eq. (10). j denotes the initial scale which we consider for scale i. For example, if $j = 1$ all features from scale 1 to scale $i-1$ are aggregated, while if $j = i-1$ only the features from scale $i-1$ are aggregated.

$$\mathbf{S}^i = \|_j^{i-1} \mathbf{T}_p^j \tag{8}$$

$$\mathbf{S}^i = \|_j^{i-1} \mathbf{T}_S^j \tag{9}$$

$$\mathbf{S}^i = \|_j^{i-1} \mathbf{Z}_L^j \tag{10}$$

Lastly, the overall image representation is defined to be $\mathbf{y} = \mathrm{LN}(\mathbf{Z}_{L,0}^I)$, where I denotes the last scale of the backbone.

Table 1
Detailed training procedures. We follow the training procedures of Haurum et al. [37] with the addition of utilizing model EMA.

Variable	Value
Image Size	224
Epochs	40
Batch Size	256
Learning Rate (LR)	0.1
Weight Decay	0.0001
LR Scheduler	Step @ 20, 30 epochs
LR Decay Factor	0.01
Optimizer	SGD w/ momentum
Loss function	Binary Cross-Entropy
Class Weighting	Effective samples [74] $\beta = 0.9999$
Model EMA	0.9997
Augmentations	Horizontal flip ($p = 0.5$) Color Jitter (± 0.1)

4. Experimental results

In this section we investigate the performance of the MSHViT architecture and Sinkhorn tokenizer on the Sewer-ML dataset, a multi-label sewer defect classification dataset [16]. Sewer-ML is the world's only public multi-label sewer defect dataset, consisting of 1.3 million images, 17 defect classes, and the implicit normal class. The dataset is split into three distinct training, validation, and testing splits, each containing 1 million, 130 k and 130 k images, respectively. We refer to the Supplementary material of Haurum and Moeslund [16] for example images. Defect predictions are evaluated using the class F2-scores weighted by the *class importance weights* (CIW), $\mathrm{F2}_{\mathrm{CIW}}$, which indicates the economic importance of the classes, and the normal pipes are evaluated by the F1-score, $\mathrm{F1}_{\mathrm{Normal}}$ [16]. An abbreviated introduction to the Sewer-ML dataset and the evaluation metrics can be found in Appendix B. Code and model weights can be found at the project webpage: https://vap.aau.dk/mshvit/.

Table 2
Hyperparameters. Overview of all searched hyperparameters, with the investigated values as well as the initial and final values.

HP	Range	Initial	Final
ϵ	[0.05, 0.25, 0.5, 0.75, 1.00, 1.25]	0.05	0.25
t_{SK}	[1, 3, 5, 7, 9]	3	5
K	[32, 64, 128, 64/32, 128/64]	64	64
Scales	[{2,3,4,5}, {3,4,5}, {4,5}, {5}]	{4,5}	{4,5}
S	$[\mathbf{T}_p, \mathbf{T}_S, \mathbf{Z}_L]$	$\mathbf{T}_S$	$\mathbf{T}_S$
j	$[i-1, \min(\text{Scales})]$	$i-1$	$i-1$
D	[512, 1024, 2048]	512	512
r	[1, 2, 3, 4]	4	4
L	[1, 2, 3]	2	2

4.1. Training procedure

We follow the training procedure of Haurum et al. [37] with the addition of using the Exponential Moving Average (EMA) technique on the model weights, see Table 1. We utilize the Fourier Network (FNet) based attention mechanism [73] in the HViT as an efficient alternative to the conventional MHSA based attention mechanism.

We define the ResNet architecture to have five natural scales: the convolutional stem followed by four ResNet blocks, numbered from 1 to 5. These stages are chosen as they act on feature maps with different spatial dimensions.

4.2. Hyperparameter search

The hyperparameter search for the MSHViT and Sinkhorn tokenizer is conducted in a sequential manner in order to reduce the search space due to the number of hyperparameters and the investigated value ranges. The investigated hyperparameter values as well as the initial and final values are shown in Table 2. The initial Sinkhorn Tokenizer values were set as in Caron et al. [69], except for the number of clusters K, where we chose 64 centers as the initial value to ensure a large average assignment probability per cluster in each scale. For the MSHViT architecture we initialized the model by appending the last two layers, where higher-order features are available. The hyperparameters of the ViTs were chosen such that only a moderate parameter increase was introduced. After each step in the sequential search we used the configuration which performed the best for the next step. The steps of the sequential search were ordered such that the Sinkhorn Tokenizer cluster and MSHViT cross-scale hyperparameters were determined, and lastly the structure of the ViTs. The entire hyperparameter search was conducted with the ResNet-50 backbone. The order of the search was as follows:

1. Search over the entropic regularization ϵ in the Sinkhorn tokenizer.
2. Search over the number of iterations t_{SK} in the Sinkhorn tokenizer.
3. Search over the number of clusters K in the Sinkhorn tokenizer.
4. Search over which scales to be used and selection of j in the MSHViT extension.
5. Search over the multi-scale method, **S**.
6. Search over token dimensionality D.
7. Search over the MLP ratio r.
8. Search over ViT depth L.

We find that the initial hyperparameters perform well, with only the entropic regularization and number of iterations in the Sinkhorn-Knopp algorithm being adapted.

4.3. Comparative models

We investigate the performance increase incurred when applying MSHViT to the ResNet-{18, 34, 50, 101}, a commonly used backbone architecture in the image classification literature [7,12,76], as well as TResNet backbone [75], an adaption of the ResNet backbone using concepts such as anti-aliased downsampling and Squeeze and Excitation (SE) [77] layers. The same MSHViT hyperparameters are used for all backbones. Furthermore, we compare performance against the HViT-like models BoTNet-50-S1 [7] and CoAtNet-{0, 1} [14], as well as the original HViT structure [13]. BotNet and CoAtNet were trained with the model structure described in the original papers, while the HViT model uses the same ViT parameters described in Table 2 with the exception of the attention mechanism where we use the classical MHSA-based token mixing. We compare using both the conventional patch based tokenizer and the proposed Sinkhorn tokenizer. Lastly, we compare to the previously published results on Sewer-ML [16,37]. We run all experiments within the same codebase, using the torchvision [78], Pytorch Lightning [79] and timm [80] libraries. All models were trained using a single Nvidia V100 GPU except for the CoAtNet models which required two V100 GPUs due to a higher VRAM consumption.

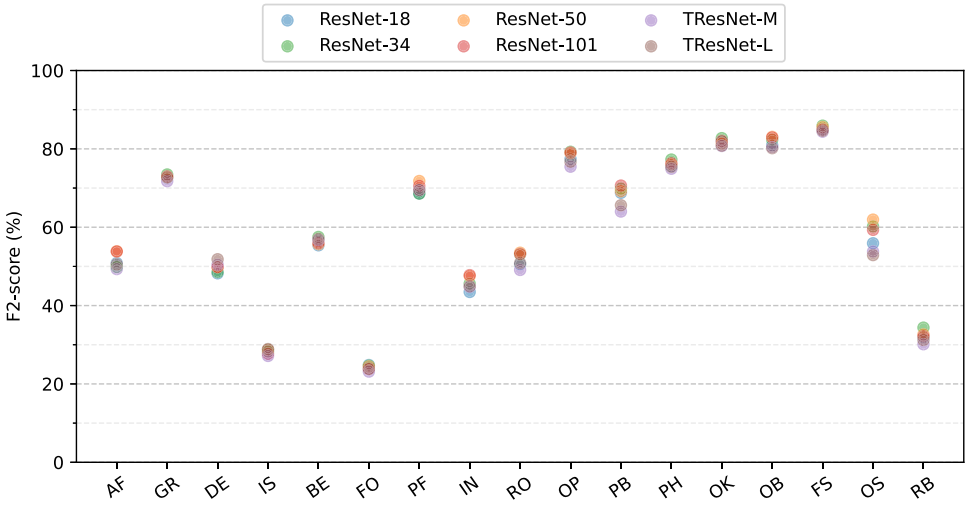
Table 3
Results on Sewer-ML. Comparison using the investigated CNN backbones. We compare each backbone with and without the MSHViT and Sinkhorn tokenizer extension (denoted MSHViT) using the $F2_{CIW}$ and $F1_{Normal}$ metrics [16]. Best performance per column is denoted in **bold**. We also include the previous published results on Sewer-ML [16,37], and HViT-like models [7,13,14]. *denotes that the method was trained in a multi-task classification framework.

		Validation Split		Test Split	
Model	**MSHViT**	$F2_{CIW}$	$F1_{Normal}$	$F2_{CIW}$	$F1_{Normal}$
Benchmark[16]	-	55.36	91.32	55.11	90.94
CT-GAT* [37]	-	**61.70**	91.94	**60.57**	91.61
ResNet-50-HViT-Patch [13]	-	59.87	92.41	57.58	91.99
ResNet-50-HViT-Sinkhorn [13]	-	60.42	92.41	58.74	92.07
BotNet-50-S1 [7]	-	61.62	**92.92**	59.69	**92.49**
CoAtNet-0 [14]	-	57.82	92.28	56.53	91.94
CoAtNet-1 [14]	-	59.37	92.50	57.42	91.11
ResNet-18 [70]	×	58.60	92.34	56.62	91.88
	✓	59.87	92.42	58.18	92.12
ResNet-34 [70]	×	60.98	92.72	59.18	92.30
	✓	61.65	92.76	59.91	92.30
ResNet-50 [70]	×	59.28	92.44	57.58	92.03
	✓	61.68	92.44	60.11	92.11
ResNet-101 [70]	×	60.06	92.48	58.01	92.13
	✓	61.25	92.50	59.93	92.19
TResNet-M [75]	×	58.04	92.22	56.08	91.90
	✓	58.68	92.25	56.93	91.84
TResNet-L [75]	×	59.17	92.36	56.97	92.00
	✓	59.19	92.27	57.16	91.87

