Multi-Task Classification of Sewer Pipe Defects and Properties using a Cross-Task Graph Neural Network Decoder

Joakim Bruslund Haurum[1] Meysam Madadi[2] Sergio Escalera[1,2,3] Thomas B. Moeslund[1]

[1] Visual Analysis and Perception (VAP) Laboratory, Aalborg University, Denmark
[2] Computer Vision Center, Autonomous University of Barcelona, Spain
[3] Dept. of Mathematics and Informatics, Universitat de Barcelona, Spain

joha@create.aau.dk, mmadadi@cvc.uab.es, sergio@maia.ub.es, tbm@create.aau.dk

Abstract

The sewerage infrastructure is one of the most important and expensive infrastructures in modern society. In order to efficiently manage the sewerage infrastructure, automated sewer inspection has to be utilized. However, while sewer defect classification has been investigated for decades, little attention has been given to classifying sewer pipe properties such as water level, pipe material, and pipe shape, which are needed to evaluate the level of sewer pipe deterioration.

In this work we classify sewer pipe defects and properties concurrently and present a novel decoder-focused multi-task classification architecture Cross-Task Graph Neural Network (CT-GNN), which refines the disjointed per-task predictions using cross-task information. The CT-GNN architecture extends the traditional disjointed task-heads decoder, by utilizing a cross-task graph and unique class node embeddings. The cross-task graph can either be determined a priori based on the conditional probability between the task classes or determined dynamically using self-attention. CT-GNN can be added to any backbone and trained end-to-end at a small increase in the parameter count. We achieve state-of-the-art performance on all four classification tasks in the Sewer-ML dataset, improving defect classification and water level classification by 5.3 and 8.0 percentage points, respectively. We also outperform the single task methods as well as other multi-task classification approaches while introducing 50 times fewer parameters than previous model-focused approaches. The code and models are available at the project page `http://vap.aau.dk/ctgnn`.

1. Introduction

The sewerage infrastructure is a key infrastructure of modern society, which needs to be regularly inspected and maintained in order to ensure its functionality [3]. These inspections require professional sewer inspectors who are capable of documenting and differentiating the fine-grained sewer defects, but also the properties of the sewer pipe such as the water level, pipe shape and pipe material, see Figure 1. All of this information can be combined to compute a single deterioration score for each sewer pipe [14] used by water utility companies for asset management. Due to the hidden nature of the sewerage infrastructure sewer inspections are hard and cumbersome to conduct, as the sewer inspectors have to inspect using a remote controlled vehicle with a movable camera. Each inspection can stretch over a long duration of time due to obstacles in the sewers and limited speed of the vehicle. This leads to prolonged duration of looking at a screen, and can potentially result in flawed inspections due to fatigue.

Task	Ground Truth	R50-MTL	CT-GNN
Defect	FS, RO	FS	FS, RO
Water	[0%,5%)	[0%,5%)	[0%,5%)
Shape	Circular	Circular	Circular
Material	VC	VC	VC

Task	Ground Truth	R50-MTL	CT-GNN
Defect	OB, FS	FS	OB, FS
Water	[0%,5%)	[0%,5%)	[0%,5%)
Shape	Circular	Circular	Circular
Material	Conc.	VC	Conc.

Figure 1: Example images from the Sewer-ML dataset [25] together with examples showing how the baseline R50-MTL model with no cross-task relationship modeling misses the noticeable roots (RO) and surface damage (OB). Additionally, the R50-MTL model misclassifies the material as vitrified clay (VC) instead of as concrete (Conc.), whereas the proposed CT-GNN model classifies all classes in each task correctly in both examples.

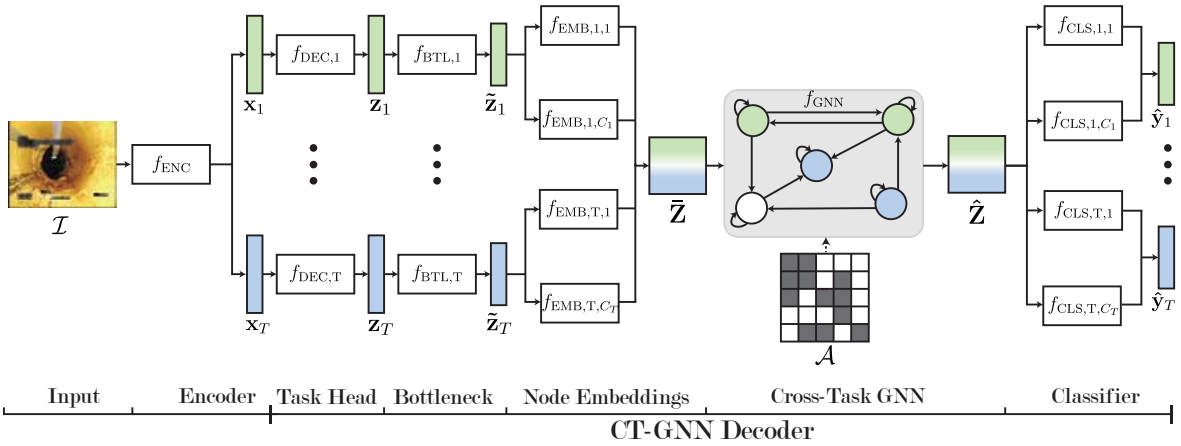


Figure 2: **CT-GNN Overview.** The proposed CT-GNN decoder and its location within the typical MTC architecture. The initial task features, $\mathbf{z}_t,\ t = 1, 2, \ldots, T$, from T disjointed task-heads, are refined using our CT-GNN decoder, which incorporates class relationship knowledge, resulting in the final class predictions $\hat{\mathbf{y}}_t,\ t = 1, 2, \ldots, T$. The CT-GNN is explained in detail in Sections 3.3 & 3.4.

In order to alleviate and assist the sewer inspectors, academia and industry have researched how to automate parts of the inspection process for more than 30 years [24]. However, the majority of work within this field has been focused on the important task of classifying the defects present in the pipes, while omitting the concurrent tasks of determining the water level, pipe material, and pipe shape needed to determine the deterioration score [24]. Furthermore, as the inspections are performed on location it is infeasible to deploy several large models for each task.

Therefore, we investigate how to utilize Multi-Task Learning (MTL), and its sub-field Multi-Task Classification (MTC), to simultaneously classify the sewer pipe defects and properties, by training a single model that is capable of processing multiple tasks during a single forward pass [66].

