Automated detection and segmentation of cracks in concrete surfaces using joined segmentation and classification deep neural network

Domen Tabernik[a], Matic Šuc[a], Danijel Skočaj[a]

[a]*Faculty of Computer and Information Science, University of Ljubljana, Vecna pot 113, Ljubljana, Slovenia*

Abstract

Automated quality control of pavement and concrete surfaces is essential for maintaining structural integrity and consistency in the construction and infrastructure industries. This paper presents a novel deep learning model designed for automated quality control of these surfaces during both construction and maintenance phases. The model employs per-pixel segmentation and per-image classification, integrating both local and broader context information. Additionally, we utilize the classification results to improve segmentation during both training and inference stages. We evaluated the proposed model on a publicly available dataset containing more than 7,000 images of pavement and concrete cracks. The model achieved a Dice score of 81% and an intersection-over-union of 71%, surpassing publicly available state-of-the-art methods by at least 6-7 percentage points. An ablation study confirms that leveraging classification information enhances overall segmentation performance. Furthermore, our model is computationally efficient, processing over 30 FPS for 512×512 images, making it suitable for real-time applications on medium-resolution images. Code and the corrected dataset ground truths are publicly available: `https://github.com/vicoslab/segdec-net-plusplus-conbuildmat2023.git`

Keywords: concrete crack segmentation, deep learning, encoder-decoder architecture, automated quality control, joint segmentation and classification

1. Introduction

Detection of visual cracks in concrete surfaces and pavements is a crucial step for various construction and maintenance processes in civil engineering. This process is often done manually which can be tedious due to the potentially high number of surfaces that require inspection [1]. Moreover, inspection needs to be done multiple times at different stages of the construction phases as well as periodically during normal use to monitor for the deterioration of the concrete structure. This makes the process time-consuming and costly. Automating this process with a vision-based system mounted on potentially autonomous vehicles has the potential to reduce labor costs while also ensuring more consistent quality control [2].

In this work, we focus on segmentation of pavement and concrete cracks from visual data as a form of concrete defect detection for quality control in construction processes. Although, a per-image based detection [1], i.e., a mere indication of a presence of a crack in an image, or a bounding box detention [3] may be sufficient for some tasks, having a more accurate per-pixel detection is more advantageous as this can be useful for further analysis of the severity of the crack. Some recent works also include 3D concrete crack segmentation based on 3D reconstruction [4] or X-Ray CT scans and volumetric data [5].

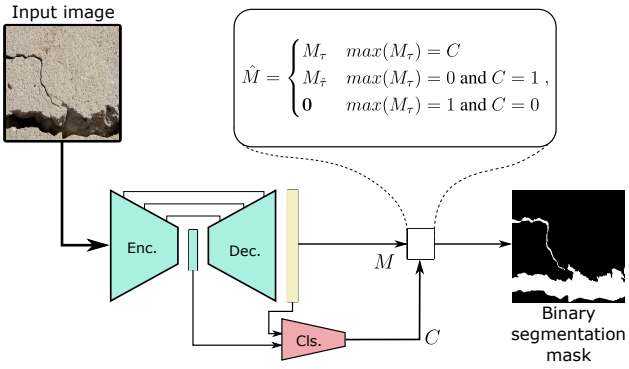

Figure 1: Overview of our proposed approach.

Per-pixel detection of surface defects is often best addressed with a supervised binary segmentation model. Semi-supervised reconstruction-based segmentation models that are trained without labels on good samples have also been explored for concrete crack segmentation [6], but they often cannot achieve the same performance as fully supervised models. On the other hand, supervised binary segmentation models have been successfully deployed for surface crack detection on concrete [4, 5, 7] and various other construction materials [8–11] as well as to other materials, such as steel [12]. Although segmentation models with a dense prediction appear as an ideal solution for per-pixel detection of cracks, this can be quite prob-

Email addresses: `domen.tabernik@fri.uni-lj.si` (Domen Tabernik), `ms0181@student.uni-lj.si` (Matic Šuc), `danijel.skocaj@fri.uni-lj.si` (Danijel Skočaj)

lematic in some defect detection problems due to too large emphasis on the per-pixel accuracy while ignoring wider context information. Since each per-pixel decision is trained (almost) independently of other pixels, wider context information is often ignored, such as the shape or presence of other defects farther away from the pixel.

On the other hand, a per-image-based detection can take into account wider context information and may not be susceptible to a per-pixel-based noise. Our previous work in the domain of industrial vision inspection [13, 14] explored a per-image defect detection with a network that also included segmentation in the loss, thus forcing to use both per-pixel and wider-context information. Although this method achieved state-of-the-art results on a per-image basis, this was never reflected in a per-pixel accuracy, since the segmentation was used only as an auxiliary objective and the main objective remained per-image-based detection. Moreover, segmentation was performed on a reduced resolution resulting in poor segmentation for thin defects.

In this work, we propose a novel method for segmentation of pavement and concrete surface cracks, termed SegDecNet++, that outperforms all existing concrete crack segmentation methods in both per-image classification and per-pixel segmentation, while at the same time enjoying lower computational cost. We propose to achieve this by jointly training a segmentation model for both per-pixel segmentation as well as for per-image classification to determine whether there is any defect present in the image or not. For this, we build on top of our previous model for a per-image defect detection [13, 14], but propose a redesigned architecture that focuses on improved segmentation accuracy in concrete cracks. Furthermore, we propose to improve the segmentation of defects during the inference time by including the classification decision in the final construction of the segmentation mask and ensuring consistency between the classification and segmentation outputs. We achieve this consistency by adjusting the segmentation threshold based on the predicted classification output. We evaluate our model on a large dataset of pavement and concrete cracks [15], demonstrating state-of-the-art performance for crack segmentation at a low computational cost. We provide an ablation study to demonstrate the effect of our proposed architecture choices, the effect of enforcing the consistency between the classification and segmentation outputs during the inference, and performance on different types of surfaces and cracks.

The remainder of this paper is structured as follows: we discuss the related work in Section 2 and present a detailed description of the proposed method in Section 3. We present the experimental validation of the proposed method in Section 4 and conclude with a discussion in Section 5.

2. Related Work

Deep learning in construction industry. Deep learning solutions are being explored for a wide range of tasks in the construction industry. In structural health monitoring, deep learning is used to identify and locate structural damage in buildings [16–18] and bridges [19–21]. These include detection and analysis of cracks [21–23], corrosion [24], loose bolts [25], pipe defects [26], potholes [27], and vibration [17]. Object detection is also being employed in the construction sites to detect equipment and material [28] to monitor and improve the efficiency of the construction process, as well as to improve safety by detecting personal protective equipment [29]. In material characterization, deep learning is being used for detection of micro-cracks in cementitious composites [30], as well as for detection and analysis of particles [31], pore structures [32] and air-voids [4, 33] of different construction material. Deep learning also plays an important role in understanding strain-hardening cementitious composites through analysis of micro-tomography data by detecting particles and fibers [34, 35]. In planing and construction phase, deep learning can also improve robotic additive manufacturing processes by analyzing the quality of concrete layer extrusion [25, 36], identify rooftop thermal bridges [37] to improve thermal insulation of buildings, and even automatically generate building layouts [38] with generative deep learning approaches. In this work, we focus on deep learning applications for crack segmentation on pavements and concrete surfaces using RGB images.

Concrete and pavement crack segmentation. Early traditional approaches to crack segmentation relied on morphological operations with edge detectors designed for cracks in sewer pipes [39], and histogram thresholding with snake models [40] or Random Structured Forests [10] for road pavements crack detection. The introduction of feature learning with deep models has quickly overtaken traditional approaches [8]. The most commonly used deep architecture is the U-Net with an encoder-decoder architecture and skip-connections, which has been particularly successful in crack segmentation on roads and pavements [41]. Some work also relied on features pre-trained on unrelated tasks to improve performance for crack detection and segmentation. Pre-trained VGG16 features are commonly used in concrete crack segmentation of various thickness [42], for example, with pyramid networks for pavement cracks [11], or in fusion of features at multiple levels for detection of mid and thick cracks on asphalt and concrete surfaces with different textures (bare, rough, dirty) [43]. Larger architectures have also been evaluated. Ni et al. [44] applied the GoogLeNet inception model trained for a classification task of various concrete cracks and combined it with an additional crack delineation network for per-pixel segmentation, while Lau et al. [45] integrated a ResNet-34 encoder into a U-Net-based architecture for application on pavement cracks.