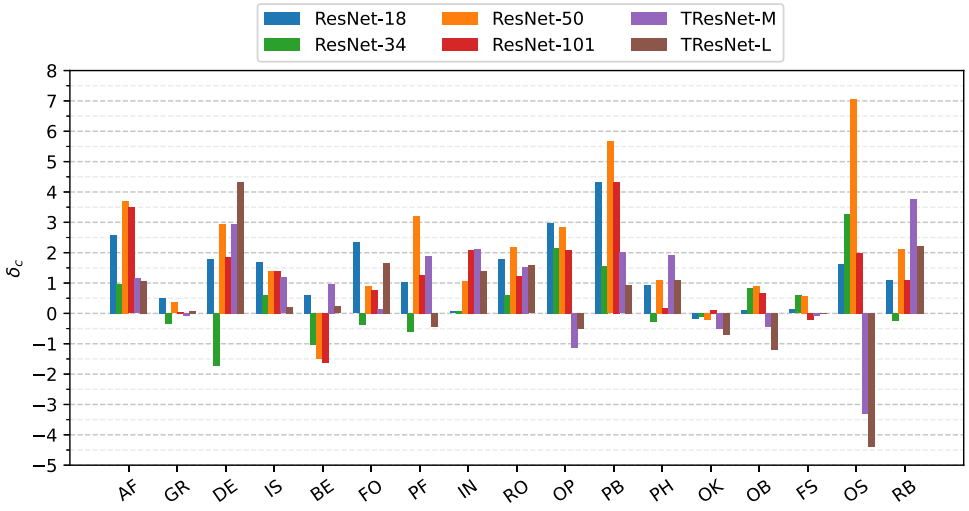
4.4. Results

We find that introducing the MSHViT and Sinkhorn Tokenizer leads to a noticeable improvement on all tested backbones, see Table 3. On the $F2_{CIW}$ metric we observe an increase between 0.7 and 2.5 percentage points, with the largest increase observed on the ResNet-50, where the performance is improved by 2.4–2.5 percentage points on both the validation and testing splits. This is significantly better than the benchmark algorithm from Haurum and Moeslund [16], and a comparable performance to the previous best performing model on Sewer-ML, the multi-task classification method CT-GAT [37], while only using the sewer defect labels during training. This demonstrates that it is possible to significantly increase the sewer defect classification performance without needing auxiliary data such as water level, pipe shape, and pipe material. For the non-defective pipes we observe a more moderate increase of up to 0.24 percentage points in the $F1_{Normal}$ metric. However, we observe a higher baseline performance compared to previous methods.

Interestingly, we observe that the ResNet-34 backbone perform



(a) Per-class performance for all MSHViT models.



(b) Per-class difference when comparing the MSHViT and baseline models.

Fig. 2. Per-Class F2-scores analysis. We present the per-class F2-scores on the validation split for all MSHViT-based models as well as the difference between the MSHViT variants and the baseline models, δ_c. The classes are sorted in ascending order by their class-importance weight [16]. Class names and abbreviations are described in Appendix B.

surprisingly well for both the baseline and MSHViT extension. Not only does the ResNet-34 baseline achieve the best performance out of the ResNet networks, it also either outperforms or matches the ResNet-101 backbone when applying the MSHViT extension. For the TResNet architectures we observe that the improvement gained by adding MSHViT extension is smaller than that for the ResNet backbones. This is most likely due to the SE layers in the TResNet model, which means the TResNet already includes some attention-based mechanisms. However, it is clear that the MSHViT extension is still beneficial.

When comparing to other HViT-like models we see that the MSHViT extension outperforms the original HViT structure, as well as all models where the Transformer structure is incorporated directly into the backbone. It should be noted that on the validation split the BotNet-50-S1 model nearly matches the ResNet-50-MSHViT's $F2_{CIW}$ score and achieves the highest $F1_{Normal}$ performance. However, on the test split the $F2_{CIW}$ performance is significantly lower compared to the ResNet-50-MSHViT, indicating the model does not generalize as well as the ResNet-50-MSHViT model.

From these results we can conclude that the proposed MSHViT extension led to improvements without tuning the hyperparameters for the backbone. We hypothesize that if hyperparameters were tuned for each backbone, the performance gain would further increase.

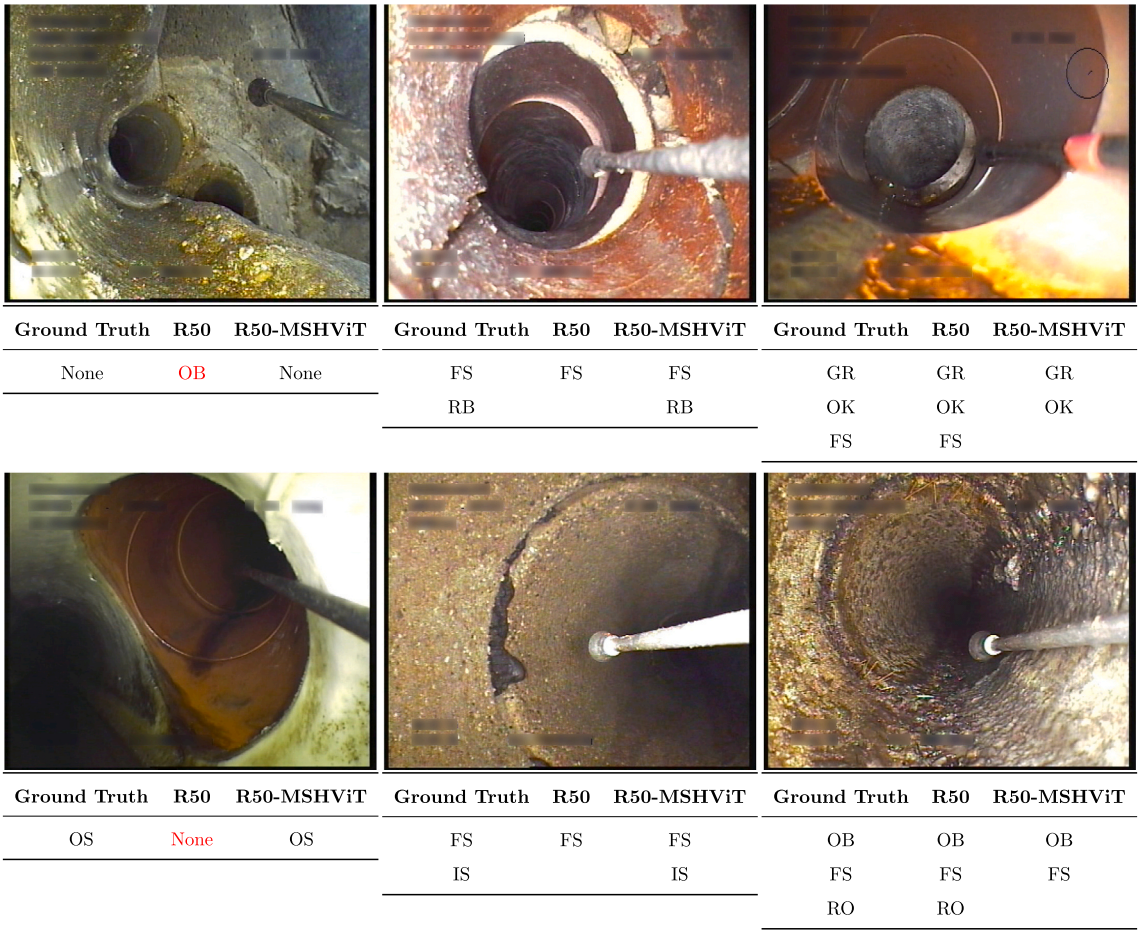
4.5. Per-class analysis

In order to better understand how the compared models work, we investigate how the baseline and MSHViT extended models differ in their class predictions on the validation split. In Fig. 2a we present the per-class F2-scores for all MSHViT models, and in Fig. 2b we determine the difference in per-class F2-scores when comparing the MSHViT variants with the baseline models, see Eq. (11).

$$\delta_c = c_{\text{MSHViT}} - c_{\text{Baseline}}, \tag{11}$$

where δ_c is the difference in F2-scores for class c, and c_{MSHViT} and c_{Baseline} are F2-scores for class c for the MSHViT and Baseline models, respectively.

When analyzing the absolute per-class performance in Fig. 2, we see that the ResNet-34, ResNet-50, and ResNet-101 all perform similarly well on nearly all classes, with the ResNet-34 and ResNet-50 achieving noticeable performances in the highest weighted classes, whereas the



Ground Truth	R50	R50-MSHViT
None	OB	None

Ground Truth	R50	R50-MSHViT
FS	FS	FS
RB		RB

Ground Truth	R50	R50-MSHViT
GR	GR	GR
OK	OK	OK
FS	FS	

Ground Truth	R50	R50-MSHViT
OS	None	OS

Ground Truth	R50	R50-MSHViT
FS	FS	FS
IS		IS

Ground Truth	R50	R50-MSHViT
OB	OB	OB
FS	FS	FS
RO	RO	

Fig. 3. Examples of classifications with MSHViT. Example cases where the MSHViT model correctly classifies all classes as well as misclassifies some classes. The class codes are described in the original Sewer-ML paper [16]. Incorrect predictions are shown in red.

TResNet models and ResNet-18 have a noticeably lower score on several classes. We also observe that for all models the performance is low on intruding sealing material (IS), the obstacles (FO), cracks, breaks, and collapses (RB). This can be explained by the fact that the IS and FO classes are some of the most rare classes in the Sewer-ML dataset, with less than 10,000 examples per class. Additionally, all three classes have a large variation in their visual appearance within the class, while being less visually distinct from other classes. For example, the FO class is defined such that it encompasses any possible foreign objects that can block the pipes. In Fig. 2b we observe that when using MSHViT together with the ResNet backbones performance increases on nearly all classes, except for consistent decreases on the attached deposits (BE) class and on the connection with construction changes (OK) class. For the ResNet-34 backbone we also observe a significant decrease in performance on the deformation (DE) class. However, there is a noticeable increase in performance on both the lateral reinstatement cuts (OS) and cracks, breaks, and collapses (RB), the two highest weighted classes, across all ResNet backbones. On the other hand we see that the TResNet backbones behaves very poorly on the OS class, which drags down the overall score, even though it performs well on nearly all other classes.