The MTC problem is often defined as learning how to solve several **unrelated** datasets with a single network [37, 55], whereas the problem of **related** and **concurrent** classification tasks, as *e.g.* during sewer inspections, is less well understood [25, 45]. The occurrence of the different task classes follows a hidden intractable joint distribution over all classes from all tasks. While the joint distribution is intractable, the co-occurrence information of the task classes can be inferred from the data, or learned by a model, and subsequently utilized to improve the classification process.

In order to handle the concurrent MTC problem, we propose a novel decoder-focused model, the Cross-Task Graph Neural Network (CT-GNN) Decoder, where the per-task features are refined using a cross-task sharing mechanism, inspired by recent dense vision decoder-focused models [67, 72, 80, 81]. Specifically, we propose applying a CT-GNN on the initial task feature representations utilizing cross-task class relationships to refine the predictions.

We find that classification of all tasks can be improved by incorporating these cross-task class relationships into the decoder, by either utilizing the a priori known co-occurrence of the different task classes or dynamically estimating it through self-attention. Our proposed method is illustrated in Figure 2. Compared to the previously limited use of graphs in MTC, we do not utilize feature vectors from different images in a batch [18, 50] nor do we consider sequential data inputs [43]. Compared to previous decoder-focused MTC models, we neither estimate the statistical relationship from batches [40], nor impose tensor-based constraints [46, 73].

Our contributions are therefore the following:

- We present the Cross-Task GNN Decoder, a novel MTC decoder that refines the per-task features through a late cross-task mechanism, trained in an end-to-end manner with only a small parameter count increase.
- In order to quantify a priori knowledge of task relationships we construct a cross-task graph adjacency matrix in a data-driven manner.
- We achieve State-of-the-Art performance on all four classification tasks in the Sewer-ML dataset [25], demonstrating the importance of utilizing cross-task relationships during automated sewer inspections.

The paper is structured as follows. In Section 2, we review the related works within the automated sewer inspection as well as MTL and MTC fields. In Section 3, we introduce the CT-GNN decoder head and how to construct the adjacency matrix. In Section 4, we compare the CT-GNN against other MTC methods on the Sewer-ML dataset, investigate per-class performances, and conduct ablation studies. Finally, in Section 5, we conclude the paper.

2. Related Works

Automated Sewer Inspections. The field of automated sewer inspections has been researched for several decades by both academia and industrial research and development [24]. However, until the release of the Sewer-ML dataset [25] there was no public dataset or commonly agreed upon evaluation protocol [24].

The majority of work within the field has instead focused on automatically classifying defects using CCTV images [25] and other sensor based approaches [1, 2, 4, 22, 27, 28, 31, 32, 38, 64]. Only within recent years [24] have deep learning based methods been utilized for defect classification [7, 21, 25, 34, 39, 49, 71], detection [12, 35, 74], segmentation [36, 53, 54, 69], and spatiotemporal based analysis [52, 70]. Defect classification models often employ a two-stage approach with a small initial classifier making a binary defect/non-defect classification, followed by a specialized defect classifier [7, 25, 34, 39, 71]. Recently, work has been conducted on classifying the water level in sewer pipes [23, 29], such that it is possible to estimate how much of the pipe can be inspected for defects. However, no work has been conducted on classifying the sewer pipe defects and properties concurrently. For an in-depth review of the vision-based automated sewer inspection field we refer to the survey by Haurum and Moeslund [24].

Multi-Task Learning. The field of multi-task learning has been applied across several different domains. Within the computer vision domain, MTL has been applied on image-level classification tasks such as facial attributes [45] and age and gender estimation [56, 57], learning several unrelated datasets at a time [37, 55], as well as learning multiple dense vision tasks such as per-pixel depth estimation and semantic segmentation [8, 15, 61, 77]. Two main research branches have been developed through the years: optimization-focused and model-focused approaches [66]. For an exhaustive review of the field we refer to the surveys of the field [58, 66, 78].

The optimization-focused approaches investigate the effect of balancing how the tasks are learned. The tasks are balanced through operations such as normalizing the gradient magnitudes [9], approaching the problem as a multi-objective optimization problem and finding a Pareto optimal solution among all tasks [42, 60], adjusting the task weights based on the loss descent rate [44], the task-dependent homoscedastic uncertainties [30, 41], and more [10, 17, 75]. Each of these approaches is built on different underlying assumptions regarding how the task balancing is controlled, and introduces either an extra computational load or extra hyperparameters.

The model-focused approaches investigate the effect of parameter sharing in the model and is classically split into two types, hard and soft parameter sharing. Hard parameter sharing approaches are built around a shared backbone split into task-specific branches and heads [5, 19, 47, 62, 65], whereas in soft parameter sharing each task is assigned its own parameters with cross-task information introduced through one or more feature sharing mechanisms [16, 44, 51, 59]. Typically, these models utilize an encoder-decoder structure, where an input is passed through an encoder generating a global or per-task feature representation, which is used by a decoder to produce the task predictions [66]. This has led to encoder- and decoder-focused methods.

In encoder-focused models the task parameters are only shared in the encoder, while the decoder consists of disjointed task-heads with no cross-task information [9, 30, 60]. In decoder-focused models, the model parameters are also shared across tasks in the decoder through mechanisms such as multi-model distillation [6, 67, 72, 80, 81], sequential task prediction [79], or cross-task consistency [76]. Decoder-focused models have been applied primarily for dense vision tasks. The few decoder-focused models that have been applied to multi-task classification depend on tensor factorization over pre-trained single task networks [73], placing a tensor normal prior over the decoder [46] and utilizing a maximum a posteriori optimization objective, or constraining the decoder layers based on the task relations [40]. However, the previous methods suffer from either requiring initially training single task networks [73], modifying the optimization loop [46], or limited to two tasks [40].

Lastly, graphs have seen recent usage in the MTL and MTC fields in modeling between- and within-task relationships. An example of this is the PSD-Net which utilized graphlets to improve per-pixel predictions [81]. For multi-task classification, graph neural networks (GNNs) have been used to model the relationship between the multiple inputs in a batch [18, 50], or across sequential data [43]. In concurrent work [63] a Laplacian graph across facial attributes is learned and used within a regularization term during optimization.

Overall, the literature on MTC decoder-focused models is scarce and existing methods either rely on compressing single task networks or constrained to two tasks. Here, we present a novel decoder-focused model, CT-GNN, which is end-to-end trainable for any number of tasks. Furthermore, in contrast to previous usage of graphs in MTC, the CT-GNN is trainable without relying on sequential or batched data for the graph construction.

3. Methodology

In this section, we present our proposed Cross-Task GNN Decoder for Multi-Task Classification. First we provide a recap of Multi-Task Learning and Graph Neural Networks, followed by an explanation of the CT-GNN decoder and how the graph adjacency matrix can be constructed in a data-driven manner.