Off-the-shelf architecture is also commonly used in some recent works in the civil engineering literature. He et al. [46] evaluated Faster R-CNN with VGG16 backbone for road-based crack detection from UAVs. This work focused on cracks captured on streets in China with small amount of road sundries or background interference by a mobile phone for ground-level images and by UAVs for aerial perspectives. Loverdos and Sarhosis [47] conducted a comprehensive evaluation of standard architectures for crack detection in masonry walls containing various types of bricks of regular pattern, different illumination and different angles. The evaluation included U-Net, DeepLabV3, LinkNet, and FPN architectures trained under dif-

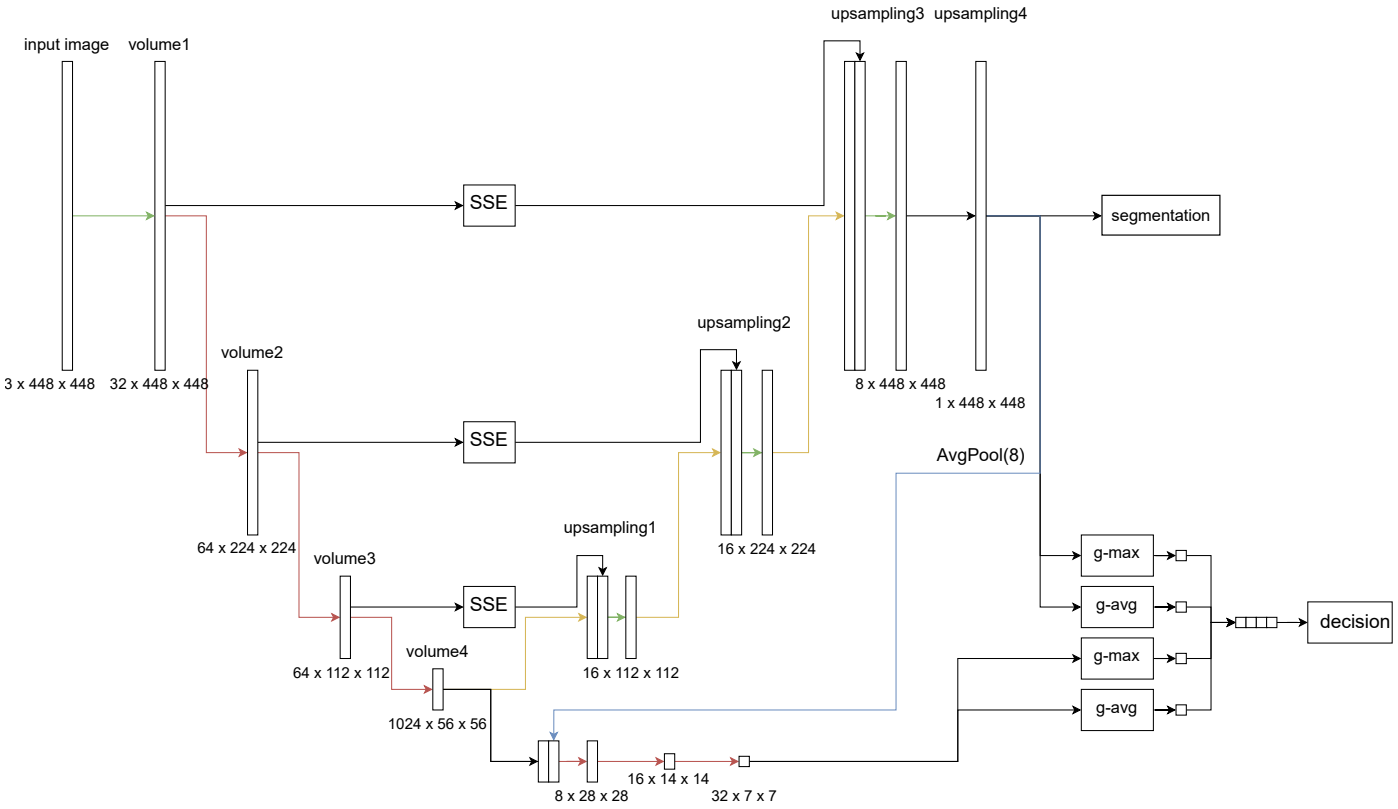

Figure 2: Architecture of the proposed model with an encoder-decoder for a per-pixel segmentation and additional layers for a per-image classification.

ferent combinations of hyperparameters. For real-time performance, some recent works also used YOLO-based architectures for crack detection in asphalt and concrete pavements [48], and for crack detection in tiled sidewalks from UAV [49]. Zhang et al. [50] instead used MobileNet for concrete surface crack detection. Recently, Dong et al. [51] also evaluated the capsule network originally introduced by Hinton [52] and proposed augmenting training images by generating novel images of cracks using StyleGAN [53]. They applied this model for segmentation of pavement cracks considering several different types of cracks (transverse, longitudinal, cross, map cracks, and potholes).

Many approaches further refined U-Net-style architectures. Zou et al. [54] proposed to replace skip-connections and instead of passing encoder features with decoder both encoder and decoder features from specific layers are fused and then passed to a separate stack of layers created from the concatenation of fused encoder-decoder layers. They applied this model to road pavements and stone cracks segmentation. For application in road pavements, Han et al. [55] followed the idea of DenseNet and added a fusion of channels from multiple levels of encoder features before they were passed to decoder layers to improve the ability of different receptive fields to perceive information at different scales. Also for road pavements, Li et al. [56] proposed to use residual blocks as encoder and decoder in U-Net , while combining the Inception-Resnet module and atrous convolution to obtain multi-scale features. They also proposed using Discrete Wavelet Transform (DWT) for feature map down-sampling to reduce information loss compared to pooling layers. Li et al. [15] proposed SCCDNet that incorporates the Squeeze-and-Excitation layer from [57] between skip-connections of U-Net architecture. They applied this to asphalt and road pavements of different brightness and contrast as well as to concrete walls from building and structures. Recently, Xiang et al. [58] proposed CDU-Net for cracks segmentation in concrete structures that includes multiple pathways in processing of encoded U-Net features. CDU-Net was combined with super-resolution to improve micro-crack detection. Recent work contemporaneous with ours [59] also employed U-Net-style architecture, while proposing several improvements to capture wide-context information for application on segmentation of various tunnel lining defect such as cracks, leakages, peeling fireproof coating, and other defects in highway tunnels. They apply cross-attention before skip connections and use atreus spatial pyramid pooling (ASPP) with spatial attention between encoder and decoder.

Finally, several recent work applied Transformers to crack detection and segmentation. Transformers inherently account for global context information since their receptive field encompasses the whole image. Liu et al. [60] proposed CrackFormer for a fine-grained crack segmentation applied to thin cracks on road pavements with asphalt and stone. This model combines SegNet-like architecture with attention mechanisms that can fully extract contextual information using large receptive fields.

Swin-based Transformer combined with SegFormer-based encoder was used in [61] for pavement crack segmentation that was evaluated on examples with heavy shadow and dense crack generated in the commonly negated application (e.g., low volume, local roads). For application on asphalt and concrete surfaces from building with cracks on different surface finishes in varying illumination conditions, [62] proposed TransUNet, which applies ViT-based global self-attention to hidden features of U-Net architecture.

Crack detection has also been explored as part of an industrial surface defect detection in inspection tasks. Joint segmentation and classification tasks for defect detection of thin cracks in surfaces of electrical commutators were explored by our previous work [13, 63] in a two-stage architecture, where per-pixel defect segmentation was used in the first stage during training as an auxiliary task to facilitate learning of the encoder features from a limited set of images. Simultaneous end-to-end model for joint learning of both tasks was performed in [14, 64], however, the performance of segmentation task was never evaluated on a per-pixels basis and instead only per-image classification was performed.

Comparison to related approaches. Combining classification and segmentation for defect detection has been explored in our previous work [13, 63] in a two-stage architecture and the architecture's end-to-end implementation [14]. However, this architecture was not purposefully designed for pixel-level segmentation accuracy and thus contains only a single-layer decoder. We propose to use a larger decoder with higher output resolution, resulting in a U-Net-style architecture. Moreover, we include Skip-Squeeze-and-Excitation (SSE) that implements skip-connections with Squeeze-and-Excitation blocks (SE).

Our U-Net architecture for segmentation is also related to SCCDNet [15], which uses encoder-decoder architecture and SSE as skip connections. However, their model does not include a per-image classification at all and contains only a segmentation module. Convolutional blocks in our encoder-decoder also follow the design of encoder from [14] with fewer channels but slightly larger kernel size than the ones used in [15]. Recent work contemporaneous with ours [59] also incorporated wide-context information but using cross-attention and atrous spatial pyramid pooling with spatial attention to achieve this. Their work also focuses only on segmentation and does not perform an explicit per-image classification as ours.

Combining segmentation and classification for crack segmentation has also been explored by [44], however, their architecture is primarily based on classification by using GoogLeNet inception model, while segmentation is only added on top of the classification module, as additional decoder layers. We instead use encoder-decoder architecture explicitly designed for segmentation and attach classification on top of it. Additionally, their classification head is composed of a single layer of fully connected neurons, while our proposed classification head follows the design of [14] with multiple additional layers and includes features from the segmentation output. More importantly, they do not utilize classification information during inference time to improve the segmentation accuracy as we do.

3. SegDecNet++ Architecture

Our proposed method uses an encoder-decoder architecture for the dense pixel-wise prediction that is combined with the classification module for per-image classification. The proposed design follows our previous end-to-end architecture [14] termed SegDecNet for per-image defect detection with segmentation, while we also propose an architectural overhaul that is needed for accurate segmentation. We also provide a mechanism to incorporate inferred classification information into the segmentation output. In the following subsections, we provide more detailed information on the segmentation and classification modules. We term our proposed method as SegDecNet++.

3.1. Segmentation module

For the segmentation module, we utilize an encoder-decoder architecture with four convolutional blocks for the encoder and four for the decoder. After each block, we perform down-sampling during the encoding and up-sampling during the decoding. A more detailed specification of the network is shown in Figure 2.