4.6. Qualitative examples

In addition to quantiative per-class comparison, we also look into specific cases where the predictions of the compared models differ. Focusing on the ResNet-50 backbone we compare cases where the MSHViT extensions match all classes correctly while the baseline misclassifies some or all classes and vice versa, see Fig. 3. Four examples are shown where the MSHViT model correctly predicts all classes. In the top left image, the MSHVIT correctly predicts the pipe to be normal, whereas the baseline predicts surface damage (OB). This is most likely due to the missing top half of the pipe, as the image is taken from within the sewer well. In the top middle and bottom left cases the baseline misses the cracks, breaks, and collapses (RB) and lateral reinstatement cuts (OS) classes, the two highest weighted classes by CIW. Missing these classes could lead to significant economic repercussions. The RB class is most likely missed due to its visual similarity to the displaced joint (FS) deeper in the pipe, whereas the OS is similarly missed as the baseline misses the fact that a lining has been inserted and the low severity of the class. In the bottom middle example, the baseline simply misses the intruding sealing material (IS) class, instead only classifying the displaced joint (FS). In the top right and bottom right, the MSHViT variant misses the displaced joint (FS) and roots (RO), respectively. It is not clear why the MSHViT missed the displaced joint, however, we hypothesize it might be due to the co-occurring connection with construction changes (OK) class, where the material of the pipe changes. For the bottom right case, the MSHVIT misses the small fine roots in the joint, most likely due to focusing on the much more prevalent displaced joint (FS) and surface damage (OB).

4.7. Efficiency analysis

In order to determine the efficiency of the MSHViT extension and verify that the increased metric performance is not simply due to an increase in learnable parameters, we compare the validation $F2_{CIW}$ against the number of trainable parameters in the models as well as the

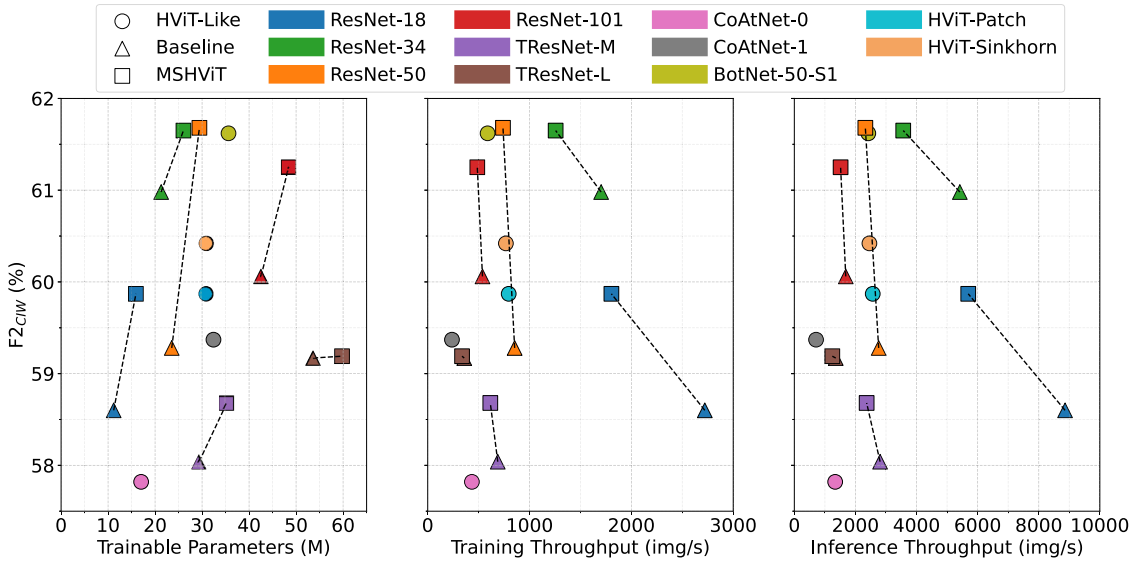


Fig. 4. Comparison of metric performance and efficiency. We compare the performance of the models in Table 3 against the parameter count of each model as well as the throughput of the models in images per second (img/s) during training and inference. MSHViT variants are linked to their baseline variant by a dotted line.

Table 4
Effect of ϵ. Comparison of different entropic regularization values in the Sinkhorn tokenizer.

ϵ	$F2_{CIW}$	$F1_{Normal}$
0.05	60.80	**92.56**
0.25	**61.68**	92.44
0.50	61.33	92.47
0.75	60.85	92.35
1.00	60.86	92.51
1.25	60.46	92.36

Table 5
Effect of t_{SK}. Comparison of number of iterations in the Sinkhorn tokenizer.

t_{SK}	$F2_{CIW}$	$F1_{Normal}$
1	61.16	**92.58**
3	61.24	92.47
5	**61.68**	92.44
7	61.13	92.50
9	61.45	92.47

throughput measured in images processed per second (img/s) during both training and inference, as recommended by Dehghani et al. [81]. The throughput performance is computed over 200 batches of 256 images with an initial 10 warmup batches, and averaged over five separate runs. As the method from Haurum and Moeslund [16] is a two-stage approach and the method from Haurum et al. [37] is designed for the multi-task classification task, we do not include these in the throughput comparison. The results are shown in Fig. 4. From these results it is clear that the increased performance obtained with the MSHViT extension is not only due to the increase number of parameters, as the extended models consistently outperform baseline variants with a higher number of parameters. When looking at the throughput of the models, we see that the MSHViT does lead to a slower processing speed, however, for the larger models such as ResNet-50 and ResNet-101 this slowdown is marginal at best.

5. Ablation studies

We conduct a series of ablation studies in order to determine the sensitivity to the hyperparameter settings in the Sinkhorn tokenizer and MSHViT architecture. All tests are conducted on the Sewer-ML validation set using a ResNet-50 backbone, with the hyperparameter values stated in Tables 1 and 2 unless otherwise stated.

5.1. Sinkhorn-Knopp hyperparameters

At the heart of the Sinkhorn tokenizer is the iterative Sinkhorn-Knopp algorithm, which is controlled by two hyperparameters: t_{SK} and ϵ. We investigate these hyperparameters' effect on the metric performance one at a time.

First, we investigate the strength of the entropic regularization term in Eq. (2) comparing values of $\epsilon = \{0.05, 0.25, 0.50, 0.75, 1.00, 1.25\}$, see Table 4. We observe that the highest $F2_{CIW}$ and $F1_{Normal}$ are achieved using $\epsilon = 0.25$, a slightly higher entropic regularization term than what has previously been used in the self-supervised training domain [69]. In general, we see that a too high or low entropic regularization negatively affects the $F2_{CIW}$ performance.

Secondly, we investigate the effect of the number of iterations conducted t_{SK}. We compare the performance when setting $t_{SK} = \{1, 3, 5, 7, 9\}$, see Table 5, as well as the effect on efficiency by measuring training and inference img/s, see Fig. 5. We observe that peak performance on both $F2_{CIW}$ and $F1_{Normal}$ is achieved when t_{SK} is set to 5, while too few or too many iterations led to degradation in performance. We also observe a monotonic decrease in throughput when t_{SK} is increased, as expected. When compared to the conventional patch tokenizer we observe that the training throughput and the inference throughput of the Sinkhorn tokenizer beats that of the patch tokenizer at all settings of t_{SK}.

5.2. Number of cluster centers K

A key part of the Sinkhorn tokenizer is the number of clusters K. We investigate the effect of setting $K = \{32, 64, 128, 64/64, 128/64\}$, where x/y denotes x clusters for the 4th scale and y clusters for the 5th scale, see Table 6. We find that increasing or decreasing the number of cluster centers slightly reduced the classification performance, whereas having more clusters for earlier scales dramatically decreased performance. This is hypothesized to be due to the earlier clusters capturing similar semantics, as the larger number of cluster centers allow a less aggressive clustering process.

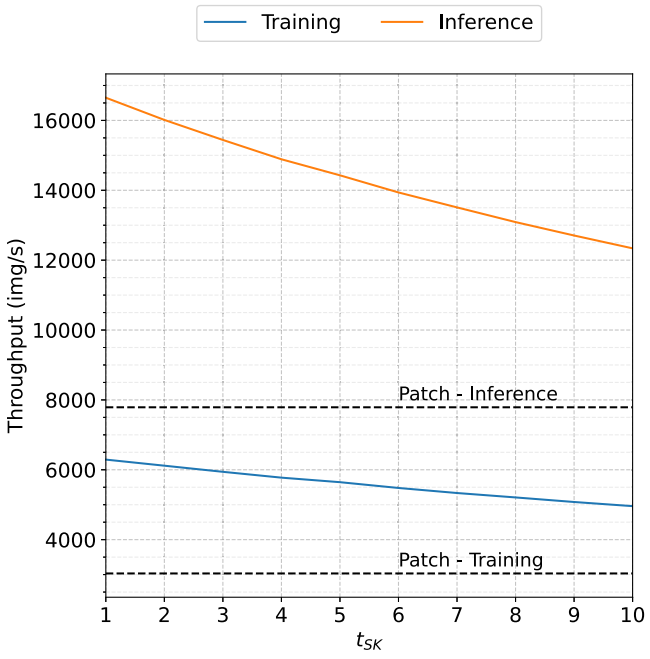


Fig. 5. Effect of t_{SK} on throughput. Comparison of the training and inference throughput at different number of iterations in the Sinkhorn tokenizer, t_{SK}. Training and inference throughput are also shown for the conventional patch tokenizer. Note that the reported throughput differs from Fig. 4, as only the processing time of the MSHViT extension is reported. The backbone processing time has been excluded, as it is simply a constant offset along the y-axis.