3.1. Multi-Task Learning Recap

Multi-Task Learning focuses on the problem of classifying a set of T tasks, $\mathcal{T}$, simultaneously. Each task contains a set of C_t classes, for a total of $C = \sum_t C_t$ classes. In the case of sewer inspection each image, $\mathcal{I}$, has T task-specific labels $\mathbf{y}_t$. The MTL networks are optimized using a linear combination of the task-specific losses:

$$\mathcal{L}_{\text{Total}} = \sum_{t=1}^{T} \lambda_t \mathcal{L}_t(\mathcal{I}, \mathbf{y}_t), \tag{1}$$

where λ_t and $\mathcal{L}_t$ are the weight and loss of the tth task, respectively.

When applying multi-task learning methods there are typically varying degrees of parameter sharing in the encoder and no parameter sharing in the decoder. An input image is processed by an encoder network, f_{ENC}, and a set of per-task features $\mathbf{x}_t \in \mathbb{R}^{d_{\text{ENC}}}$ are extracted. If there are no task-specific parameters in f_{ENC} all T tasks will use the same encoded feature $\mathbf{x} \in \mathbb{R}^{d_{\text{ENC}}}$. The encoder features are processed by a decoder network, f_{DEC}, producing predictions for each of the tasks, $\tilde{\mathbf{y}}_t \in \mathbb{R}^{C_t}$. Classically, f_{DEC} is constructed as T disjointed classifiers.

3.2. Graph Neural Network Recap

A graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, is defined as a set of nodes, $\mathcal{V}$, and edges connecting two nodes, $\mathcal{E}$, together with a set of d-dimensional node features $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$. A graph can be represented using an adjacency matrix $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, where entry $\mathbf{A}[u, v]$ is the edge weight from node v to u. The basic GNN is defined by its neural message passing structure where the feature vectors of the nodes are exchanged and updated, constituting a GNN layer [20]. The neural message passing structure for node u and its neighbors $\mathcal{N}(u)$ is defined as:

$$\mathbf{h}_u^{(l+1)} = \psi(\mathbf{h}_u^{(l)}, \phi(\{\mathbf{h}_v^{(l)}, \forall v \in \mathcal{N}(u)\})), \tag{2}$$

where ψ and ϕ are arbitrary differentiable *update* and *aggregation* functions, respectively, and $\mathbf{h}_u^l$ is the hidden embedding of node u at layer l with $\mathbf{h}_u^0 = \mathbf{x}_u$.

3.3. CT-GNN Decoder for Multi-Task Classification

The Cross-Task GNN Decoder builds upon the encoder features, $\mathbf{x}_t$, and consists of the following four parts illustrated in Figure 2: T task-specific decoder heads producing the initial per-task feature representations, T bottleneck layers reducing the dimensionality of the per-task feature vectors, C non-linear node embedding layers, and a cross-task GNN which jointly refines the different class representations based on an a priori or learned directed graph $\mathcal{G}_{\mathcal{T}}$.

Task-Specific Decoders. The task-specific decoder heads are realized as a set of T disjointed networks, $f_{\text{DEC},t}$, each generating a task-specific feature vectors $\mathbf{z}_t = f_{\text{DEC},t}(\mathbf{x}_t)$, $\mathbf{z}_t \in \mathbb{R}^{d_{\text{DEC}}}$. Classically, $\mathbf{z}_t$ is used directly to obtain the class predictions, $\check{\mathbf{y}}_t$, by applying a linear layer followed by the classification activation function of choice. In the CT-GNN decoder framework, however, the task-feature $\mathbf{z}_t$ is used as the foundation for the class-specific node embeddings, in order to allow for initial task-adaption of the encoder feature, $\mathbf{x}_t$.

Bottleneck Layer. In previous work, the dimensionality of the task-specific feature representation $\mathbf{z}_t$ is equal to that of the encoder feature, meaning $d_{\text{ENC}} = d_{\text{DEC}}$ [19, 60]. In the CT-GNN decoder framework this is problematic, as the model parameter count would increase dramatically when transforming the T task-specific features into C unique class-specific features of size d_{EMB}. Therefore, a non-linear down projection layer, $f_{\text{BTL},t}$, is applied in order to reduce the dimensionality of the task-specific features and generate a more compact feature representation, $\tilde{\mathbf{z}}_t \in \mathbb{R}^{d_{\text{BTL}}}$. The bottleneck is realized as a dense layer, $\tilde{\mathbf{z}}_t = f_{\text{BTL},t}(\mathbf{z}_t) = \sigma(\mathbf{z}_t \mathbf{B}_t)$, consisting of the down projection weight matrix, $\mathbf{B}_t \in \mathbb{R}^{d_{\text{DEC}} \times d_{\text{BTL}}}$, where $d_{\text{BTL}} \leq d_{\text{EMB}} \leq d_{\text{DEC}}$, and applying a differentiable non-linear function, σ. $\mathbf{B}_t$ can be task-specific or shared across all T tasks, depending on the number of tasks. For a large number of tasks, using task-specific bottleneck layers would result in a large parameter increase, decreasing the parameter-wise benefits of using a MTL network.

Node Embeddings. The dimensionality-reduced task feature representation, $\tilde{\mathbf{z}}_t$, is subsequently turned into C_t class-specific node embeddings. $\bar{\mathbf{z}}_{t,c} \in \mathbb{R}^{d_{\text{EMB}}}$. Similar to the bottleneck layer, this is realized as a dense layer, $\bar{\mathbf{z}}_{t,c} = f_{\text{EMB},t,c}(\tilde{\mathbf{z}}_t) = \sigma(\tilde{\mathbf{z}}_t \mathbf{E}_{t,c})$, consisting of a matrix multiplication and non-linearity. In order to get the C_t unique node embeddings, we use C_t unique embedding layers, parameterized by C_t unique learnable matrices $\mathbf{E}_{t,c} \in \mathbb{R}^{d_{\text{BTL}} \times d_{\text{EMB}}}$.

Cross-Task GNN. The stacked initial per-class node embeddings, $\bar{\mathbf{Z}} \in \mathbb{R}^{C \times d_{\text{EMB}}}$, of the cross-task graph, $\mathcal{G}_{\mathcal{T}}$, are refined by passing them through a GNN, $\hat{\mathbf{Z}} = f_{\text{GNN}}(\bar{\mathbf{Z}})$, where $\hat{\mathbf{Z}} \in \mathbb{R}^{C \times d_{\text{EMB}}}$ is the stacked GNN-refined node features. The GNN fundamentally builds upon an adjacency matrix of $\mathcal{G}_{\mathcal{T}}$, $\mathcal{A} \in \mathbb{R}^{C \times C}$, which can be learned, provided a priori, or obtained by a combination thereof. The GNN propagates the node embeddings through L hidden layers with d_{EMB} channels, adding contextual information to each node embedding based on its incoming neighbors.