Encoder. Architecture of the convolutional blocks for the encoder follows the design of our previous work [14], with three, four, and one Conv2D operations (Conv2d with feature normalization and ReLu) for the corresponding features at 1×, 2× and 4× reduced resolution, and using 32, 64 and 64 channels, respectively. All convolutions are performed using kernels of the size 5×5 to extend the receptive field as far as possible. The last encoder features contain 1024 channels with 8× reduced resolution compared to the original image.

Decoder. Existing decoder from [14] is inadequate for segmentation task. We instead use a decoder of the same depth as the encoder but in a reversed order and with a transpose 2D convolution for the up-sampling operation instead of down-sampling. The number of Conv2D operations for the convolutional blocks in the first, second, and third layer is therefore 16, 16, and 8 channels, respectively. This results in a U-Net-like architecture.

Skip-connections. We also utilize U-Net-style skip connections and connect outputs from each encoder block to the inputs into the corresponding decoder block at the same level of resolution. We use Skip-Squeeze-and-Excitation (SSE) [15] that implements Squeeze-and-Excitation (SE) blocks between skip connections to increase the information content, as this has proven beneficial in recent architectures for crack segmentation [15].

Segmentation output. Finally, 1×1 convolution is applied to the final decoder features to produce a single-channel output, representing a pixel-wise probability of a crack. To obtain the final binary segmentation mask, we apply a threshold τ to the pixel-wise probability output.

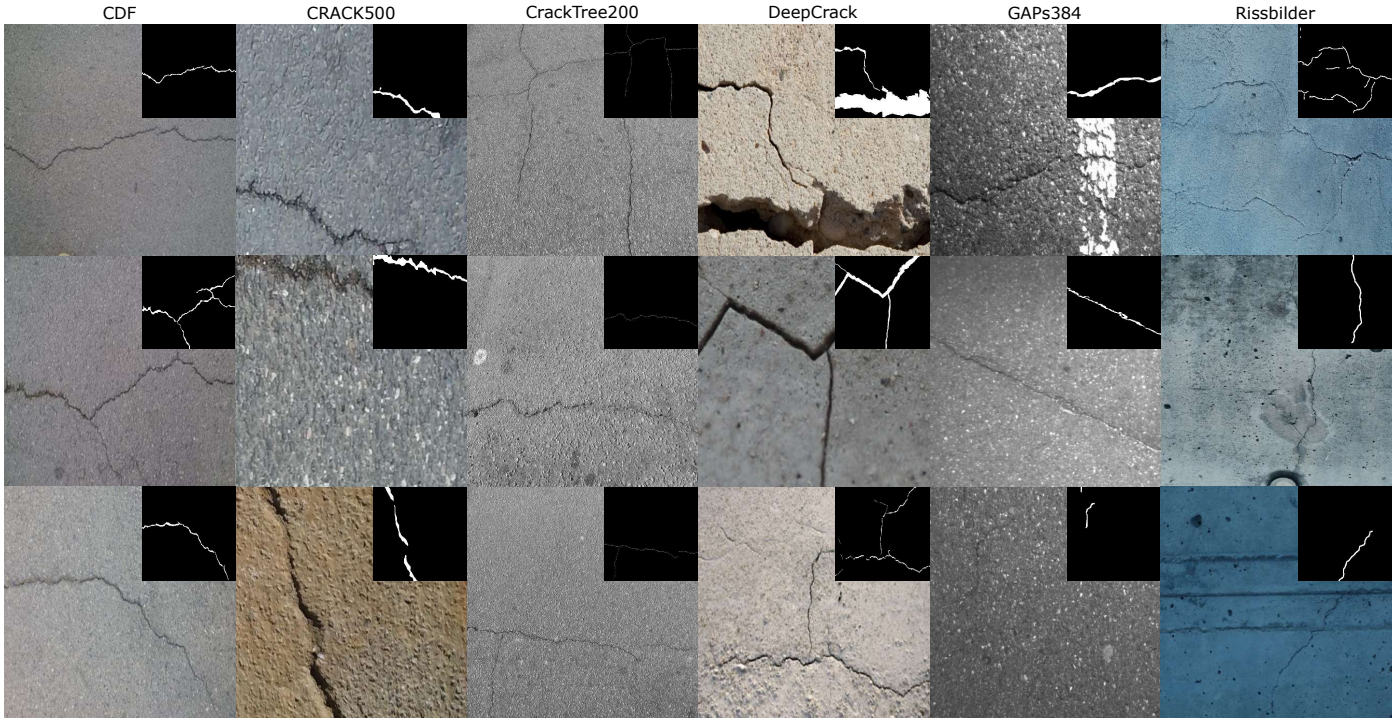

Figure 3: Examples of images and segmentation masks from the combined evaluation dataset [15] based on individual data source.

3.2. Classification module

Per-image classification module is added on top of the encoder-decoder architecture for segmentation to further extract wider context information about the presence of the surface crack. We retain the same design of the classification module from Božič et al. [14], which has proven useful when combining classification and segmentation for defect detection. Classification layers are applied on top of the encoder features, while also using down-sampled final segmentation output as an additional input channel. This results in 1025-channel input features with 16x smaller resolution than the original image. Features are further processed with three convolution blocks, each containing one Conv2D operation (Conv2d with feature normalization and ReLu) with 8, 16, and 32 output channels, respectively. Finally, the global max and average pooling of both final classification features as well as of the final segmentation output are concatenated into 66 output neurons. An additional fully connected layer is applied to obtain the final binary output probability for the whole image.

3.3. Improving per-pixel segmentation with per-image classification

We propose to utilize the information from the per-image classification to further improve the segmentation accuracy. The classification and segmentation outputs may not always agree. For instance, when a classification network determines that there is no crack present in the image then the segmentation network should output an empty segmentation mask. If this does not happen, then either the segmentation or the classification output is incorrect. In our experience, classification has often proven to be more accurate than segmentation thus trusting classification and adjusting the segmentation output by considering the classification decision is advantageous, as shown by our ablation study (see, Table 5).

To ensure consistency between the classification and the segmentation modules, we propose adjusting the segmentation output. We incorporate the per-image classification information for an additional adjustment of the final threshold that is used to produce the final segmentation mask $\hat{M}$:

$$\hat{M} = \begin{cases} M_\tau & max(M_\tau) = C \\ M_{\hat{\tau}} & max(M_\tau) = 0 \text{ and } C = 1 \\ \mathbf{0} & max(M_\tau) = 1 \text{ and } C = 0 \end{cases}, \quad (1)$$

where $C \in \{0, 1\}$ is a per-image binary classification output, $M \in \mathbb{R}^{n \times m}$ is a per-pixel output segmentation probability, τ is the segmentation mask threshold, $M_\tau \in \{0, 1\}^{n \times m}$ is a binary segmentation mask, obtained by thresholding M at the threshold τ, and $\hat{\tau}$ is an adjusted threshold. The operation $max(.)$ serves as a segmentation-based classifier; if there is any defective pixel present, the image is classified as defective.

The proposed mechanism results in two important adjustments. When the classification output determines that there is no crack in the image, then the segmentation mask is zeroed. When the classification head determines that there is a crack in the image, but the obtained segmentation mask is empty, then the threshold τ is adjusted to $\hat{\tau} = 0.9 \cdot \tau$. The segmentation threshold is therefore lowered to include the pixels that are most likely to be defective in the particular image into the segmentation mask. We selected the factor of 0.9 based on our prior experiments applied to independent internal data.

	Total	*Train*	*Test*	*Thickness*			*Brightness*		*Contrast ratio*	*Surface and crack types*
				≤ 3 *px*	*Mid*	≥ 8 *px*	*Non-crack*	*Crack*		
CFD [10]	228	192	36	38%	56.00%	5%	135.29	111.90	0.83	Light urban road surface.
CRACK500 [11]	478	478	0	8%	36.00%	56%	130.30	73.67	0.56	Dark and brown/light pavements.
CrackTree200 [65]	206	175	31	100%	0.00%	0%	136.45	96.37	0.70	Light pavements, thin cracks.
DeepCrack [43]	515	438	77	22%	35.00%	43%	151.83	68.97	0.46	78% concrete (walls), 22% asphalt.
GAPs384 [66]	509	433	76	23%	59.00%	18%	102.24	80.82	0.80	Dark asphalt, soft cracks.
Rissbilder [67]	3822	3249	573	4%	94.00%	3%	112.39	99.45	0.89	Buildings, structures, walls.
Non-crack [68]	1411	1199	212	-	-	-	159.52	-	-	Bridge deck, walls, pavements.
Total	7169	6164	1005	12%	76%	12%	118.31	93.28	0.80	-

Table 1: Detailed statistics of the evaluation dataset compiled by Li et al. [15]

Figure 4: Example of non-crack images from the evaluation dataset [15].