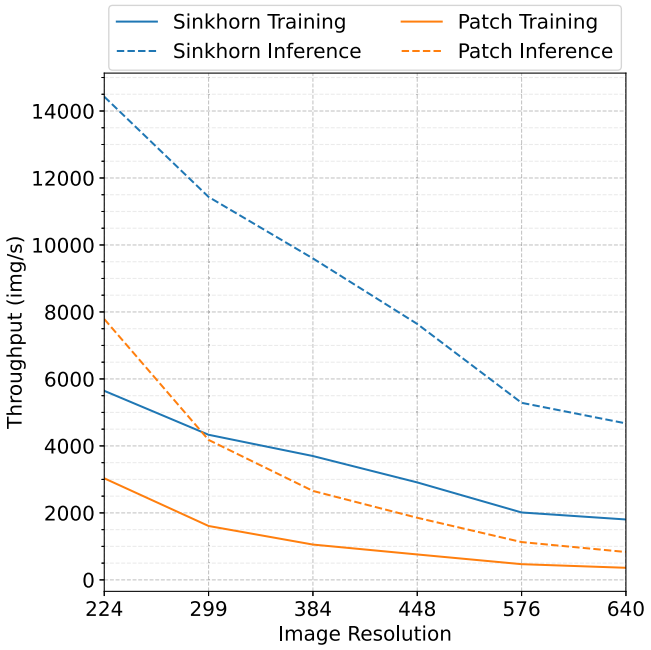


Fig. 6. Effect of image resolution on throughput. We compare the training and inference throughput for the Sinkhorn and patch tokenizers across commonly used image resolutions. The Sinkhorn tokenizer consistently achieves a higher throughput than the conventional patch tokenizer. Throughput is measured only for the MSHViT extension, as the backbone processing time is simply an offset.

Table 6
Effect of number of cluster centers. Comparison of metric performance when varying the number of cluster centers K in the Sinkhorn tokenizer.

K	$F2_{CIW}$	$F1_{Normal}$
32	61.33	92.47
64	**61.68**	92.44
128	61.33	92.34
64/32	60.56	92.46
128/64	60.73	**92.54**

5.3. Tokenizer efficiency at different image resolutions

A key benefit of the Sinkhorn Tokenizer is the constant efficiency when the image resolution is increased. To demonstrate this we compare the training and inference throughput of the MSHViT model (excluding the backbone, which would simply be an offset) at different image resolutions, when using the conventional patch tokenizer and the proposed Sinkhorn tokenizer, see Fig. 6. From this it is clear that the throughput of the Sinkhorn tokenizer better handles the changes in image resolutions, whereas the throughput of the conventional patch tokenizer suffers greatly when the resolution is increased.

5.4. Effect of ℓ_2 normalization

Within the Sinkhorn-Knopp algorithm is the calculation of the cosine similarities between cluster centers and input features, $\mathbf{V}$. This step requires an ℓ_2 normalization of all cluster centers and input features in order to yield output values between -1 and 1. We investigate the effect of skipping this normalization step, see Table 7. We see that the metric performance clearly drops when the features are not normalized onto the unit D-sphere. We can therefore conclude the normalization step is crucial for the Sinkhorn tokenizer.

Table 7
Effect of ℓ_2 normalization. Comparison of performance when ℓ_2 normalizing the cluster centers $\mathbf{C}$ and input features $\mathbf{T}_p$, before computing the similarity scores $\mathbf{V}$.

ℓ_2 **normalized**	$F2_{CIW}$	$F1_{Normal}$
×	60.40	92.40
✓	**61.68**	**92.44**

Table 8
Effect of sharing tokenizer. Comparison of metric performance when sharing tokenizer cluster centers.

Shared Tokenizer	$F2_{CIW}$	$F1_{Normal}$
×	**61.68**	92.44
✓	61.22	**92.53**

5.5. Effect of shared Sinkhorn tokenizer

Inspired by the Perceiver papers [82,83] we investigate the performance when sharing the tokenizer cluster centers and linear projection weights, see Table 8. We find that when sharing the tokenizer parameters, the performance decreases by nearly a half percentage point. This is expected as the same cluster centers have to meaningfully represent CNN features from all considered scales, even though the CNN features are hierarchical in nature.

5.6. Comparison of attention mechanisms and tokenizers

We investigate whether the Sinkhorn tokenizer leads to improvements compared to the standard non-overlapping tokenizer from Dosovitskiy et al. [13], as well as the effect of attention mechanism, see Table 9. Fourier and MHSA refers to the blocks used in the FNet and Transformer models without the per token MLPs, respectively. The

Table 9
Effect of tokenizer and attention mechanism. Comparison of metric performance when using the standard non-overlapping patch tokenizer and the Sinkhorn tokenizer. #P indicates the number of trainable parameters in the MSHViT head in millions.

Attention	Tokenizer	#P	$F2_{CIW}$	$F1_{Normal}$
Fourier	Patch	1.72	59.61	92.46
	Sinkhorn		61.03	92.41
MHSA	Patch	3.82	58.95	92.24
	Sinkhorn		61.09	**92.49**
FNet	Patch	5.92	59.46	92.37
	Sinkhorn		**61.68**	92.44
Transformer	Patch	8.02	59.20	92.38
	Sinkhorn		61.11	92.41

Table 10
Effect of using different scales. Comparison of metric performance when using different scales and different cross-scale sharing range j.

Scales	j	$F2_{CIW}$	$F1_{Normal}$
2, 3, 4, 5	$i-1$	61.36	92.49
3, 4, 5	$i-1$	60.92	92.44
4, 5	$i-1$	**61.68**	92.44
2, 3, 4, 5	2	60.45	92.49
3, 4, 5	3	60.86	92.37
5	-	61.03	**92.52**

Table 11
Comparison of cross-scale mechanisms Comparison of metric performance when using a late-stage scale fusion step or cross-scale mechanism **S** (Eq. (7)) using either CNN (Eq. (8)), Sinkhorn (Eq. (9)), or ViT (Eq. (10)) features.

S	Shared ViT	$F2_{CIW}$	$F1_{Normal}$
-	×	59.88	92.31
-	✓	60.06	92.40
$\mathbf{T}_p$	-	60.25	92.38
$\mathbf{T}_S$	-	**61.68**	92.44
$\mathbf{Z}_L$	×	60.75	**92.49**
$\mathbf{Z}_L$	✓	61.37	92.48

patch based tokenizer uses a kernel size and stride of $P = 1$ for both scales. We observe that the Sinkhorn tokenizer outperforms the conventional patch tokenizer on all attention mechanisms, and that the inverted bottleneck yields little benefit in all cases but the Sinkhorn tokenizer combined with FNet. This shows a clear benefit from the clustering-based Sinkhorn tokenizer.

5.7. Effect of multi-scale approach

In order to determine the effect of the multi-scale approach, we compare the performance when using different scales and the range of the cross-scale connections j. Specifically, we compare using subsets of the scales 2–5 of the ResNet architecture *i.e.* all but the convolutional stem scale, as well as cross-scale connections with $j = i-1$ where only the previous scale is relevant, or j set equal to the initial scale. The comparison is listed in Table 10, where it is clear that a multi-scale approach outperforms the classic single-scale HViT architecture, and that using too many scales diminish the performance.

Table 12
Effect of token dimensionality D**.** We see that increasing the token dimensionality leads to poorer performance.

D	#P	$F2_{CIW}$	$F1_{Normal}$
512	5.92	**61.68**	92.44
1024	20.23	61.34	**92.49**
2048	74.01	60.36	92.44

Table 13
Effect of MLP ratio r**.** We see that increasing the MLP ratio in general leads to better performance.

r	#P	$F2_{CIW}$	$F1_{Normal}$
1	2.77	60.98	92.45
2	3.82	61.31	92.48
3	4.87	61.02	**92.50**
4	5.92	**61.68**	92.44

Table 14
Effect of depth of the ViTs L**.** We observe that increasing or decreasing the depth of the ViTs leads to poorer performance, with the best performance obtained when $L = 2$.

L	#P	$F2_{CIW}$	$F1_{Normal}$
1	3.82	61.05	92.51
2	5.92	**61.68**	92.44
3	8.02	60.53	**92.55**

5.8. Comparison of cross-scale connections

A key part of the MSHViT architecture is the multi-scale connections which enable information sharing across scales. Three variations are presented in Eqs. (8)–(10), and compared in Table 11. We also compare against a scenario with no cross-scale information sharing between the ViTs, instead using a late-stage scale-fusion step. The late-stage fusion step combines the CLS tokens from each scale together with a learnable cross-scale CLS token, using a MHSA operation with 8 heads. We find that all cross-scale connections outperform the late-stage scale-fusion variation and that using the ViT or linearly embedded CNN features led to a decrease in metric performance. Instead the best performance is achieved by sharing the clustered tokens from the Sinkhorn tokenizer across scales, indicating that the clustering process is crucial for performance. We also compare sharing weights for the ViTs when applicable, and find that sharing ViT weights results in a clear performance benefit, unlike when sharing weights and cluster centers in the tokenizer (See Section 5.5).

5.9. Effect of ViT hyperparameters

Lastly, we investigate the effect of varying the hyperparameters of the ViT. Specifically, we investigate the effect of the token dimensionality, D, the MLP ratio, r, in the inverted bottleneck, and the depth of the ViT, L. The effect on the metrics are reported in Tables 12–14, as well as the number of trainable parameters in the MSHViT extension, #P. From these results we observe a clear decrease in metric performance when increasing the token dimensionality D, as well as when the ViT is too shallow or deep. For the MLP ratio we observe that best performance is achieved when $r = 4$, with performance in general decreasing when lowering r as the inverted bottleneck becomes narrower.