Each node embedding, $\hat{\mathbf{z}}_{t,c} \in \mathbb{R}^{d_{\text{EMB}}}$, is passed through a class-specific linear projection layer, $\hat{z}_{t,c} = f_{\text{CLS},t,c}(\hat{\mathbf{z}}_{t,c})$, to generate a scalar node embedding for each class. The scalar embeddings, $\hat{z}_{t,c}$, are stacked per-task, and the task-specific activation functions are applied to generate the per-task probability vectors, $\hat{\mathbf{y}}_t$. For multi-label and multi-class classification we use the sigmoid and softmax activation.

3.4. Adjacency Matrix Construction

A key part of the CT-GNN Decoder is the construction of the graph, realized by the adjacency matrix $\mathcal{A}$. This adjacency matrix can in theory be arbitrarily set. However, in order to utilize the a priori knowledge of the task relationships, we follow a data-driven approach based on the co-occurrence of the classes. We generalize the graph construction method Chen *et al*. [11] to the multi-task classification scenario.

$\mathcal{A}$ consists of several sub-matrices, $\mathcal{A}_{i,j}$, each describing the relationship between the tasks i and j. Note that in the case that only binary and multi-class classification tasks are considered, $\mathcal{A}$ will be a directed T-partite graph with self-loops. Firstly, the conditional probabilities between the classes in task i and j, $\mathbf{P}_{i,j} \in \mathbb{R}^{C_i \times C_j}$, are calculated based on the co-occurrence matrix between the two tasks, $\mathbf{C}_{i,j} \in \mathbb{R}^{C_i \times C_j}$, see Eq. 3–4. The co-occurrence matrices are calculated using the training splits. We follow the convention that $\mathbf{P}_{i,j}[u, v]$ defines the conditional probability of class u given class v.

$$\mathbf{P}_{i,j}[u, v] = \frac{\mathbf{C}_{i,j}[u, v]}{N_v} \tag{3}$$

$$N_v = \begin{cases} \mathbf{C}_{i,j}[v, v], & i = j \\ \sum_{u=1}^{C_i} \mathbf{C}_{i,j}[u, v], & i \neq j \end{cases} \tag{4}$$

$\mathbf{P}_{i,j}$ is subsequently binarized in order to filter out noisy edges using a task-pair specific threshold $\tau_{i,j}$, see Eq. 5. By utilizing task-pair specific thresholds the different task-pairs can be binarized according to different rules, if desired. The binarized adjacency matrices are then combined into a single adjacency matrix, $\mathbf{A}$, see Eq. 6.

$$\mathbf{A}_{i,j}[u, v] = \begin{cases} 0, & \mathbf{P}_{i,j}[u, v] < \tau_{i,j} \\ 1, & \mathbf{P}_{i,j}[u, v] \geq \tau_{i,j} \end{cases} \tag{5}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \dots & \mathbf{A}_{1,K} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{K,1} & \dots & \mathbf{A}_{K,K} \end{bmatrix} \tag{6}$$

Lastly, the adjacency matrix is re-weighted across the incoming edges per node, in order to counteract the over-smoothing problem with GNNs [11], leading to the final adjacency matrix, $\mathcal{A}$. This is done using $\mathbf{A}$, and enforcing the sum of all incoming edge weights to equal one, setting the sum of the neighbor edge weights to p, while the center node self-loop weight is $1 - p$, see Eq. 7.

$$\mathcal{A}[u, v] = \begin{cases} \mathbf{A}[u, v] \frac{p}{\sum_{v=1, v \neq u}^{C_j} \mathbf{A}[u, v]}, & u \neq v \\ 1 - p, & u = v \end{cases} \tag{7}$$

The larger p is the more weight will be assigned to the incoming neighbor nodes, while a smaller p value will result in more weight assigned to the center node. If a center node has no incoming edges a part from the self-loop, *i.e*. $\sum_{v=1, v \neq u}^{C_j} \mathbf{A}[u, v] = 0$, we set the self-loop weight to one, to avoid the center node embedding decaying to a zero vector.

Table 3: **CT-GNN hyperparameters**. The hyperparameters were found through a sequential search. L is the number of layers in the CT-GNN, d_{ENB} is the dimensionality of the class features, d_{BTL} is the dimensionality of the bottleneck, H is the number of attention heads in the GAT GNN, and τ and p are the thresholding and re-weighting parameters in the adjacency matrix construction, respectively.

Hyperparameter	L	d_{EMB}	d_{BTL}	H	τ	p
GCN	3	512	32	-	0.05	0.2
GAT	1	128	32	8	0.65	-

4. Experimental Results

We evaluate on the Sewer-ML sewer defect and pipe property dataset [25]. The dataset focuses on the multi-label defect classification problem and contains 1.3 million images collected over a nine year period. The data are split into a preset training, validation, and test split, containing 1 million, 130k, and 130k images each [25]. The defect classification problem consists of 17 different classes as well as the implicit normal class. Additionally, the water level, pipe material and pipe shape are also annotated. The water level is annotated in 11 classes from 0 to 100% of the pipe filled with water in 10% steps, and the pipe material and shape tasks contain eight and six classes each. Example images can be found in the supplementary material.

4.1. Evaluation metrics

Model evaluation is done using the per-task evaluation metrics and number of parameters, #P. As the classes in each task are imbalanced the tasks cannot be evaluated using the traditional accuracy metric. Instead, the defect task is evaluated using the F2_{CIW} defect score and the $\text{F1}_{\text{Normal}}$ score [25]. The three remaining tasks are evaluated using both the micro-F1 (mF1) and macro-F1 (MF1) scores.

Lastly, we report the average per-task performance increase for a multi-task model, Δ_{MTL}, with respect to the single task learning (STL) baselines of the same base architecture [48]:

$$\Delta_{\text{MTL}} = \frac{1}{T} \sum_{t=1}^{T} \frac{(M_{m,t} - M_{b,t})}{M_{b,t}}, \tag{8}$$

where $M_{m,t}$ and $M_{b,t}$ are the multi-task and single-task metric performance for task t, receptively.

Table 4: **Results on Sewer-ML.** Comparison between the STL and MTL networks. We compare the effect of CT-GNN using GCN and GAT, denoted CT-GCN and CT-GAT respectively, as well as compare a hard-shared ResNet-50 encoder, and the soft-shared MTAN encoder with a ResNet-50 backbone. #P indicates the number of parameters in millions. * indicates that the method was tested on a subset of the Sewer-ML dataset. Best performance in each column is denoted in **bold**.