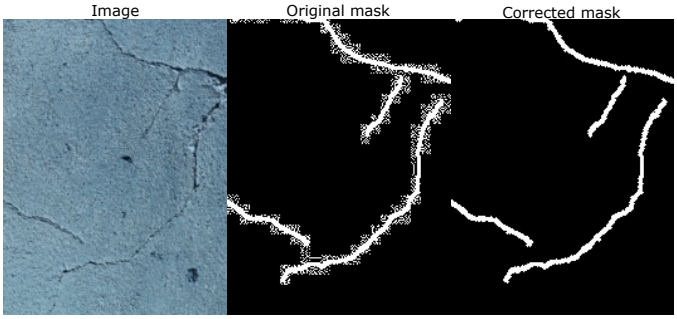

Figure 5: An example of dataset image with the original annotation containing artifacts (middle) and annotation after our correction (right).

4. Experiments

In this section, we present our experimental evaluation of the proposed method for the surface crack segmentation task. We compare our method to several related state-of-the-art approaches with publicly available code. We also provide an ablation study with additional details on how much does classification help to improve the segmentation and what is the computational cost of our proposed method.

4.1. Dataset

Although many datasets for pixel-level crack segmentation have been proposed in recent years, there is still a lack of a consistent evaluation for different methods on all public datasets, thus making comparisons between networks difficult. Majority of previous papers train and evaluate their networks on small datasets (with less than 500 images) established by themselves, which cannot verify the generalization ability of the model. For a fair comparison of our method, we train and evaluate our network on one of the largest datasets proposed by the authors in [15] which they compiled from previously published datasets [10, 11, 43, 65–68], containing crack of road pavements, asphalt and concrete structures and walls.

Dataset [15] consists of 7169 images with manually annotated labels with a resolution of 448×448 pixels, making it one of the biggest crack segmentation datasets. It contains crack images of different environments, different forms, and different shooting distances, covering the common crack characteristics to the greatest extent as shown in Figure 3. The inclusion of negative samples in this dataset, unlike others, makes it appropriate to demonstrate the effectiveness of the mechanism for incorporating the inferred classification information in our proposed method. Some negative samples without cracks are also depicted in Figure 4. Detailed statistics on images and defects for individual subsets is also shown in Table 1. We also calculate brightness of crack and non-cracked pixels, contrast between them and thickness of cracks using skeletonization and distance transform to the edge of mask. We report a percentage of thin (≤ 3 px), thick (≥ 8 px) and medium ($3 \text{ px} < x < 8 \text{ px}$) cracks based on pixels in skeleton.

Groundtruth corrections. While dataset [15] is useful due to its size, we have found that the dataset contains a large number of significant artifacts in the segmentation masks. Artifacts can be found around labels for 3800 out of 7169 segmentation masks, however, they do not correspond to actual defects in the image[1]. We have applied simple morphological operations to remove as many artifacts as possible[2]. On all segmentation masks with visible artifacts we applied a single step of erosion followed

[1]None of the images reported in the original paper or on their website show any similar artifacts, while authors were unable to provide original uncompressed segmentation masks.

[2]Dataset with corrected ground truths is available at `https://go.vicos.si/sccdnetdbcorrected`

Method	Precision [%]	Recall [%]	Dice [%]	IoU [%]
DeepCrack [43]	72.07	77.48	71.59	60.37
CrackFormer [60]	74.99	82.58	75.81	64.72
SCCDNet-D32 [15]	74.50	75.93	73.48	62.24
SegDecNet++ (our)	**78.76 ± 0.34**	**85.53 ± 0.67**	**80.96 ± 0.31**	**70.95 ± 0.37**

Table 2: Segmentation results compared to related methods using train/test split. Best scores are in bold.

Method	Precision [%]	Recall [%]	F1 [%]	FP	FN
DeepCrack [43]	97.01	98.98	97.99	28	8
CrackFormer [60]	99.49	98.98	99.23	4	8
SCCDNet-D32 [15]	98.61	99.36	98.99	11	5
SegDecNet++ (our)	**99.87 ± 0.11**	**99.64 ± 0.15**	**99.76 ± 0.10**	**1.0 ± 0.89**	**2.8 ± 1.17**

Table 3: Results of a per-image defect detection (classification) compared to related methods using train/test split. Best scores are shown in bold.

by a single step of dilation, both using a kernel size of 3×3. This significantly removed compression artifacts and preserved annotation thicknesses. In images were correction method was applied, the removed artifacts on average accounted for 33% of pixels originally marked as defective and less than 1% of all pixels. An example of the mask with artifacts and one after erosion and dilation fixes is shown in Figure 5.

4.2. Evaluation metrics

For our evaluation, we report two metrics that have been established in the literature for the evaluation of per-pixel segmentation: Intersection-over-Union (IoU or Jaccard index) and Dice score. The IoU metric is calculated based on the intersection and union of the groundtruth and predicted mask, while the Dice score is computed on a per-pixel basis as a ratio between twice the number of intersected pixels and the sum of the ground-truth pixels and predicted pixels. This metric is equivalent to F1-measure. We obtained the results for each image individually and averaged them over all images to obtain the final score. We additionally report the results of the evaluation of the classification module in the ablation study. We report this as the number of false positive and false negative images and then calculate precision, recall, and F1-measure.

Note, that some papers report segmentation results at an ideal threshold across the whole dataset (Optimal Dataset Scale - ODS), while others use an ideal threshold for each image (Optimal Image Scale - OIS). In our evaluation, we report ODS, i.e., having a fixed threshold over the whole dataset, but the threshold can also be adjusted based on the classification output. Note, that some papers also ignore 1-2 pixels around labels since often the exact boundary of a crack is difficult to determine. We do not ignore the boundary pixels in our evaluation.

4.3. Implementation details

The proposed architecture is implemented in PyTorch framework[3]. We train segmentation and classification models simultaneously in an end-to-end approach. Both networks use binary cross-entropy loss and are trained using Adam optimizer. We used a learning rate of 0.0001 and applied an additional weighting factor of 0.1 to the classification loss. We used a batch size of 10 and trained all models for 200 epochs. We evaluated every fifth epoch and selected the best model based on the best segmentation score. Since the authors of the dataset did not provide a separate validation set, we used their test set for selecting the best model. During training, we also augmented training data with random horizontal and vertical flips, 180° rotations, and color jittering. Each type of augmentation was applied with a probability of 0.5 and we used brightness, contrast, saturation and hue factors of 0.2 for color jittering.

Related methods. We also had to re-train all related methods due to corrected groundtruth masks in the dataset. We, therefore, included the latest methods for crack segmentation that have publicly available code. We trained each method for at least 100 epochs and selected the best model on a test set. Increasing the number of epochs did not contribute to improving performance. We also applied the same augmentation to related methods where this helped to improve performance.

4.4. Results

Comparison to state-of-the-art. We compared our method against the state-of-the-art methods for crack segmentation. The results for our method obtained with a fixed train/test split are shown in Table 2. We repeated the training of our model five times with different random initializations on the same training data to also evaluate the variance of the obtained results. We