6. Sinkhorn tokenizer cluster visualizations

We visualize the cluster assignments within the Sinkhorn Tokenizer of the ResNet-50-MSHViT model to get a better understanding of how the non-local features are combined. For each cluster k we get the

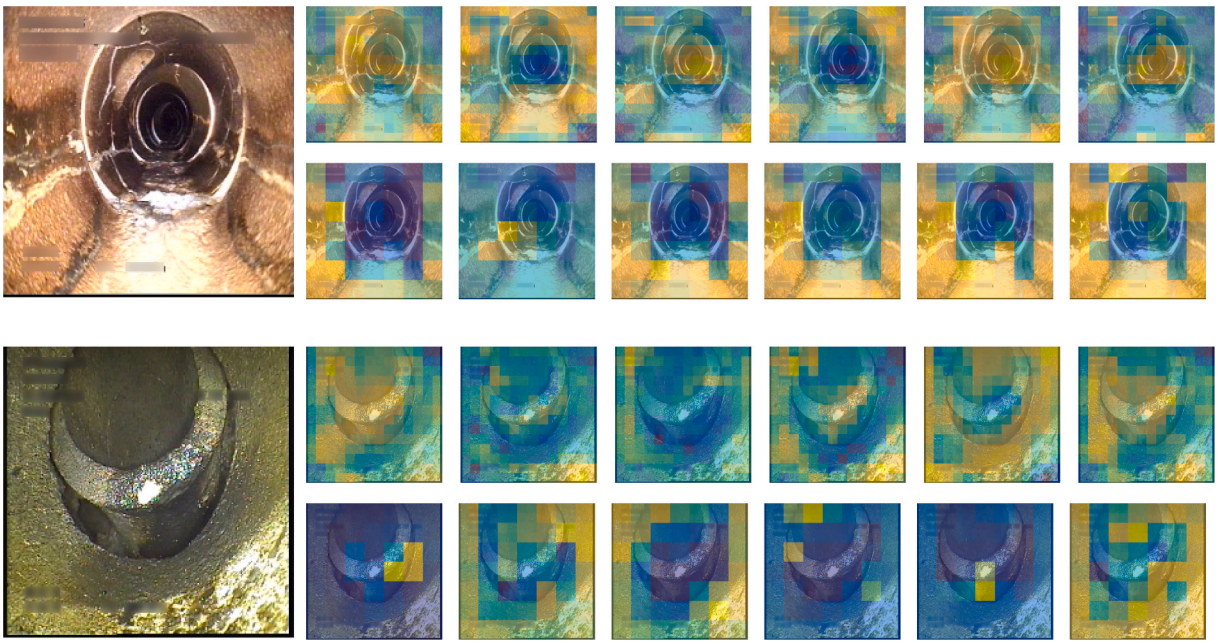


Fig. 7. Visualization of the Sinkhorn Tokenizer clusters. We show a subset of the cluster assignments for two images using the ResNet-50-MSHViT model. The first image contains the classes **cracks, breaks, and collapses (RB)**, **displaced joint (FS)**, and **branch pipe (GR)**, and the second image contains the classes **surface damage (OB)**, **displaced joint (FS)**, and **connection with construction changes (OK)**. For each image, two rows of cluster assignment map examples are shown along the columns. The top row shows six examples from the 4th scale clusters, whereas the bottom row shows six examples from the 5th scale clusters. See the description of the computation of the cluster assignment maps in Section 6.

probability for each pixel that the pixel belongs to cluster k. We then visualize this map using a "PARULA" color mapping, where the mapping ranges from the minimum to maximum probability assignment. The PARULA color mapping maps the lowest value to blue and the largest value to yellow, with teal as the intermediate color.

In tokenizers where information from previous scales is included, we visualize the clusters by first determining the assignment probability per pixel for the scale in focus. Then, for each cluster center from the previous scales we normalize the cluster assignments such that the maximum value is one. The cluster assignments are then multiplied by the assignment probability from the current scale cluster center and added to the overall assignment map. Lastly, the combined probability map is colored with a PARULA color mapping as before.

Examples are shown in Fig. 7. From these examples it is clear that not only does the Sinkhorn Tokenizer lead to non-local interactions, but it also captures the different scales of the defects. This is exemplified by the highlight of the multi-scale cracks as shown in the top example of Fig. 7 and the displaced pipe in the bottom example of Fig. 7. We observe that the clusters capture parts of the same regions, but in different context such as one cluster center capturing a crack running along the pipe wall while another cluster center captures a cross section of the pipe.

7. Limitations and practical use

We have demonstrated that the proposed MSHViT framework improves the sewer defect classification performance, while needing less information in the training process compared to the previously best method, the CT-GAT [37]. However, the proposed method is not yet in a state where it can be used to fully automate sewer inspections, due to poor performance on defect classes such as the very important cracks, breaks, and collapses (RB) class as shown in Fig. 2a. This is, however, true for the entire sewer defect classification field, as demonstrated by the low performance of all methods compared by Haurum and Moeslund [16]. Instead, it is more plausible that the MSHViT framework can be used as an *assistive* tool during the inspection process, providing defect predictions to the sewer inspectors, who can then choose to use, adapt, or reject the proposed classifications. Furthermore, similar to prior work from Yang et al. [28] and Dang et al. [29], the MSHViT framework has an extra benefit in that the cluster assignment maps can be used to relate the output predictions to the input image, as demonstrated in Section 6. This makes the automatic classification process less opaque to the sewer inspector, and may help reduce variability in sewer inspections [84,85]. The proposed framework is also limited in that it has only been evaluated for frame-level recognition of sewer defects, and not dense recognition tasks such as object- and instance-level recognition of sewer defects. This choice has primarily been motivated by the lack of publicly available data sewer inspection data with object, semantic, or instance annotations. However, it would be possible to extend the MSHViT framework for dense sewer defect recognition by utilizing an upsampling framework [86,87], where the soft assignment scores $\mathbf{Q}^*$ can be used to reverse the Sinkhorn-Knopp clustering step. This is, however, left for future work.

8. Conclusions

Vision Transformers (ViTs) have taken the computer vision domain by storm, and led a surge in Transformer focused research. A large part of this research focuses on exclusively using a Transformer based architecture, while in comparison little attention has been given to the fusion of CNNs and Transformers.

In this paper, we presented the Multi-Scale Hybrid Vision Transformer (MSHViT) for image classification, a natural extension of the Hybrid Vision Transformer (HViT) which combines CNNs and ViTs, and the Sinkhorn Tokenizer, a clustering-based tokenizer based on Sinkhorn distances. The MSHViT extension enables the model to learn multi-scale non-local spatial semantics in the input, while the Sinkhorn tokenizer produces a smaller set of tokens that captures non-local spatial semantics.

We investigated the relative performance difference when extending ResNets with MSHViT and Sinkhorn tokenizer on the Sewer-ML multi-label sewer defect classification dataset, demonstrating a relative

improvement of up to 2.53 percentage points. Through an extensive ablation study, we provided insights into the sensitivity of the introduced hyperparameters, verifying that the multi-scale extension outperforms regular HViTs, as well as qualitatively showing how the Sinkhorn tokenizer cluster centers captures distinct spatial semantics from one another.

While the focus of this work has been on the sewer defect classification task, the MSHViT framework can in the future be extended to more dense recognition tasks such as defect detection and segmentation, by following commonly used upsampling-based approaches. However, this has been left for future work, due to the lack of publicly available datasets for sewer defect detection and segmentation. We hope that this work will inspire future work in the sewer defect classification area.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

A link to the code and model weights is available at https://vap.aau.dk/mshvit/. The Sewer-ML dataset is already freely available.

Acknowledgments

This research was funded by Innovation Fund Denmark [Grant No. 8055-00015A] and is part of the Automated Sewer Inspection Robot (ASIR) project, and partially supported by the Spanish projectPID2019-105093 GB-I00 (MINECO/FEDER, UE), and by ICREA under the ICREA Academia programme. The authors declare no conflict of interest.

Appendix A. Symbol and notation guide

An overview of the introduced symbols and relevant notation is presented in Table A.15, with headers indicating in which part of the methodology the notation and symbols are used.

Table A.15
Symbols and notation. Overview of key symbols and notations used in this paper.

Symbols and Notation	
Vision Transformer (ViT)	
L	Number of layers in the ViT.
P	Patch size of the linear embedding tokenizer.
N	Number of linearly embedded tokens.
D	Dimensionality of the linear embeddings.
r	Inverted bottleneck ratio in MLPs.
$\mathbf{X}$	Input image/feature map.
$\mathbf{T}_p$	Linearly embedded feature tokens.
$\mathbf{x}_{\text{CLS}}$	Class token appended to $\mathbf{T}_p$.
$\mathbf{E}_{\text{pos}}$	Positional embedding of $\mathbf{T}_p$ and $\mathbf{x}_{\text{CLS}}$.
$\mathbf{Z}_0$	The embedded input tokens of the ViT.
$\mathbf{Z}_L$	The tokens after L ViT layers.
$\mathbf{Z}_{L,0}$	The CLS token after L ViT layers.
$\mathbf{y}$	The final ViT feature representation.
Sinkhorn Tokenizer	
K	Number of cluster centers.
t_{SK}	Iterations of the Sinkhorn-Knopp algorithm.
ϵ	Entropy regularization hyperparameter.
$\mathbf{C}$	Cluster centers.
$\mathbf{V}$	Cosine similarity between $\mathbf{T}_p$ and $\mathbf{C}$.
$\mathbf{1}_K$	K-dimensional vector filled with ones.
$\mathbf{Q}^*$	Soft assignment matrix for clustering of $\mathbf{T}_p$.
$\mathbf{T}_S$	Set of clustered tokens.
Cross-scale Connections	
$\mathbf{S}$	Set of features from previous scales.
j	Initial scale for cross-scale connections.
ψ	Tokenization function (Sinkhorn or identity).
ϕ	ViT of depth L.
$\mathbf{T}^i$	Output tokens of ψ
Functions and Notation	
$\|$	Concatenation of two or more inputs.
$[\cdot]^i$	Scale indicator, *e.g.* $\mathbf{X}^i$ is the input at scale i.