	Model		Overall	Defect		Water		Shape		Material	
	Model	#P	$\Delta_{\mathbf{MTL}}$	$\mathbf{F2_{CIW}}$	$\mathbf{F1_{Normal}}$	**MF1**	**mF1**	**MF1**	**mF1**	**MF1**	**mF1**
Validation Split	Benchmark [25]	62.8	-	55.36	91.32	-	-	-	-	-	-
	R50-FT* [23]	23.5	-	-	-	62.53	78.15	-	-	-	-
	STL	94.0	+0.00	58.42	**92.42**	69.11	79.71	46.55	98.06	65.99	96.71
	R50-MTL	23.5	+10.36	59.73	91.87	70.51	80.47	71.64	99.34	80.28	98.09
	MTAN	48.2	+10.40	61.21	92.10	70.06	**80.59**	68.34	**99.40**	83.48	**98.25**
	CT-GCN	25.2	+12.39	61.35	91.84	**70.57**	80.47	**76.17**	99.33	82.63	98.18
	CT-GAT	24.0	**+12.81**	**61.70**	91.94	**70.57**	80.43	74.53	**99.40**	**86.63**	98.24
Test Split	Benchmark [25]	62.8	-	55.11	90.94	-	-	-	-	-	-
	R50-FT* [23]	23.5	-	-	-	62.88	79.29	-	-	-	-
	STL	94.0	+0.00	57.48	**92.16**	69.87	80.09	56.15	97.59	69.02	96.67
	R50-MTL	23.5	+7.39	58.29	91.57	71.17	81.09	79.48	99.19	**76.35**	98.08
	MTAN	48.2	+6.83	59.91	91.72	70.61	**81.16**	78.50	99.21	72.73	**98.27**
	CT-GCN	25.2	+7.64	60.07	91.60	70.69	80.91	80.32	99.19	75.13	98.15
	CT-GAT	24.0	**+7.84**	**60.57**	91.61	**71.30**	80.91	**81.10**	**99.22**	73.95	98.26

4.2. Training Procedure

We utilize the ResNet-50 network [26] as our base encoder, with no task-specific decoders, meaning $\mathbf{x}_t = \mathbf{z}_t$. We cast the defect classification problem as a multi-label classification task with a single task weight, λ_{defect}, while the water level, pipe material, and pipe shape are multi-class classification tasks. For the water level classification task, we adapt the label discretization approach from [23], leading to four water level classes.

We compare performance using the Graph Convolutional Network (GCN) [33] and Graph Attention Network (GAT) [68] in the CT-GNN, denoting the variations CT-GCN and CT-GAT, respectively. We use the reweighted adjacency matrix, $\mathcal{A}$, for GCN, and the binary adjacency matrix, $\mathbf{A}$, for GAT where the edge weights are inferred through self-attention. While the GAT architecture could fully determine the adjacency matrix through self-attention, we found that performance increases if we provide the set of possible graph edges beforehand. The GCN adjacency matrix was symmetrically normalized [33] using the in-degree matrix, and skip connections were inserted between the GNN layers. Finally, we use task-specific bottleneck layers.

Hyperparameters. The networks are trained for 40 epochs using SGD with a learning rate of 0.1, momentum of 0.9, weight decay of 0.0001, and a batch size of 256. The learning rate is multiplied by 0.01 at the 20th and 30th epoch. The hyperparameters used in the CT-GNN, including the number of attention heads in GAT, H, are described in Table 3, and are found through a sequential hyperparameter search described in the supplementary material. Through initial tests we found that a single global threshold τ in the adjacency graph construction leads to the best performance.

Data Augmentation. We follow the data augmentation process by [25], rescaling the images to 224×224, horizontal flipping and jittering the brightness, contrast, hue, and saturation values by $\pm 10\%$. Due to class imbalance in each task, we use class-weighted task-losses with the class weighting method of [13] with $\beta = 0.9999$, except for the defect task where the positive class examples are weighted by their *class importance weights* (CIW) [25].

Loss considerations. For all CT-GNN models the final task loss is a convex combination of the final probability vector $\hat{\mathbf{y}}_t$ and the probability vector produced by applying a classification layer to $\mathbf{z_t}$, denoted $\check{\mathbf{y}}_t$:

$$\mathcal{L}_t = \omega \mathcal{L}_t(\hat{\mathbf{y}}_t, \mathbf{y}_t) + (1 - \omega)\mathcal{L}_t(\check{\mathbf{y}}_t, \mathbf{y}_t), \tag{9}$$

where $\mathcal{L}_t$ is the task-specific loss function for task t, and ω is a weighting hyperparameter in the interval $[0, 1]$. This is to ensure the feature representation $\mathbf{z_t}$ is representative for task t, through an auxiliary loss signal. We set $\omega = 0.75$, such that the primary loss signal is propagated through the CT-GNN.

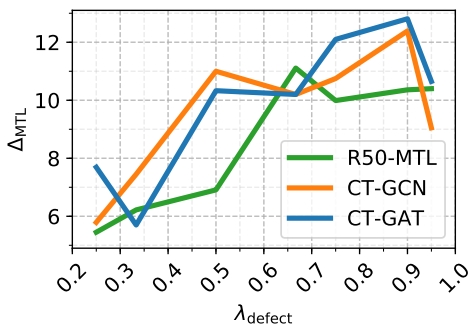


Figure 3: **Evaluating Δ_{MTL} for different λ_{defect}.** Comparison of performance of the R50-MTL, CT-GCN and CT-GAT models. Evaluated on the validation split.

We constrain the task weights to be a convex combination and set to $\lambda_{\text{defect}} = 0.90$ and $\lambda_{\text{water}} = \lambda_{\text{shape}} = \lambda_{\text{material}} = \frac{1-\lambda_{\text{defect}}}{3}$. In order to keep the losses comparable across different settings, we multiply the task weights by T such that $\sum_t \lambda_t = T$, similar to [44].

4.3. Comparative Models

As there are no ResNet-50 STL baselines for all of the tasks, we train these using the same hyperparameters as the in MTL networks. Note that we got the best single-task performance for the defect task using the class weighting method from [13]. We also compare with the benchmark defect classification model from [25], as well as the water level classification model from [23]. As there are no prior work on multi-task classification in the sewer domain [24], we compare with a set of MTL baselines: A hard-shared ResNet-50 MTL network with no CT-GNN (R50-MTL), and the encoder-focused soft-shared MTAN model with a ResNet-50 backbone, see Table 4. Results for the DWA [44] and the uncertainty [30, 41] optimization-based methods can be found in the supplementary materials.