[3] https://github.com/vicoslab/segdec-net-plusplus-conbuildmat2023.git

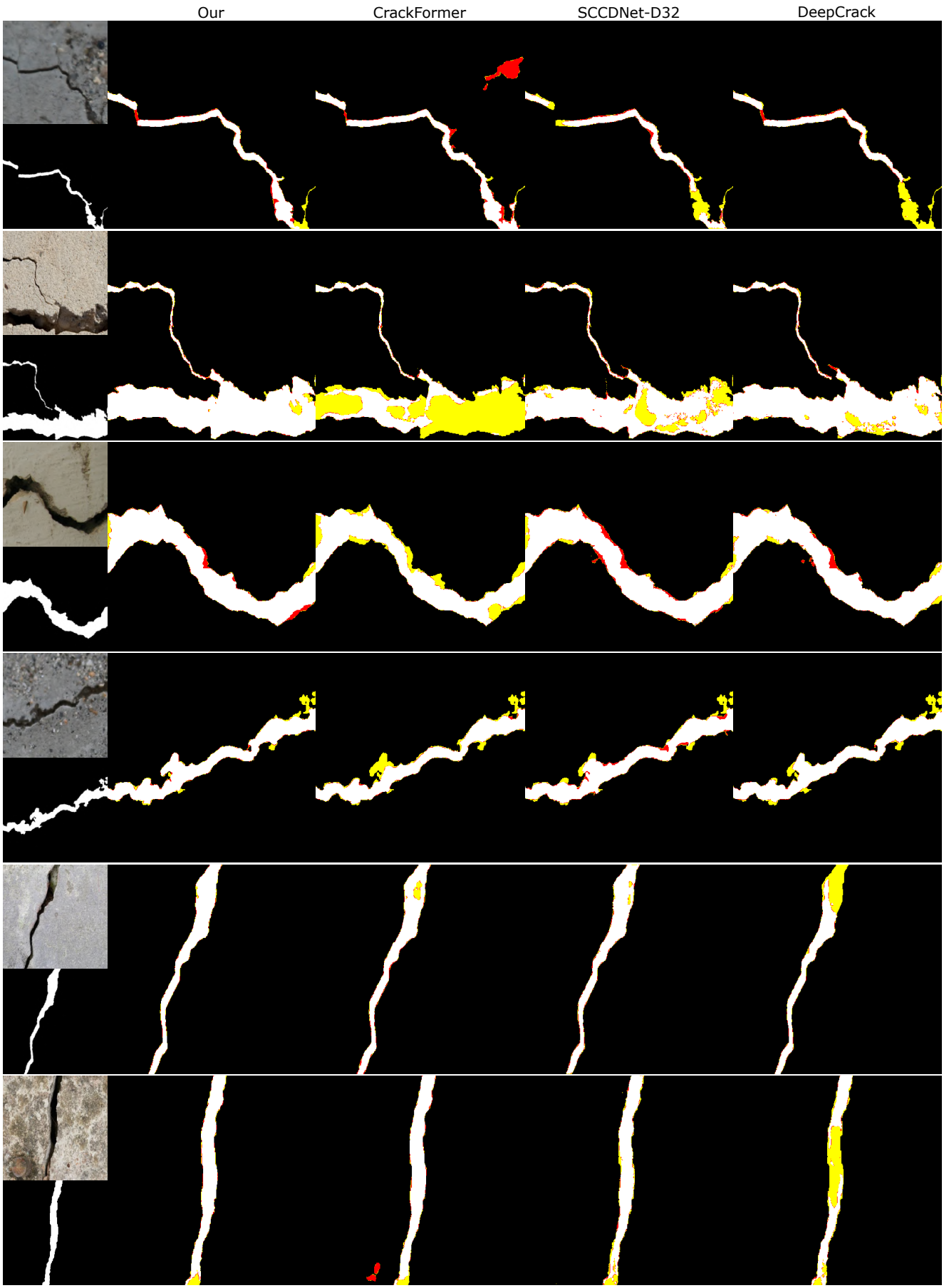

Figure 6: Segmentation examples for our proposed method and all related methods. False positive pixels are depicted in red, false negatives in yellow, while correct background segmentation is in black and correct foreground in white.

	Decoder	*SSE*	*Cls. training*	*Seg. adjust.*	*Dice [%]*	*IoU [%]*
SegDecNet [14]			✓		66.88 ± 0.21	54.49 ± 0.17
+ decoder for segmentation (w/o cls. training)	✓				77.95 ± 0.72	67.75 ± 0.84
+ SSE skip connections (w/o cls. training)	✓	✓			79.55 ± 0.62	69.51 ± 0.67
+ per-image cls. training	✓	✓	✓		80.09 ± 0.76	70.11 ± 0.82
+ seg. mask adjustment (SegDecNet++)	✓	✓	✓	✓	**80.96 ± 0.31**	**70.95 ± 0.37**

Table 4: Segmentation results when incrementally including our proposed architectural choices.

therefore report the mean and standard deviation for the proposed method.

The proposed method is able to out-compete all related state-of-the-art methods such as DeepCrack [43], a U-Net-style encoder-decoder model with SSE, SCCNet-D32 [15], and a Transformer-based model, CrackFormer [60]. Overall, our proposed method achieved segmentation accuracy of 80.9% and 70.9% in Dice score and Intersection-over-Union, respectively. Compared to the second-best method (CrackFormer), this represents a 6 percentage points better Dice score and 7 percentage points better intersection-over-union (IoU). Several examples of segmentation outputs for each evaluated method are shown in Figure 6. We can notice that CrackFormer still has some false positives in the background (shown in red color), while in some cases it misses certain larger areas of cracks (shown in yellow color). Some additional examples of crack segmentation with our proposed model are also shown in Figure 9 at the end of the paper.

Per-image classification by segmentation. For quality control inspection in the construction setting, it is also important to correctly identify images with cracks to ensure all defects are found regardless of their segmentation accuracy. To better understand how well do evaluated methods detect the presence of a defect, we performed an additional per-image evaluation. This allowed us to measure how many images with cracks were not found and how many background images were misidentified as cracks. We obtained a per-image detection of a defect by using a binary segmentation mask returned from each method, and then considered the presence of the crack when at least one pixel was segmented as a crack. We term this as classification-by-segmentation. Note that, our method can directly output the classification neuron and does not require classification-by-segmentation, however, we performed classification-by-segmentation on our method as well for fair comparison. We measured per-image precision, recall, and F1 score, and report results in Table 3.

We can observe that our proposed method achieves the best result with the highest F1-score at 99.76%. The proposed method had on average around 1-2 false positives (FP) and 2-3 false negatives (FN) out of 1005 test images. This is significantly better than related methods. CrackFormer achieved the second-best result with 4 FP, 8 FN, and an F1-score of 99.23%, while SCCDNet-D32 had an F1-score of 98.99%, which resulted in 11 FP and 5 FN. Results well demonstrate that the proposed method outperforms related methods not just in better segmentation, but also in better identification of defective images. This provides more context for the segmentation error reported before. Segmentation metric, therefore, captures mostly wrong segmentation of pixels around defects but the defects were correctly found. We consider this error as less severe since it is more important to at least identify images with cracks, while slightly less accurate pixel-wise localization does not pose such a significant harm when performing quality control in construction.

		Classification by segmentation	
	Classification	*w/o mask adj.*	*w/ mask adj.*
FP	1.0 ± 0.89	11.4 ± 8.4	1.0 ± 0.89
FN	2.6 ± 1.36	1.4 ± 0.80	2.8 ± 1.17

Table 5: Number of false positive (FP) and false negative (FN) image classifications. Mean and standard deviation over five runs are reported.

	$M = 0 \longmapsto M = 1$	$M = 1 \longmapsto M = 0$
Correct	0	10.4 ± 7.89
Incorrect	0.2 ± 0.4	1.4 ± 1.2

Table 6: Number of images where mask segmentation was correctly or incorrectly adjusted. Mean and standard deviation over five runs are reported.

Ablation study. We also performed an ablation study to determine the performance benefits for segmentation accuracy stemming from the individual architectural choices. In this experiment, we used the same fixed train/test split as in Table 2, repeated experiments five times with different random initialization and reported averaged values in Table 4. We used SegDecNet from our prior work [14] as our baseline. We then incrementally added individual architectural choices, resulting in an increase of the segmentation accuracy. Note that each row in Table 4 includes all architectural choices from all previous rows.

The best results are obtained when all our architectural choices are included in the model. On the other hand, the baseline method achieves the worst result while also significantly underperforming all the related methods reported in Table 2. The poor performance of the baseline model is mostly due to

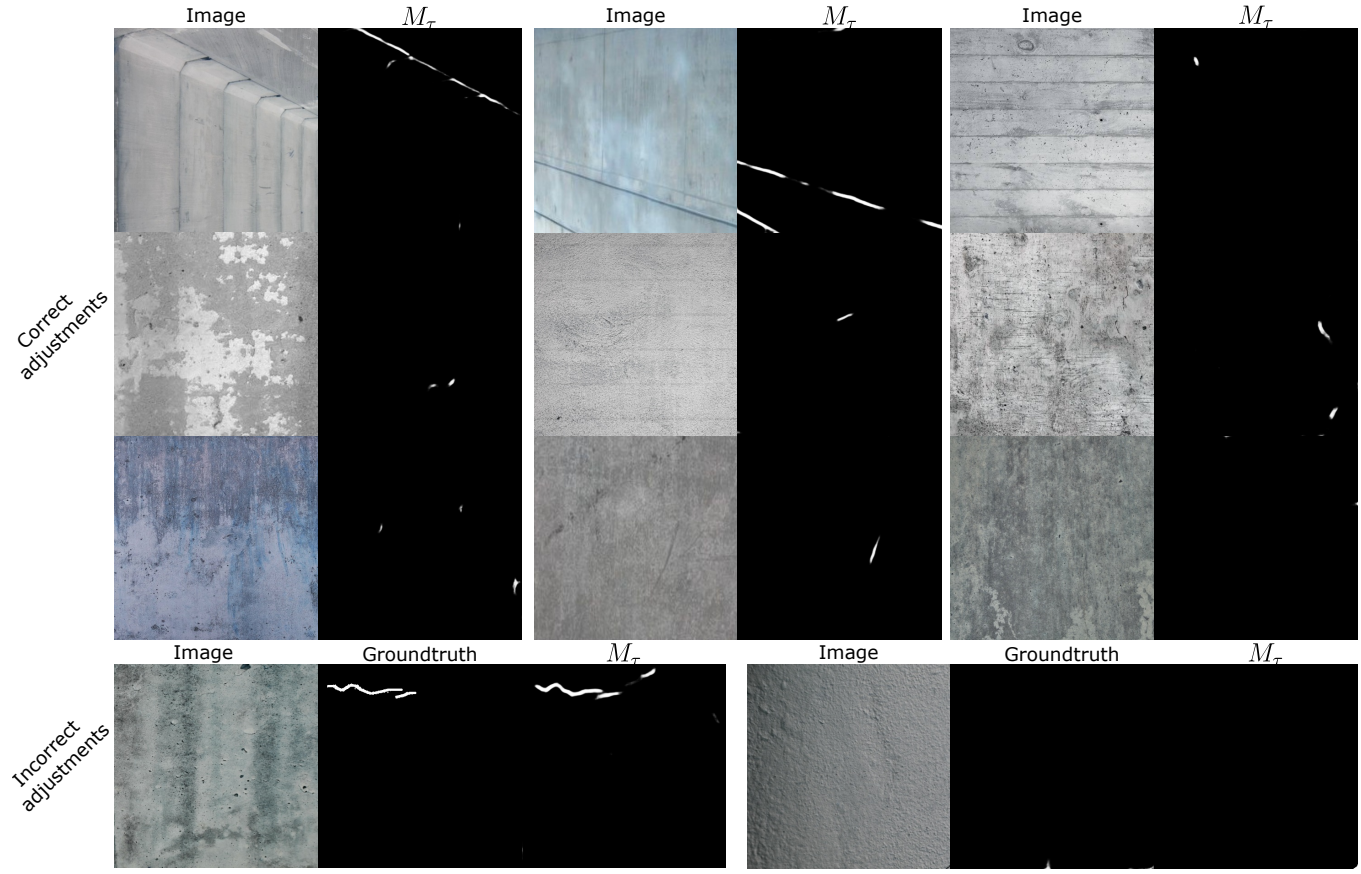

Figure 7: Samples where segmentation masks were adjusted on several non-defective (top) and defective (bottom) images.

inadequate decoder and output resolution. However, when we include our proposed decoder for segmentation with improved resolution, the model becomes competitive with related methods. In this case, segmentation accuracy is improved by 10 percentage points, pushing the model to slightly outperform CrackFormer, the best-performing related method. SSE skip connections, per-image class learning, and segmentation mask adjustments have less of an impact but each change still contributes by around 0.5-2 percentage points, accumulating to around 3-4 percentage points in total, pushing the performance of the proposed method significantly beyond the state-of-the-art.