Appendix B. Sewer-ML dataset overview

The Sewer-ML is the world's first and only publicly available sewer defect image recognition dataset, presented by Harum and Moeslund [16]. The dataset was constructed from 75,618 annotated sewer inspection videos obtained over 9 years from three different Danish water utilities. Each video was annotated by a professionally licensed sewer inspector following the Danish sewer inspection standard [88]. We refer to Haurum [89] for an introduction to this standard. The sewer inspectors annotated the videos by assigning a frame-level annotation of a specific defect at a specific time. Using a set of heuristic rules 1.3 million images were extracted, all text redacted using an automated pipeline, and multi-label ground truth labels constructed based on spatial proximity of the annotations. A comprehensive breakdown of the sewer and data properties can be found in the Supplementary materials of the Sewer-ML paper [16].

Classes and Class Importance Weighting. Following the Danish sewer inspection standard [88] there is a total of 18 named defect classes, each with a score representing the economic consequence of the class [90], see Table B.16. Haurum and Moeslund normalized these scores into the range $[0, 1]$ to create a "class-importance weight" (CIW), representing the economic importance of each defect class. It should be noted that the Water Level (VA) class was excluded as an explicit class in the experiments, as it was continuously defined throughout the videos. Instead it has since been treated as a separate classification task by Haurum et al. [38,37]. Lastly, in order to represent the non-defective segment of sewer pipes the implicit "Normal" class was introduced, evaluated by the lack of classification of any of the 18 annotated sewer defect classes.

Evaluation Protocol. In the survey conducted by Haurum and Moeslund [2], it was determined that there has been no consensus on how to evaluate sewer defect recognition systems. A commonly used metric has been the accuracy metric, often used in the general computer vision domain. However, this is a poor metric for imbalanced datasets as well as multi-label datasets, such as the Sewer-ML dataset. Therefore, Haurum and Moeslund [16] proposed to evaluate the model performance using two metrics based on the Fβ metric [91], while incorporating domain knowledge,

$$\mathrm{F}\beta = (1+\beta^2)\frac{\mathrm{Prc}\cdot\mathrm{Rcll}}{\beta^2\mathrm{Prc}+\mathrm{Rcll}} \tag{B.1}$$

where Prc and Rcll are the precision and recall of the classifier, respectively, and β is a weighting of recall, such that the recall β times more important than precision.

A key insight made by Haurum and Moeslund was that in the sewer inspection process false negatives have a larger economic impact than false positives. This is due to false positives being verified by human inspectors before initiating a rehabilitation process, whereas false negatives allows defective pipes to further degrade. The second key insight was that the different defects do not have the same importance, as some have a larger economic impact, see Table B.16. These domain insights were incorporated into the defect evaluation metric $\mathrm{F2}_{\mathrm{CIW}}$ by setting $\beta = 2$, meaning the recall is weighted higher than the precision, and by weighting the class F2 scores by their CIW scores, see Eq. (B.2).

$$\mathrm{F2}_{\mathrm{CIW}} = \frac{\sum_{c=1}^{C}\mathrm{F2}_c\cdot\mathrm{CIW}_c}{\sum_{c=1}^{C}\mathrm{CIW}_c} \tag{B.2}$$

where CIW_c and $\mathrm{F2}_c$ are the CIW and F2-score for class c, respectively, and C is the number of annotated classes.

In order to evaluate the normal pipes, which have a CIW of 0 and therefore not included in $\mathrm{F2}_{\mathrm{CIW}}$, Haurum and Moeslund proposed to simply use the F1 score, denoted as $\mathrm{F1}_{\mathrm{Normal}}$.

Table B.16
Sewer inspection classes. Overview and short description of each annotation class [88] and the class-importance weights (CIW) [90]. Reproduced from Haurum and Moeslund [16].

Code	Description	CIW
VA	Water Level (in percentages)	0.0310
RB	Cracks, breaks, and collapses	1.0000
OB	Surface damage	0.5518
PF	Production error	0.2896
DE	Deformation	0.1622
FS	Displaced joint	0.6419
IS	Intruding sealing material	0.1847
RO	Roots	0.3559
IN	Infiltration	0.3131
AF	Settled deposits	0.0811
BE	Attached deposits	0.2275
FO	Obstacle	0.2477
GR	Branch pipe	0.0901
PH	Chiseled connection	0.4167
PB	Drilled connection	0.4167
OS	Lateral reinstatement cuts	0.9009
OP	Connection with transition profile	0.3829
OK	Connection with construction changes	0.4396

References

[1] American Society of Civil Engineers, 2017 Infrastructure Report Card - Wastewater, https://www.infrastructurereportcard.org/wp-content/uploads/2017/01/Wastewater-Final.pdf, accessed: 20/3-2022, 2017.

[2] J.B. Haurum, T.B. Moeslund, A Survey on Image-Based Automation of CCTV and SSET Sewer Inspections, Automation in Construction. ISSN: 0926-5805 111 (2020), 103061, https://doi.org/10.1016/j.autcon.2019.103061.

[3] C.H. Bahnsen, A.S. Johansen, M.P. Philipsen, J.W. Henriksen, K. Nasrollahi, T.B. Moeslund, 3D Sensors for Sewer Inspection: A Quantitative Review and Analysis, Sensors 21 (7), ISSN 1424–8220, doi:10.3390/s21072553, URL:https://www.mdpi.com/1424-8220/21/7/2553.

[4] Z. Liu, Y. Kleiner, State of the art review of inspection technologies for condition assessment of water pipes, Measurement. ISSN: 0263-2241 46 (1) (2013) 1–15, https://doi.org/10.1016/j.measurement.2012.05.032.

[5] O. Duran, K. Althoefer, L.D. Seneviratne, State of the art in sensor technologies for sewer inspection, IEEE Sensors Journal. ISSN: 1530-437X 2 (2) (2002) 73–81, https://doi.org/10.1109/JSEN.2002.1000245.

[6] M. Cuturi, Sinkhorn Distances: Lightspeed Computation of Optimal Transport, in: Advances in Neural Information Processing Systems, vol. 26, Curran Associates Inc., 2013. URL:https://proceedings.neurips.cc/paper/2013/file/af21d0c97db2e27e13572cbf59eb343d-Paper.pdf, Accessed: 7/9-2022.

[7] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, A. Vaswani, Bottleneck Transformers for Visual Recognition, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 16514–16524, doi:10.1109/CVPR46437.2021.01625.

[8] X. Wang, R. Girshick, A. Gupta, K. He, Non-local Neural Networks, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 7794–7803, doi:10.1109/CVPR.2018.00813.

[9] H. Wang, Y. Zhu, B. Green, H. Adam, A. Yuille, L.-C. Chen, Axial-DeepLab: Stand-Alone Axial-Attention for Panoptic Segmentation, in: European Conference on Computer Vision, Springer International Publishing, Cham, 2020, ISBN 978-3-030-58548-8, pp. 108–126, https://doi.org/10.1007/978-3-030-58548-8_7.

[10] H. Zhao, J. Jia, V. Koltun, Exploring Self-Attention for Image Recognition, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10073–10082, doi:10.1109/CVPR42600.2020.01009.

[11] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-End Object Detection with Transformers, in: European Conference on Computer Vision, Springer International Publishing, Cham, 2020, ISBN 978-3-030-58452-8, pp. 213–229, https://doi.org/10.1007/978-3-030-58452-8_13.

[12] S. Liu, L. Zhang, X. Yang, H. Su, J. Zhu, Query2Label: A Simple Transformer Way to Multi-Label Classification, ArXiv URL:https://arxiv.org/abs/2107.10834, Accessed: 7/9-2022.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale, in: International Conference on Learning Representations, URL:https://openreview.net/forum?id=YicbFdNTTy, Accessed: 7/9-2022, 2021.

[14] Z. Dai, H. Liu, Q.V. Le, M. Tan, CoAtNet: Marrying Convolution and Attention for All Data Sizes, in: Advances in Neural Information Processing Systems, vol. 34, Curran Associates Inc., 2021, pp. 3965–3977. URL:https://proceedings.neurips.cc/paper/2021/file/20568692db622456cc42a2e853ca21f8-Paper.pdf, Accessed: 7/9-2022.

[15] K. Desai, J. Johnson, VirTex: Learning Visual Representations from Textual Annotations, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 11157–11168, doi:10.1109/CVPR46437.2021.01101.

[16] J.B. Haurum, T.B. Moeslund, Sewer-ML: A Multi-Label Sewer Defect Classification Dataset and Benchmark, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13451–13462, doi:10.1109/CVPR46437.2021.01325.

[17] L.M. Dang, H. Wang, Y. Li, T.N. Nguyen, H. Moon, DefectTR: End-to-end defect detection for sewage networks using a transformer, Construction and Building Materials. ISSN: 0950-0618 325 (2022) 126584, https://doi.org/10.1016/j.conbuildmat.2022.126584.

[18] S.S. Kumar, D.M. Abraham, M.R. Jahanshahi, T. Iseley, J. Starr, Automated defect classification in sewer closed circuit television inspections using deep convolutional neural networks, Automation in Construction. ISSN: 0926-5805 91 (2018) 273–283, https://doi.org/10.1016/j.autcon.2018.03.028.

[19] D. Meijer, L. Scholten, F. Clemens, A. Knobbe, A defect classification methodology for sewer image sets with convolutional neural networks, Automation in Construction. ISSN: 0926-5805 104 (2019) 281–298, https://doi.org/10.1016/j.autcon.2019.04.013.

[20] Q. Xie, D. Li, J. Xu, Z. Yu, J. Wang, Automatic Detection and Classification of Sewer Defects via Hierarchical Deep Learning, IEEE Transactions on Automation Science and Engineering. ISSN: 1545-5955 (2019) 1–12, https://doi.org/10.1109/TASE.2019.2900170.

[21] K. Chen, H. Hu, C. Chen, L. Chen, C. He, An Intelligent Sewer Defect Detection Method Based on Convolutional Neural Network, in: 2018 IEEE International Conference on Information and Automation (ICIA), ISSN null, 2018, pp. 1301–1306, doi:10.1109/ICInfA.2018.8812445.