4.4. Results

We find that the CT-GNN outperforms all other methods, beating state-of-the-art defect [25] and water level [23] classifiers by 5.3 and 8.0 percentage points, respectively. We also outperform the baseline STL and MTL networks, by a significant margin on the defect, shape, and material tasks.

The CT-GCN and CT-GAT achieve comparable or better metric performance on all tasks while adding 0.5-1.7 million parameters compared to MTAN encoder-focused method which adds 25 million parameters. Specifically, CT-GAT achieves the highest Δ_{MTL} while introducing 50 times fewer parameters than the MTAN encoder. Unlike soft-shared encoders, the backbone only influences the parameter count of the CT-GNN through the size of the encoder feature $\mathbf{x}_t$.

Comparatively, the optimization-based methods performed worse than using a fixed set of task weights, echoing the results from [66], resulting in a Δ_{MTL} of -15.70% and -4.07% on the validation split and -11.57% and -4.07% on the test split, for the DWA and uncertainty methods respectively. Details are available in the supplementary material.

We also find that the CT-GAT outperforms the CT-GCN on the defect and materials task, while the CT-GCN performs slightly better on the shape task MF1 score. This indicates that there is a clear value in letting the edge weights be dynamically inferred during inference, while prior information can be imbued beforehand through the structure of the adjacency matrix. Furthermore, it demonstrates that good performance can be achieved with limited prior knowledge of the task and class relationships.

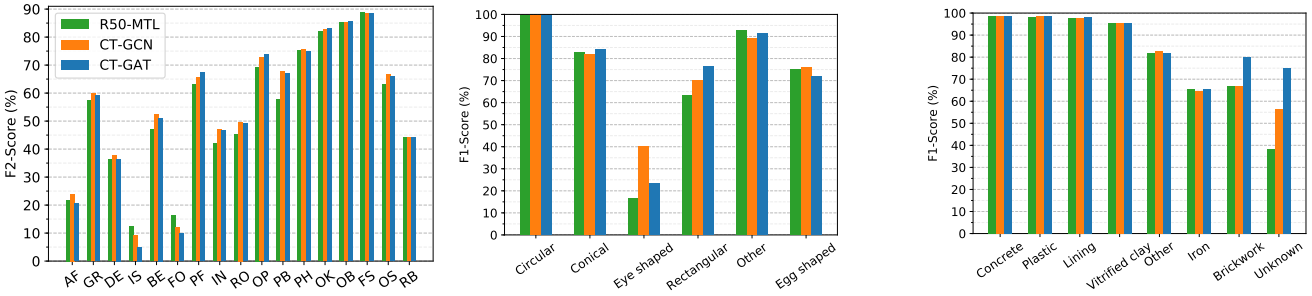
Lastly, we observe that the general performance, as measured by Δ_{MTL}, increases when using MTL networks. By inspecting the results, one can see that the water task performance is not affected by the MTL networks. However, for the defect, material and shape tasks the performance increases dramatically, beating the STL method and benchmark method from [25] by several percentage points, indicating a clear benefit of utilizing an MTL approach. We also observe a clear difference in Δ_{MTL} across the validation and test splits. This is attributed to the shape and material tasks where the classes are very imbalanced, leading to few labels to learn from during training and a potentially large difference between the examples in the different splits.

4.5. Per-task Analysis of Results

To get a better understanding of the performance difference between the CT-GNNs and R50-MTL, we dive into the per-class task performances. Images of the different classes can be found in the supplementary material.

When comparing the individual defect F2-scores, shown in Figure 4a, we see that the CT-GNN performs better on defects with high CIWs but few training examples such as OS, PB, and PS, while the performance is worse on the rare defect classes with a low CIW such as IS and FO. For classes where there are plenty of examples to learn from we observe that the performance is comparable across all models.

When investigating the water task, we observe that all models perform equally well on all classes. On the shape task it is clear the CT-GNN performs better on the rectangular and eye shaped pipes, see Figure 4b. It should be noted that the amount of validation examples of eye shaped pipes is very low. The CT-GNN does, however, achieve a slightly lower F1-score on the egg shaped pipes. On the material task, the CT-GNN again improves performance compared to the baseline, see Figure 4c. By using the CT-GAT performance on the Brickwork and Unknown classes increase by 13 and 37 percentage points, respectively.



(a) Per-class defect performance (b) Per-class shape performance (c) Per-class material performance

Figure 4: **Per task class comparisons.** We compare model performance on the validation set. The F2 defect scores are plotted for each defect class in Figure 4a ordered by increasing CIW from left to right. We refer to the Sewer-ML paper [25] for an explanation of the defect class codes. The class F1-scores for the shape and material tasks are plotted in Figure 4b-4c. The scores are plotted by decreasing number of training samples per class.

4.6. Ablation Studies

Importance of λ_{defect}. The most critical part of an automated sewer inspection system, is the capability to classify the presence of defects correctly. Therefore, we investigate the effect of different λ_{defect} values on the overall performance metric Δ_{MTL}. We compare the performance when setting $\lambda_{\text{defect}} = \{0.25, 0.33, 0.50, 0.67, 0.75, 0.90, 0.95\}$ ranging from an equal weighting between all four tasks ($\lambda_{\text{defect}} = 0.25$) to focusing on the defect task ($\lambda_{\text{defect}} = 0.95$). We train an MTL model with a hard-shared ResNet-50 encoder with and without the CT-GNN decoder heads, see Figure 3. We observe that the Δ_{MTL} increases steadily together with λ_{defect}, peaking at $\lambda_{\text{defect}} = 0.90$, before decreasing when prioritizing the defect task too much when $\lambda_{\text{defect}} = 0.95$.

Combining MTAN and CT-GNN. The combination of soft parameter sharing encoder- and decoder-focused models has not previously been investigated. Therefore, we compare the effect of combining MTAN encoder and the CT-GNN decoder, to determine whether the two approaches are complementary. We find that the CT-GCN and CT-GAT obtains a Δ_{MTL} of 12.72% and 11.48% when trained with MTAN, respectively. This shows that the combination of MTAN and CT-GCN leads to a higher performance with the CT-GCN compared to using a hard-shared encoder. However, when using the CT-GAT the performance decreases. This indicates the GNN settings cannot just be transferred from a hard to soft-shared encoder, instead requiring a small search over how the graph is constructed.
The per-task metric performances for both ablation studies can be found in the supplementary material.

5. Conclusion

One of the most important infrastructures in modern society is the sewerage infrastructure, but it is difficult to inspect and maintain. Automated sewer inspection methods have been investigated for decades, with an emphasis of the important defect classification task, while sewer properties such as water level, pipe material, and pipe shape, which are needed to determine the deterioration level, have been neglected.