Classification influence. Results for the ablation study in Table 4 also reveal that including classification information in segmentation has a positive effect on segmentation accuracy. We further analyzed how this is reflected in a per-image classification to determine how well is our model improved by the classification information in concrete crack detection. For positive detection, we considered any image that had at least one segmented pixel in the image, i.e., classification by segmentation. We calculated this before and after applying segmentation mask adjustment by classification output and compared it against using only the classification output neuron.

The results are presented in Table 5. Our analysis shows that classification-by-segmentation without our proposed mask adjustment produces 11.4 (±8.4) false positives and 1.4 (±0.80) false negatives, while directly learned classification produces only 1.0 (±0.89) false positive and 2.6 (±1.36) false negatives. This suggests that many non-defective images are incorrectly identified as defective due to false positive segmentation masks. However, by using our proposed mask adjustment based on the classification neuron, the false positive rate is reduced by 10-fold to only 1.0 (±0.89) out of 1005 testing images.

As shown in Table 6, our mask adjustment is able to on average correctly fix almost all invalid masks in images without visible cracks. A few such examples are depicted in Figure 7. Nevertheless, a few incorrect adjustments to the segmentation mask can also occur. The initial segmentation on actual cracks was incorrectly adjusted to empty mask in 1-2 images, thus resulting in an additional 1-2 false negatives. A few such examples are depicted in the bottom row in Figure 7. However, since the error from classification is significantly lower than from the initial segmentation, it becomes beneficial to adjust the segmentation output based on the classification despite a few incorrect decisions.

Surface and crack type influence. Next, we tested methods separately for each individual subset images contained in the combined dataset. Since images from the same dataset source contain fairly similar types of surfaces and cracks, this provides more detail on the performance under different conditions, while also enabling comparison to others in the literature that evaluate only on individual datasets. We trained model on all training images from the whole combined dataset and then

Dataset	*SCCDNet-D32 [15]*		*DeepCrack [43]*		*CrackFormer [60]*		*SegDecNet++*	
	Dice [%]	*IoU [%]*	*Dice [%]*	*IoU [%]*	*Dice [%]*	*IoU [%]*	*Dice [%]*	*IoU [%]*
CFD	65.59	49.47	67.84	51.94	71.52	56.16	**77.80 ± 0.62**	**64.14 ± 1.0**
CrackTree200	3.88	2.04	5.10	2.67	**35.46**	**21.65**	33.02 ± 1.86	20.12 ± 1.26
DeepCrack	78.08	65.47	76.57	63.61	79.61	67.89	**81.17 ± 0.32**	**69.78 ± 0.39**
GAPs384	52.86	37.81	48.94	34.88	**55.18**	**40.03**	54.36 ± 0.53	39.05 ± 0.46
Rissbilder	71.78	56.86	70.98	56.13	72.22	58.06	**80.40 ± 0.43**	**67.95 ± 0.56**
Non-crack	95.28	95.28	89.15	89.15	98.11	98.11	**99.53 ± 0.42**	**99.53 ± 0.42**

Table 7: Results for individual subsets of images from the evaluation dataset.

Threshold ($\hat{\tau}$)	*0.10*	*0.20*	*0.30*	*0.40*	*0.50*	*0.60*	*0.70*	*0.80*	*0.90*	*0.95*	*0.99*	$\hat{\tau} = 0.9 \cdot \tau$
IoU	57.8	59.6	60.9	62.0	63.1	64.2	65.3	66.9	68.7	**70.6**	69.4	70.95
Dice	70.0	71.7	72.9	73.9	74.8	75.7	76.7	77.9	79.6	**80.7**	79.3	80.96

Table 8: Intersection over Union (IoU) and Dice score for different thresholds $\hat{\tau}$.

applied it to test images for each dataset subset individually. We skipped evaluation on CRACK500 images since they were all used for training.

Results are reported in Table 7. SegDecNet++ outperforms all related methods in most cases except on CrackTree200 and GAPs384 subsets, where CrackFormer outperforms SegDecNet++ by 1 - 2 percentage points. Based on image statistics of individual subsets in Table 1, it seams SegDecNet++ does not provide any additional benefits for segmentation of thin cracks, since CrackTree200 contains exclusively thin cracks with ≤ 3 pixels of thickness. On the other hand, a performance gap on GAPs384 cannot be explained with crack thickness since this subset contains similar distribution of crack thickness as others where SegDecNet++ outperforms all other methods. However, GAPs384 contains mostly images of dark asphalt with extremely soft cracks and poor contrast ratio between cracked pixels and background pixels. Main reason for the performance gap can be found in dark asphalt with extremely soft cracks, while poor contrast also appears in other subsets, for instance Rissbilder and CFD datasets have even poorer contrast, where performance is not affected. In fact, on Risbilder subset our SegDecNet++ significantly outperforms all other methods despite poor contrast ratio between cracks and background. Overall, SegDecNet++ outperforms CrackFormer except in thin cracks and dark asphalt with soft cracks were it performs comparable, while completely outperforming SCCDNet-D32 and DeepCrack regardless of surface type, crack thickness, brightness or contrast ratio between crack and background pixels. SegDecNet++ also has the lowest number of false positives on non-crack images compared to the other methods, achieving IoU and Dice of 99.53% versus 98.11%, 95.28% and 89.15% for CrackFormer, SCCDNet-D32 and DeepCrack, respectively.

Threshold influence. We additionally evaluated the effect of threshold $\hat{\tau}$ on the performance of SegDecNet++. For this ablation study, we replaced final adjusted threshold $\hat{\tau}$ with a fixed value. This removes ODS threshold as well as any threshold adjustments from Eq. 1 when classification by neurons and classification by segmentation are in a disagreement.

Results are reported in Table 8. We repeat experiments five times and report averaged values. Results that were obtained with the adjusted thresholds in ODS as per Eq. 1 are also reported in the last column for reference. Among fixed thresholds, the best results are obtained with higher thresholds (0.9 - 0.99), while lowering them gradually decreases performance by up to 10 - 15 percentage points in the end at $\hat{\tau} = 0.1$. Since lower thresholds results in worse performance it indicates that this method returns overconfident scores, however, calibrating them to thresholds $\hat{\tau} > 0.9$ is sufficient for obtaining best performance. Even $\hat{\tau} = 0.99$ results in close to the most optimal performance.

Computational cost. Finally, we measured the computational cost of the inference step for all evaluated methods. We measured inference time on a PC with Intel Xeon E5-1650 and NVIDIA Geforce RTX 2080 Ti. All methods were run on images of 448×448 pixels in size and we report values averaged over 1000 images. We measure only the forward pass without any input loading and prepossessing. Results are reported in Figure 8, where we also report segmentation accuracy as the Dice score in the second axis to better account for the trade-off between computational cost and segmentation accuracy. Our proposed model has proven to have low computational cost with a processing time of 27.5 ms, which is three times lower than the cost for CrackFormer with 78.3 ms, the second-best method on segmentation accuracy. Other related methods achieved 23.3 ms and 60.8 ms for the SCCDNet-D32 and DeepCrack, respectively. While SCCDNet-D32 had a comparable computational cost to our model, this was at the expense of lower segmentation accuracy. On the other hand, our proposed model achieves much better segmentation accuracy while at the same time retaining low computational cost, therefore representing the most optimal approach even for applications that require real-time processing on medium sized images (at least 30 FPS for up to 512×512 images). We also depict the number of trainable parameters as the marker size in Figure 8. Note that CrackFormer

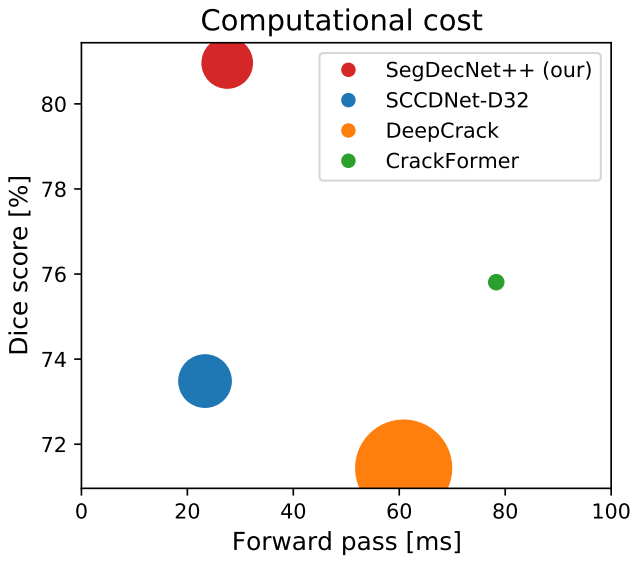

Figure 8: Comparing segmentation performance in Dice score and forward pass in milliseconds conducted on 448×448 images. The size of the marker correlates to the number of training parameters (16 mio for our SegDecNet++ and SCCDNet-D32, 30 mio for DeepCrack and 4 mio for CrackFormer).

has the lowest number of parameters, however, this does not result in low computational cost due to $O(n^2)$ with respect to the input size for self-attention.