[22] S.I. Hassan, L.M. Dang, I. Mehmood, S. Im, C. Choi, J. Kang, Y.-S. Park, H. Moon, Underground sewer pipe condition assessment based on convolutional neural networks, Automation in Construction. ISSN: 0926-5805 106 (2019) 102849, https://doi.org/10.1016/j.autcon.2019.102849.

[23] J. Myrans, R. Everson, Z. Kapelan, Automated detection of fault types in CCTV sewer surveys, Journal of Hydroinformatics. ISSN: 1464-7141 21 (1) (2018) 153–163, https://doi.org/10.2166/hydro.2018.073.

[24] Z. Situ, S. Teng, H. Liu, J. Luo, Q. Zhou, Automated Sewer Defects Detection Using Style-Based Generative Adversarial Networks and Fine-Tuned Well-Known CNN Classifier, IEEE Access 9 (2021) 59498–59507, https://doi.org/10.1109/ACCESS.2021.3073915.

[25] D. Ma, J. Liu, H. Fang, N. Wang, C. Zhang, Z. Li, J. Dong, A Multi-defect detection system for sewer pipelines based on StyleGAN-SDM and fusion CNN, Construction and Building Materials. ISSN: 0950-0618 312 (2021) 125385, https://doi.org/10.1016/j.conbuildmat.2021.125385.

[26] Y. Gu, W. Tu, Q. Li, T. Zhao, D. Zhao, S. Zhu, J. Zhu, Collaboratively Inspect Large-Area Sewer Pipe Networks Using Pipe Robotic Capsules, in: Proceedings of the 29th International Conference on Advances in Geographic Information Systems, Association for Computing Machinery, 2021, ISBN 9781450386647, pp. 211–220, https://doi.org/10.1145/3474717.3483948.

[27] M. Klusek, T. Szydlo, Supporting the Process of Sewer Pipes Inspection Using Machine Learning on Embedded Devices, in: International Conference on Computational Science, Springer International Publishing, Cham, 2021, ISBN 978-3-030-77980-1, pp. 347–360, https://doi.org/10.1007/978-3-030-77980-1_27.

[28] S. Yang, Z. Zhao, Q. Yang, J. Wang, Attention Guided Image Enhancement Network for Sewer Pipes Defect Detection, in: 4th International Conference on Intelligent Robotics and Control Engineering, 2021, pp. 109–113, doi:10.1109/IRCE53649.2021.9570948.

[29] L.M. Dang, S. Kyeong, Y. Li, H. Wang, T.N. Nguyen, H. Moon, Deep learning-based sewer defect classification for highly imbalanced dataset, Computers & Industrial Engineering. ISSN: 0360-8352 161 (2021) 107630, https://doi.org/10.1016/j.cie.2021.107630.

[30] M. Wang, H. Luo, J.C. Cheng, Towards an automated condition assessment framework of underground sewer pipes based on closed-circuit television (CCTV) images, Tunnelling and Underground Space Technology. ISSN: 0886-7798 110 (2021) 103840, https://doi.org/10.1016/j.tust.2021.103840.

[31] M. Wang, J.C.P. Cheng, A unified convolutional neural network integrated with conditional random field for pipe defect segmentation, Computer-Aided Civil and Infrastructure Engineering 35 (2) (2020) 162–177, https://doi.org/10.1111/mice.12481.

[32] Q. Zhou, Z. Situ, S. Teng, H. Liu, W. Chen, G. Chen, Automatic sewer defect detection and severity quantification based on pixel-level semantic segmentation, Tunnelling and Underground Space Technology. ISSN: 0886-7798 123 (2022) 104403, https://doi.org/10.1016/j.tust.2022.104403.

[33] S.S. Kumar, M. Wang, D.M. Abraham, M.R. Jahanshahi, T. Iseley, J.C.P. Cheng, Deep Learning Based Automated Detection of Sewer Defects in CCTV Videos, Journal of Computing in Civil Engineering 34 (1) (2020), 04019047, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000866.

[34] Y. Tan, R. Cai, J. Li, P. Chen, M. Wang, Automatic detection of sewer defects based on improved you only look once algorithm, Automation in Construction. ISSN: 0926-5805 131 (2021), 103912, https://doi.org/10.1016/j.autcon.2021.103912.

[35] M. Wang, S.S. Kumar, J.C. Cheng, Automated sewer pipe defect tracking in CCTV videos based on defect detection and metric learning, Automation in Construction. ISSN: 0926-5805 121 (2021), 103438, https://doi.org/10.1016/j.autcon.2020.103438.

[36] Y. Li, H. Wang, L. Dang, M. Jalil Piran, H. Moon, A robust instance segmentation framework for underground sewer defect detection, Measurement. ISSN: 0263-2241 190 (2022), 110727, https://doi.org/10.1016/j.measurement.2022.110727.

[37] J.B. Haurum, M. Madadi, S. Escalera, T.B. Moeslund, Multi-Task Classification of Sewer Pipe Defects and Properties using a Cross-Task Graph Neural Network Decoder, in: IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 1441–1452, doi:10.1109/WACV51458.2022.00151.

[38] J.B. Haurum, C.H. Bahnsen, M. Pedersen, T.B. Moeslund, Water Level Estimation in Sewer Pipes Using Deep Convolutional Neural Networks, Water 12 (12), ISSN 2073–4441, doi:10.3390/w12123412.

[39] F. Plana Rius, M.P. Philipsen, J.M. Mirats Tur, T.B. Moeslund, C. Angulo Bahón, M. Casas, Autoencoders for Semi-Supervised Water Level Modeling in Sewer Pipes with Sparse Labeled Data, Water 14 (3), ISSN 2073–4441, doi:10.3390/w14030333.

[40] H.W. Ji, S.S. Yoo, B.-J. Lee, D.D. Koo, J.-H. Kang, Measurement of Wastewater Discharge in Sewer Pipes Using Image Analysis, Water 12 (6), ISSN 2073–4441, doi:10.3390/w12061771.

[41] H.W. Ji, S.S. Yoo, D.D. Koo, J.-H. Kang, Determination of Internal Elevation Fluctuation from CCTV Footage of Sanitary Sewers Using Deep Learning, Water 13 (4), ISSN 2073–4441, doi:10.3390/w13040503.

[42] C. Siu, M. Wang, J.C. Cheng, A framework for synthetic image generation and augmentation for improving automatic sewer pipe defect detection, Automation in Construction. ISSN: 0926-5805 137 (2022), 104213, https://doi.org/10.1016/j.autcon.2022.104213.

[43] K.S. Henriksen, M.S. Lynge, M.D.B. Jeppesen, M.M.J. Allahham, I.A. Nikolov, J. B. Haurum, T.B. Moeslund, Generating Synthetic Point Clouds of Sewer Networks: An Initial Investigation, in: International Conference on Augmented Reality, Virtual Reality and Computer Graphics, Springer International Publishing, Cham, 2020, ISBN 978-3-030-58468-9, pp. 364–373, https://doi.org/10.1007/978-3-030-58468-9_26.

[44] J.B. Haurum, M.M.J. Allahham, M.S. Lynge, K.S. Henriksen, I.A. Nikolov, T. B. Moeslund, Sewer Defect Classification using Synthetic Point Clouds, in: Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, vol. 5, VISAPP, INSTICC, SciTePress, 2021, ISBN 978-989-758-488-6, pp. 891–900, https://doi.org/10.5220/0010207908910900.

[45] Y.-W. Jeong, K.-B. Sim, S. Park, J. Oh, W.J. Choi, Generation of CNN Architectures Using the Harmonic Search Algorithm and its Application to Classification of Damaged Sewer, IEEE Access 10 (2022) 32150–32160, https://doi.org/10.1109/ACCESS.2022.3160719.

[46] Y. Zhou, A. Ji, L. Zhang, Sewer defect detection from 3D point clouds using a transformer-based deep learning model, Automation in Construction. ISSN: 0926-5805 136 (2022), 104163, https://doi.org/10.1016/j.autcon.2022.104163.

[47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L.u. Kaiser, I. Polosukhin, Attention is All you Need, in: Advances in Neural Information Processing Systems, vol. 30, Curran Associates Inc., 2017. URL:https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf, Accessed: 7/9-2022.

[48] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jegou, Training data-efficient image transformers & distillation through attention, in: Proceedings of the 38th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 139, PMLR, 2021, pp. 10347–10357. URL:https://proceedings.mlr.press/v139/touvron21a.html, Accessed: 7/9-2022.

[49] T. Xiao, M. Singh, E. Mintun, T. Darrell, P. Dollár, R. Girshick, Early Convolutions Help Transformers See Better, in: Advances in Neural Information Processing Systems, vol. 34, Curran Associates Inc., 2021. URL:https://proceedings.neurips.cc/paper/2021/file/ff1418e8cc993fe8abcfe3ce2003e5c5-Paper.pdf, Accessed: 7/9-2022.

[50] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, L. Zhang, CvT: Introducing Convolutions to Vision Transformers, in: IEEE/CVF International Conference on Computer Vision, 2021, pp. 22–31, doi:10.1109/ICCV48922.2021.00009.

[51] H. Zhao, L. Jiang, J. Jia, P. Torr, V. Koltun, Point Transformer, in: IEEE/CVF International Conference on Computer Vision, 2021, pp. 16239–16248, doi:10.1109/ICCV48922.2021.01595.

[52] R. Ranftl, A. Bochkovskiy, V. Koltun, Vision Transformers for Dense Prediction, in: IEEE/CVF International Conference on Computer Vision, 2021, pp. 12159–12168, doi:10.1109/ICCV48922.2021.01196.

[53] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin Transformer: Hierarchical Vision Transformer using Shifted Windows, in: IEEE/CVF International Conference on Computer Vision, 2021, pp. 9992–10002, doi:10.1109/ICCV48922.2021.00986.