We approach the automated sewer inspection problem as a multi-task classification problem. To this end we introduce our novel Cross-Task Graph Neural Network (CT-GNN) Decoder, which utilizes the cross-task information between concurrent and related tasks to refine the per-task predictions. This is realized by generating unique per-class node embeddings that are combined and refined through the use of a graph neural network.

Using our novel method, we not only beat the state-of-the-art on the defect and water level classification tasks by 5.3 and 8.0 percentage points, respectively, but also outperform other single-task and multi-task learning methods on all four classification tasks in the Sewer-ML dataset [25]. Furthermore, the CT-GNN decoder introduces 50 times fewer parameters compared to encoder-focused models.

The novel CT-GNN approach is focused on handling the concurrent image-level classification tasks present in the Sewer-ML dataset. It is, however, important to note that the method is not specific to the sewer data and can therefore be expected to generalize to other domains containing concurrent classification tasks. Another interesting future direction for the CT-GNN is to adapt it to regression tasks where the values cannot be discretized.

Acknowledgments

This research was funded by Innovation Fund Denmark [grant number 8055-00015A] and is part of the Automated Sewer Inspection Robot (ASIR) project, and partially supported by the Spanish project PID2019-105093GB-I00 (MINECO/FEDER, UE), and by ICREA under the ICREA Academia programme.

References

[1] David Alejo, Fernando Caballero, and Luis Merino. Rgbd-based robot localization in sewer networks. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4070–4076, 2017.

[2] David Alejo, Gonzalo Mier, Carlos Marques, Fernando Caballero, Luís Merino, and Paulo Alvito. *SIAR: A Ground Robot Solution for Semi-autonomous Inspection of Visitable Sewers*, pages 275–296. Springer International Publishing, Cham, 2020.

[3] American Society of Civil Engineers. 2017 infrastructure report card - wastewater. `https://www.infrastructurereportcard.org/wp-content/uploads/2017/01/Wastewater-Final.pdf`, 2017.

[4] Chris H. Bahnsen, Anders S. Johansen, Mark P. Philipsen, Jesper W. Henriksen, Kamal Nasrollahi, and Thomas B. Moeslund. 3d sensors for sewer inspection: A quantitative review and analysis. *Sensors*, 21(7):2553, Apr 2021.

[5] David Bruggemann, Menelaos Kanakis, Stamatios Georgoulis, and Luc Van Gool. Automated search for resource-efficient branched multi-task networks. In *Proceedings of the 31st British Machine Vision Conference*, 2020.

[6] David Bruggemann, Menelaos Kanakis, Anton Obukhov, Stamatios Georgoulis, and Luc Van Gool. Exploring relational context for multi-task dense prediction, 2021. arXiv:2104.13874.

[7] Kefan Chen, Hong Hu, Chaozhan Chen, Long Chen, and Caiying He. An intelligent sewer defect detection method based on convolutional neural network. In *2018 IEEE International Conference on Information and Automation (ICIA)*, pages 1301–1306, Aug 2018.

[8] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1979–1986, 2014.

[9] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 794–803, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[10] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2039–2050. Curran Associates, Inc., 2020.

[11] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5172–5181, 2019.

[12] Jack C.P. Cheng and Mingzhu Wang. Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques. *Automation in Construction*, 95:155 – 171, 2018.

[13] Yun Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9260–9269, 2019.

[14] European Committee for Standardization. *Investigation and assessment of drain and sewer systems outside buildings – Part 2: Visual inspection coding system*. Dansk Vand og Spildevandsforening (DANVA), 1 edition, 2011.

[15] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, Sept. 2009.

[16] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L. Yuille. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3200–3209, 2019.

[17] Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. Dynamic task prioritization for multitask learning. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 282–299, Cham, 2018. Springer International Publishing.

[18] Pengxin Guo, Chang Deng, Linjie Xu, Xiaonan Huang, and Yu Zhang. Deep multi-task augmented feature learning via hierarchical graph neural network, 2020. arxiv: 2002.04813.

[19] Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to branch for multi-task learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3854–3863. PMLR, 13–18 Jul 2020.

[20] William L. Hamilton. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159, 2020.

[21] Syed Ibrahim Hassan, L. Minh Dang, Irfan Mehmood, Suhyeon Im, Changho Choi, Jaemo Kang, Young-Soo Park, and Hyeonjoon Moon. Underground sewer pipe condition assessment based on convolutional neural networks. *Automation in Construction*, 106:102849, 2019.

[22] Joakim Bruslund Haurum, Moaaz M. J. Allahham., Mathias S. Lynge., Kasper Schøn Henriksen, Ivan A. Nikolov., and Thomas 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 - Volume 5: VISAPP,*, pages 891–900. INSTICC, SciTePress, 2021.

[23] Joakim Bruslund Haurum, Chris H. Bahnsen, Malte Pedersen, and Thomas B. Moeslund. Water level estimation in sewer pipes using deep convolutional neural networks. *Water*, 12(12), 2020.

[24] Joakim Bruslund Haurum and Thomas B. Moeslund. A survey on image-based automation of cctv and sset sewer inspections. *Automation in Construction*, 111:103061, 2020.

[25] Joakim Bruslund Haurum and Thomas B. Moeslund. Sewer-ml: A multi-label sewer defect classification dataset and benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Jun 2016.

[27] Kasper Schøn Henriksen, Mathias S. Lynge, Mikkel D. B. Jeppesen, Moaaz M. J. Allahham, Ivan A. Nikolov, Joakim Bruslund Haurum, and Thomas B. Moeslund. Generating synthetic point clouds of sewer networks: An initial investigation. In Lucio Tommaso De Paolis and Patrick Bourdot, editors, *Augmented Reality, Virtual Reality, and Computer Graphics*, pages 364–373, Cham, 2020. Springer International Publishing.

[28] Shivprakash Iyer, Sunil K. Sinha, Michael K. Pedrick, and Bernhard R. Tittmann. Evaluation of ultrasonic inspection and imaging systems for concrete pipes. *Automation in Construction*, 22:149 – 164, 2012. Planning Future Cities-Selected papers from the 2010 eCAADe Conference.

[29] Hyon Ji, Sung Yoo, Bong-Jae Lee, Dan Koo, and Jeong-Hee Kang. Measurement of wastewater discharge in sewer pipes using image analysis. *Water*, 12(6):1771, Jun 2020.

[30] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[31] Muhammad Safeer Khan and Rajvardhan Patil. Acoustic characterization of pvc sewer pipes for crack detection using frequency domain analysis. In *2018 IEEE International Smart Cities Conference (ISC2)*, pages 1–5, 2018.