5. Conclusion

In this paper, we proposed SegDecNet++, a novel method for per-pixel segmentation of pavement and concrete cracks using a joint training of per-pixel segmentation as well as per-image classification. Our proposed architecture follows the existing U-Net-style encoder-decoder approaches for crack segmentation that have proven successful but also includes a classification module built on top of the encoder features to capture wider-context information. We proposed to use this classification output during the inference to adjust the segmentation output and further improve segmentation accuracy.

We have demonstrated the effectiveness of our proposed method and have shown that the proposed method outperforms several state-of-the-art crack segmentation models on a public dataset for pavement and concrete crack segmentation [15]. Our ablation study also demonstrated that each individual architectural choice contributed positively to the final segmentation accuracy. Furthermore, we have also performed an analysis on a per-image classification-by-segmentation, where we have demonstrated that adjusting the segmentation output in accordance with the classification decision benefits the accuracy of segmentation. This has proven particularly useful for reducing the false positive rate. This resulted in a model that can achieve state-of-the-art segmentation accuracy while also minimizing the number of missed defects from a per-image count, which is an important requirement for quality control in construction processes. Our performance analysis on individual subset of images showed that thin cracks and dark asphalt with soft cracks are the most challenging for SegDecNet++. However, proposed method still resulted in performance comparable to CrackFromer and outperforming others, while fully outperforming all methods including CrackFormer on other types of surfaces and cracks. Finally, we also demonstrated that the proposed model has the best trade-off between segmentation accuracy and computational cost as it achieves the best segmentation accuracy at a low computational cost. Our proposed model has proven to achieve over 30 FPS, whereas CrackFormer, the second-best model based on segmentation metric, is three times slower. This makes our proposed model suitable for real-time applications on medium sized images.

Further utilization of the classification output may also be possible in future work. Since currently the classification output is only considered during the inference time, it may be possible to further improve learning by also considering classification output already during the learning process. Additionally, a more intelligent procedure for setting the threshold adjustment factor may be considered. Instead of using the currently fixed adjustment factor, an image-specific factor, based on the maximal probability value could be used. Finally, we plan on applying the proposed method to other problem domains as well, such as surface defect detection in visual inspection for quality control in production processes, since it has shown a great potential for robustification of the surface defect segmentation step.

Acknowledgments

This work was in part supported by the ARRS research projects J2-3169 (MV4.0) and J2-4457 (RTFM) as well as by research programme P2-0214.

References

[1] Philipp Hüthwohl and Ioannis Brilakis. Detecting healthy concrete surfaces. *Advanced Engineering Informatics*, 37(May):150–162, 2018.

[2] Christian Koch, Kristina Georgieva, Varun Kasireddy, Burcu Akinci, and Paul Fieguth. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Advanced Engineering Informatics*, 29(2):196–210, 2015.

[3] Benoit Hilloulin, Imane Bekrine, Emmanuel Schmitt, and Ahmed Loukili. Modular deep learning segmentation algorithm for concrete microscopic images. *Construction and Building Materials*, 349(June):128736, 2022.

[4] Jueqiang Tao, Haitao Gong, Feng Wang, Xiaohua Luo, Xin Qiu, and Jinli Liu. Deep learning based automated segmentation of air-void system in hardened concrete surface using three dimensional reconstructed images. *Construction and Building Materials*, 324(January):126717, 2022.

[5] Yijia Dong, Chao Su, Pizhong Qiao, and Lizhi Sun. Microstructural crack segmentation of three-dimensional concrete images based on deep convolutional neural networks. *Construction and Building Materials*, 253:119185, 2020.

[6] Zhiming Dong, Jiajun Wang, Bo Cui, Dong Wang, and Xiaoling Wang. Patch-based weakly supervised semantic segmentation network for crack detection. *Construction and Building Materials*, 258:120291, 2020.

[7] Pai Pan, Yaming Xu, Cheng Xing, and Yang Chen. Crack detection for nuclear containments based on multi-feature fused semantic segmentation. *Construction and Building Materials*, 329(January):127137, 2022.

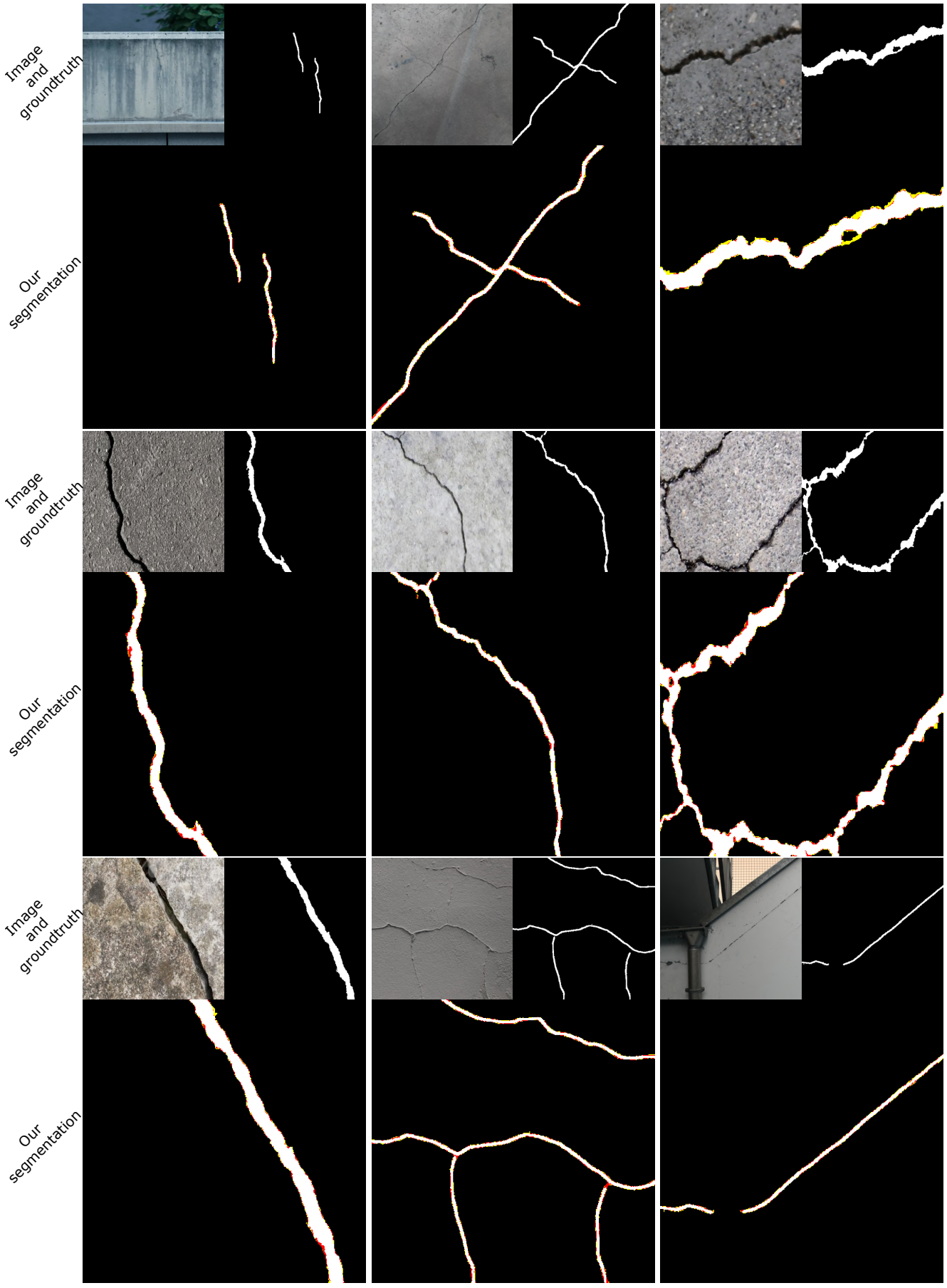

Figure 9: Examples of crack segmentation with our proposed method. We depict false positive pixels in red, and false negatives in yellow, while the correct background segmentation is in black and the correct foreground in white.

[8] Lei Zhang, Fan Yang, Yimin Daniel Zhang, and Ying Julie Zhu. Road crack detection using deep convolutional neural network. In *Proceedings - International Conference on Image Processing, ICIP*, volume 2016-Augus, pages 3708–3712, 2016.