[54] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, J. Gao, Focal Attention for Long-Range Interactions in Vision Transformers, in: Advances in Neural Information Processing Systems, vol. 34, Curran Associates Inc., 2021, pp. 30008–30022. URL:https://proceedings.neurips.cc/paper/2021/file/fc1a36821b02abbd2503fd949bfc9131-Paper.pdf, Accessed: 7/9-2022.

[55] X. Cheng, H. Lin, X. Wu, D. Shen, F. Yang, H. Liu, N. Shi, MLTR: Multi-Label Classification with Transformer, in: IEEE International Conference on Multimedia and Expo, 2022, pp. 1–6, doi:10.1109/ICME52920.2022.9860016.

[56] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, C. Feichtenhofer, Multiscale Vision Transformers, in: IEEE/CVF International Conference on Computer Vision, 2021, pp. 6804–6815, doi:10.1109/ICCV48922.2021.00675.

[57] J. He, J.-N. Chen, S. Liu, A. Kortylewski, C. Yang, Y. Bai, C. Wang, TransFG: A Transformer Architecture for Fine-Grained Recognition, Proceedings of the AAAI Conference on Artificial Intelligence 36 (1) (2022) 852–860, https://doi.org/10.1609/aaai.v36i1.19967.

[58] C.-F.R. Chen, Q. Fan, R. Panda, CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification, in: IEEE/CVF International Conference on Computer Vision, 2021, pp. 347–356, doi:10.1109/ICCV48922.2021.00041.

[59] P. Zhang, X. Dai, J. Yang, B. Xiao, L. Yuan, L. Zhang, J. Gao, Multi-Scale Vision Longformer: A New Vision Transformer for High-Resolution Image Encoding, in: IEEE/CVF International Conference on Computer Vision, 2021, pp. 2978–2988, doi:10.1109/ICCV48922.2021.00299.

[60] J. Gu, H. Kwon, D. Wang, W. Ye, M. Li, Y.-H. Chen, L. Lai, V. Chandra, D.Z. Pan, Multi-Scale High-Resolution Vision Transformer for Semantic Segmentation, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 12094–12103, URL:https://openaccess.thecvf.com/content/CVPR2022/html/Gu_Multi-Scale_High-Resolution_Vision_Transformer_for_Semantic_Segmentation_CVPR_2022_paper.html, Accessed: 7/9-2022.

[61] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: International Conference on Learning Representations, 2013, URL:http://arxiv.org/abs/1301.3781, Accessed: 7/9-2022.

[62] J. Pennington, R. Socher, C.D. Manning, GloVe: Global Vectors for Word Representation, in: Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1532–1543, doi:10.3115/v1/D14-1162.

[63] Y. Zhu, Y. Zhu, J. Du, Y. Wang, Z. Ou, F. Feng, J. Tang, Make A Long Image Short: Adaptive Token Length for Vision Transformers, ArXiv abs/2112.01686, URL:https://arxiv.org/abs/2112.01686, Accessed: 7/9-2022.

[64] S. Goyal, A.R. Choudhury, S. Raje, V. Chakaravarthy, Y. Sabharwal, A. Verma, PoWER-BERT: Accelerating BERT Inference via Progressive Word-vector Elimination, in: Proceedings of the 37th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 3690–3699. URL:https://proceedings.mlr.press/v119/goyal20a.html, Accessed: 7/9-2022.

[65] Y. Rao, W. Zhao, B. Liu, J. Lu, J. Zhou, C.-J. Hsieh, DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification, in: Advances in Neural Information Processing Systems, vol. 34, Curran Associates Inc., 2021, pp. 13937–13949. URL:https://proceedings.neurips.cc/paper/2021/file/747d3443e319a22747fbb873e8b2f9f2-Paper.pdf, Accessed: 7/9-2022.

[66] D. Marin, J.-H.R. Chang, A. Ranjan, A. Prabhu, M. Rastegari, O. Tuzel, Token Pooling in Vision Transformers, ArXiv URL:https://arxiv.org/abs/2110.03860, Accessed: 7/9-2022.

[67] J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, ArXiv URL:https://arxiv.org/abs/1607.06450, Accessed: 7/9-2022.

[68] Y.M. Asano, C. Rupprecht, A. Vedaldi, Self-labelling via simultaneous clustering and representation learning, in: International Conference on Learning Representations, URL:https://openreview.net/forum?id=Hyx-jyBFPr, Accessed: 7/9-2022, 2020.

[69] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, A. Joulin, Unsupervised Learning of Visual Features by Contrasting Cluster Assignments, in: Advances in Neural Information Processing Systems, vol. 33, Curran Associates Inc., 2020, pp. 9912–9924. URL:https://proceedings.neurips.cc/paper/2020/file/70feb62b69f16e0238f741fab228fec2-Paper.pdf, Accessed: 7/9-2022.

[70] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778, doi:10.1109/CVPR.2016.90.

[71] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the Inception Architecture for Computer Vision, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826, doi:10.1109/CVPR.2016.308.

[72] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, Proceedings of the AAAI Conference on Artificial Intelligence 10 (2017) 4278–4284. URL:https://arxiv.org/abs/1602.07261v2, Accessed: 7/9-2022.

[73] J. Lee-Thorp, J. Ainslie, I. Eckstein, S. Ontanon, FNet: Mixing Tokens with Fourier Transforms, in: Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 2022, pp. 4296–4313, https://doi.org/10.18653/v1/2022.naacl-main.319.

[74] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, S. Belongie, Class-Balanced Loss Based on Effective Number of Samples, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9260–9269, doi:10.1109/CVPR.2019.00949.

[75] T. Ridnik, H. Lawen, A. Noy, E. Ben, B.G. Sharir, I. Friedman, TResNet: High Performance GPU-Dedicated Architecture, in: IEEE Winter Conference on Applications of Computer Vision, 2021, pp. 1399–1408, doi:10.1109/WACV48630.2021.00144.

[76] T. Ridnik, E. Ben-Baruch, N. Zamir, A. Noy, I. Friedman, M. Protter, L. Zelnik-Manor, Asymmetric Loss For Multi-Label Classification, in: IEEE/CVF International Conference on Computer Vision, 2021, pp. 82–91, doi:10.1109/ICCV48922.2021.00015.

[77] J. Hu, L. Shen, G. Sun, Squeeze-and-Excitation Networks, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141, doi:10.1109/CVPR.2018.00745.

[78] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An Imperative Style, High-Performance Deep Learning Library, in: Advances in Neural Information Processing Systems, vol. 32, Curran Associates Inc., 2019, pp. 8024–8035. URL:https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf, Accessed: 7/9-2022.

[79] W. Falcon, PyTorch Lightning, URL:https://github.com/PyTorchLightning/pytorch-lightning, Accessed: 7/9-2022, 2019.

[80] R. Wightman, PyTorch Image Models, doi:10.5281/zenodo.4414861, URL:https://github.com/rwightman/pytorch-image-models, Accessed: 7/9-2022, 2019.

[81] M. Dehghani, Y. Tay, A. Arnab, L. Beyer, A. Vaswani, The Efficiency Misnomer, in: International Conference on Learning Representations, 2022, URL:https://openreview.net/forum?id=iulEMLYh1uR, Accessed: 7/9-2022.

[82] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, J. Carreira, Perceiver: General Perception with Iterative Attention, in: Proceedings of the 38th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 139, PMLR, 2021, pp. 4651–4664. URL:https://proceedings.mlr.press/v139/jaegle21a.html, Accessed: 7/9-2022.

[83] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer, O.J. Henaff, M. Botvinick, A. Zisserman, O. Vinyals, J. Carreira, Perceiver IO: A General Architecture for Structured Inputs & Outputs, in: International Conference on Learning Representations, 2022, URL:https://openreview.net/forum?id=fILj7WpI-g, Accessed: 7/9-2022.

[84] J. Dirksen, F.H. Clemens, H. Korving, F. Cherqui, P.L. Gauffre, T. Ertl, H. Plihal, K. Müller, C.T. Snaterse, The consistency of visual sewer inspection data, Structure and Infrastructure Engineering 9 (3) (2013) 214–228, https://doi.org/10.1080/15732479.2010.541265.

[85] A.J. van der Steen, J. Dirksen, F.H. Clemens, Visual sewer inspection: detail of coding system versus data quality? Structure and Infrastructure Engineering 10 (11) (2014) 1385–1393, https://doi.org/10.1080/15732479.2013.816974.

[86] V. Badrinarayanan, A. Kendall, R. Cipolla, SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (12) (2017) 2481–2495, https://doi.org/10.1109/TPAMI.2016.2644615.

[87] W. Zeng, S. Jin, W. Liu, C. Qian, P. Luo, W. Ouyang, X. Wang, Not All Tokens Are Equal: Human-Centric Visual Analysis via Token Clustering Transformer, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11101–11111, URL:https://openaccess.thecvf.com/content/CVPR2022/html/Zeng_Not_All_Tokens_Are_Equal_Human-Centric_Visual_Analysis_via_Token_CVPR_2022_paper.html, Accessed: 7/9-2022.

[88] Dansk Vand og Spildevandsforening (DANVA), Fotomanualen: TV-inspektion af afløbsledninger, Dansk Vand og Spildevandsforening (DANVA), sixth ed., ISBN 87-90455-81-9, ISBN: 87-90455-81-9, 2010.

[89] J.B. Haurum, A Deep Dive into Computer Vision Aided Sewer Inspections, Ph.D. thesis, 2022, URL:https://vbn.aau.dk/da/publications/a-deep-dive-into-computer-vision-aided-sewer-inspections, Accessed: 7/9-2022.

[90] Dansk Vand og Spildevandsforening (DANVA), Fotomanualen: Beregning af Fysisk Indeks ved TV-inspektion, Dansk Vand og Spildevandsforening (DANVA), first ed., ISBN 87-90455-52-5, ISBN: 87-90455-52-5, 2005.

[91] C.J. van Rijsbergen, Information Retrieval, second ed., Butterworth-Heinemann, USA, 1979. ISBN 04-08709-29-4, ISBN: 04-08709-29-4.