[32] Muhammad Safeer Khan and Rajvardhan Patil. Statistical analysis of acoustic response of pvc pipes for crack detection. In *SoutheastCon 2018*, pages 1–5, 2018.

[33] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[34] Srinath S. Kumar, Dulcy M. Abraham, Mohammad R. Jahanshahi, Tom Iseley, and Justin Starr. Automated defect classification in sewer closed circuit television inspections using deep convolutional neural networks. *Automation in Construction*, 91:273 – 283, 2018.

[35] Srinath Shiv Kumar, Mingzhu Wang, Dulcy M. Abraham, Mohammad R. Jahanshahi, Tom Iseley, and Jack C. P. Cheng. Deep learning-based automated detection of sewer defects in cctv videos. *Journal of Computing in Civil Engineering*, 34(1):04019047, 2020.

[36] Johannes Kunzel, Thomas Werner, Peter Eisert, and Jan Waschnewski. Automatic analysis of sewer pipes based on unrolled monocular fisheye images. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2019–2027, 2018.

[37] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[38] Mathieu Lepot, Nikola Stanić, and François H.L.R. Clemens. A technology for sewer pipe inspection (part 2): Experimental assessment of a new laser profiler for sewer defect detection and quantification. *Automation in Construction*, 73:1 – 11, 2017.

[39] Duanshun Li, Anran Cong, and Shuai Guo. Sewer damage detection from imbalanced cctv inspection data using deep convolutional neural networks with hierarchical classification. *Automation in Construction*, 101:199 – 208, 2019.

[40] Jianshu Li, Pan Zhou, Yunpeng Chen, Jian Zhao, Sujoy Roy, Yan Shuicheng, Jiashi Feng, and Terence Sim. Task relation networks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 932–940, 2019.

[41] Lukas Liebel and Marco Körner. Auxiliary tasks in multi-task learning, 2018. arxiv: 1805.06334.

[42] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[43] Pengfei Liu, Jie Fu, Yue Dong, Xipeng Qiu, and Jackie Chi Kit Cheung. Learning multi-task communication with message passing for sequence learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4360–4367, Jul. 2019.

[44] Shikun Liu, Edward. Johns, and Andrew J. Davison. End-to-end multi-task learning with attention. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1871–1880, 2019.

[45] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[46] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Philip S Yu. Learning multiple tasks with multilinear relationship networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[47] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio Feris. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[48] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[49] Dirk Meijer, Lisa Scholten, Francois Clemens, and Arno Knobbe. A defect classification methodology for sewer image sets with convolutional neural networks. *Automation in Construction*, 104:281 – 298, 2019.

[50] Zihang Meng, Nagesh Adluru, Hyunwoo J. Kim, Glenn Fung, and Vikas Singh. Efficient relative attribute learning using graph neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[51] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning.

In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3994–4003, 2016.

[52] Saeed Moradi, Tarek Zayed, Fuzhan Nasiri, and Farzaneh Golkhoo. Automated anomaly detection and localization in sewer inspection videos using proportional data modeling and deep learning-based text recognition. *Journal of Infrastructure Systems*, 26(3):04020018, 2020.

[53] Gang Pan, Yaoxian Zheng, Shuai Guo, and Yaozhi Lv. Automatic sewer pipe defect semantic segmentation based on improved u-net. *Automation in Construction*, 119:103383, 2020.

[54] Claudio Piciarelli, Danilo Avola, Daniele Pannone, and Gian Luca Foresti. A vision-based system for internal pipeline inspection. *IEEE Transactions on Industrial Informatics*, 15(6):3289–3299, 2019.

[55] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[56] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Dex: Deep expectation of apparent age from a single image. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, December 2015.

[57] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, 126(2-4):144–157, 2018.

[58] Sebastian Ruder. An overview of multi-task learning in deep neural networks, 2017. arxiv: 1706.05098.

[59] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Latent multi-task architecture learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4822–4829, Jul. 2019.

[60] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 525–536. Curran Associates, Inc., 2018.

[61] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 746–760, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[62] Trevor Standley, Amir R. Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *Proceedings of 37th International Conference on Machine Learning*, 2020.

[63] Fariborz Taherkhani, Ali Dabouei, Sobhan Soleymani, Jeremy Dawson, and Nasser M. Nasrabadi. Tasks structure regularization in multi-task learning for improving facial attribute prediction, 2021. arXiv:2108.04353.

[64] Abbasali Dehghan Tezerjani, Mehran Mehrandezh, and Raman Paranjape. Defect detection in pipes using a mobile laser-optics technology and digital geometry. *MATEC Web of Conferences*, 32:06006, 2015.

[65] Simon Vandenhende, Stamatios Georgoulis, Bert De Brabandere, and Luc Van Gool. Branched multi-task networks: Deciding what layers to share. In *Proceedings of the 31st British Machine Vision Conference*, 2020.

[66] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

[67] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 527–543, Cham, 2020. Springer International Publishing.

[68] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.

[69] Mingzhu Wang and Jack 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):162–177, 2020.

[70] Mingzhu Wang, Srinath Shiv Kumar, and Jack C.P. Cheng. Automated sewer pipe defect tracking in cctv videos based on defect detection and metric learning. *Automation in Construction*, 121:103438, 2021.

[71] Qian Xie, Dawei Li, Jinxuan Xu, Zhenghao Yu, and Jun Wang. Automatic detection and classification of sewer defects via hierarchical deep learning. *IEEE Transactions on Automation Science and Engineering*, pages 1–12, 2019.

[72] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nieu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 675–684, 2018.

[73] Yongxin Yang and Timothy M. Hospedales. Deep multi-task representation learning: A tensor factorisation approach. In *5th International Conference on Learning Representations, ICLR*, 2017.

[74] Xianfei Yin, Yuan Chen, Ahmed Bouferguene, Hamid Zaman, Mohamed Al-Hussein, and Luke Kurach. A deep learning-based framework for an automated defect detection system for sewer pipes. *Automation in Construction*, 109:102967, 2020.

[75] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5824–5836. Curran Associates, Inc., 2020.

[76] Aamir R. Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J. Guibas. Robust learning through cross-task consistency. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11194–11203, 2020.

[77] Amir R. Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[78] Yu Zhang and Qiang Yang. A survey on multi-task learning, 2018. arxiv: 1707.08114.

[79] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Zequn Jie, Xiang Li, and Jian Yang. Joint task-recursive learning for semantic segmentation and depth estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[80] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nieu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4101–4110, 2019.

[81] Ling Zhou, Zhen Cui, Chunyan Xu, Zhenyu Zhang, Chaoqun Wang, Tong Zhang, and Jian Yang. Pattern-structure diffusion for multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.