[9] Yue Fei, Kelvin C.P. Wang, Allen Zhang, Cheng Chen, Joshua Q. Li, Yang Liu, Guangwei Yang, and Baoxian Li. Pixel-Level Cracking Detection on 3D Asphalt Pavement Images through Deep-Learning- Based CrackNet-V. *IEEE Transactions on Intelligent Transportation Systems*, 21(1):273–284, 2020.

[10] Yong Shi, Limeng Cui, Zhiquan Qi, Fan Meng, and Zhensong Chen. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3434–3445, 2016.

[11] Fan Yang, Lei Zhang, Sijia Yu, Danil Prokhorov, Xue Mei, and Haibin Ling. Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection, 2019.

[12] Hui Lin, Bin Li, Xinggang Wang, Yufeng Shu, and Shuanglong Niu. Automated defect inspection of LED chip using deep convolutional neural network. *Journal of Intelligent Manufacturing*, pages 1–10, 2018.

[13] Domen Tabernik, Samo Šela, Jure Skvarč, and Danijel Skočaj. Segmentation-Based Deep-Learning Approach for Surface-Defect Detection. *Journal of Intelligent Manufacturing*, 2020.

[14] Jakob Božič, Domen Tabernik, and Danijel Skočaj. Mixed supervision for surface-defect detection: from weakly to fully supervised learning. *Computers in Industry*, 129, 2021.

[15] Haotian Li, Zhuang Yue, Jingyu Liu, Yi Wang, Huaiyu Cai, Kerang Cui, and Xiaodong Chen. Sccdnet: A pixel-level crack segmentation network. *Applied Sciences*, 11(11), 6 2021.

[16] Isaac Osei Agyemang, Xiaoling Zhang, Daniel Acheampong, Isaac Adjei-Mensah, Goodlet Akwasi Kusi, Bernard Cobbinah Mawuli, and Bless Lord Y. Agbley. Autonomous health assessment of civil infrastructure using deep learning and smart devices. *Automation in Construction*, 141(February):104396, 2022.

[17] Jau Yu Chou and Chia Ming Chang. Low-story damage detection of buildings using deep neural network from frequency phase angle differences within a low-frequency band. *Journal of Building Engineering*, 55(May):104692, 2022.

[18] Yuqing Gao and Khalid M. Mosalam. Deep learning visual interpretation of structural damage images. *Journal of Building Engineering*, 60(February), 2022.

[19] Sattar Dorafshan and Hoda Azari. Deep learning models for bridge deck evaluation using impact echo. *Construction and Building Materials*, 263:120109, 2020.

[20] Ruoxian Li, Jiayong Yu, Feng Li, Ruitao Yang, Yudong Wang, and Zhihao Peng. Automatic bridge crack detection using Unmanned aerial vehicle and Faster R-CNN. *Construction and Building Materials*, 362(November 2022):129659, 2023.

[21] Thai Son Tran, Son Dong Nguyen, Hyun Jong Lee, and Van Phuc Tran. Advanced crack detection and segmentation on bridge decks using deep learning. *Construction and Building Materials*, 400(August):132839, 2023.

[22] Shanaka Kristombu Baduge, Sadeep Thilakarathna, Jude Shalitha Perera, Gihan P. Ruwanpathirana, Lachlan Doyle, Mitchell Duckett, Joel Lee, Jiratigan Saenda, and Priyan Mendis. Assessment of crack severity of asphalt pavements using deep learning algorithms and geospatial system. *Construction and Building Materials*, 401(May):132684, 2023.

[23] L. Minh Dang, Hanxiang Wang, Yanfen Li, Le Quan Nguyen, Tan N. Nguyen, Hyoung Kyu Song, and Hyeonjoon Moon. Deep learning-based masonry crack segmentation and real-life crack length measurement. *Construction and Building Materials*, 359(October):129438, 2022.

[24] Deegan J Atha and Mohammad R Jahanshahi. Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection. *Structural Health Monitoring*, 17(5):1110–1128, 2018.

[25] Ali Kazemian, Xiao Yuan, Omid Davtalab, and Behrokh Khoshnevis. Computer vision for real-time extrusion quality monitoring and control in robotic construction. *Automation in Construction*, 101(August 2018):92–98, 2019.

[26] L. Minh Dang, Hanxiang Wang, Yanfen Li, Le Quan Nguyen, Tan N. Nguyen, Hyoung Kyu Song, and Hyeonjoon Moon. Lightweight pixel-level semantic segmentation and analysis for sewer defects using deep learning. *Construction and Building Materials*, 371(November 2022):130792, 2023.

[27] Niannian Wang, Jiaxiu Dong, Hongyuan Fang, Bin Li, Kejie Zhai, Duo Ma, Yibo Shen, and Haobang Hu. 3D reconstruction and segmentation system for pavement potholes based on improved structure-from-motion (SFM) and deep learning. *Construction and Building Materials*, 398(July):132499, 2023.

[28] Rui Duan, Hui Deng, Mao Tian, Yichuan Deng, and Jiarui Lin. SODA: A large-scale open site object detection dataset for deep learning in construction. *Automation in Construction*, 142(February):104499, 2022.

[29] Nipun D. Nath, Amir H. Behzadan, and Stephanie G. Paal. Deep learning for site safety: Real-time detection of personal protective equipment. *Automation in Construction*, 112(January):103085, 2020.

[30] Haoyuan Guo, Xi Yang, Nannan Wang, and Xinbo Gao. A CenterNet++ model for ship detection in SAR images. *Pattern Recognition*, 112:107787, 2021.

[31] Ziyue Zeng, Yongqi Wei, Zhenhua Wei, Wu Yao, Changying Wang, Bin Huang, Mingzi Gong, and Jiansen Yang. Deep learning enabled particle analysis for quality assurance of construction materials. *Automation in Construction*, 140(January):104374, 2022.

[32] Hua Zhang, Rui Zhang, Daquan Sun, Fan Yu, Zhang Gao, Shuifa Sun, and Zichang Zheng. Analyzing the pore structure of pervious concrete based on the deep learning framework of Mask R-CNN. *Construction and Building Materials*, 318(July 2021):125987, 2022.

[33] Yu Song, Zilong Huang, Chuanyue Shen, Humphrey Shi, and David A. Lange. Deep learning-based automated image segmentation for concrete petrographic analysis. *Cement and Concrete Research*, 135(February):106118, 2020.

[34] Renata Lorenzoni, Iurie Curosu, Sidnei Paciornik, Viktor Mechtcherine, Martin Oppermann, and Flavio Silva. Semantic segmentation of the micro-structure of strain-hardening cement-based composites (SHCC) by applying deep learning on micro-computed tomography scans. *Cement and Concrete Composites*, 108(December 2019):103551, 2020.

[35] Ke Xu, Qingxu Jin, Jiaqi Li, Daniela M. Ushizima, Victor C. Li, Kimberly E. Kurtis, and Paulo J.M. Monteiro. In-situ microtomography image segmentation for characterizing strain-hardening cementitious composites under tension using machine learning. *Cement and Concrete Research*, 169(January), 2023.

[36] Omid Davtalab, Ali Kazemian, Xiao Yuan, and Behrokh Khoshnevis. Automated inspection in robotic additive manufacturing using deep learning for layer deformation detection. *Journal of Intelligent Manufacturing*, 33(3):771–784, 2022.

[37] Zoe Mayer, James Kahn, Yu Hou, Markus Götz, Rebekka Volk, and Frank Schultmann. Deep learning approaches to building rooftop thermal bridge detection from aerial images. *Automation in Construction*, 146(December 2022):104690, 2023.

[38] Lufeng Wang, Jiepeng Liu, Yan Zeng, Guozhong Cheng, Huifeng Hu, Jiahao Hu, and Xuesi Huang. Automated building layout generation using deep learning and graph algorithms. *Automation in Construction*, 154(July):105036, 2023.

[39] Tung Ching Su and Ming Der Yang. Application of morphological segmentation to leaking defect detection in sewer pipelines. *Sensors*, 14(5):8686–8704, 2014.

[40] Jinshan Tang and Yanliang Gu. Automatic Crack Detection and Segmentation Using a Hybrid Algorithm for Road Distress Analysis. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 3026–3030, 2013.

[41] Mark David Jenkins, Thomas Arthur Carr, Maria Insa Iglesias, Tom Buggy, and Gordon Morison. A deep convolutional neural network for semantic pixel-wise segmentation of road and pavement surface cracks. In *European Signal Processing Conference*, volume 2018-Septe, pages 2120–2124, 2018.

[42] Cao Vu Dung and Le Duc Anh. Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction*, 99(July 2018):52–58, 2019.

[43] Yahui Liu, Jian Yao, Xiaohu Lu, Renping Xie, and Li Li. DeepCrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing*, 338:139–153, 2019.

[44] Fu Tao Ni, Jian Zhang, and Zhi Qiang Chen. Pixel-level crack delineation in images with convolutional feature fusion. *Structural Control and Health Monitoring*, 26(1), 2019.

[45] Stephen L.H. Lau, Edwin K.P. Chong, Xu Yang, and Xin Wang. Auto-

mated Pavement Crack Segmentation Using U-Net-Based Convolutional Neural Network. *IEEE Access*, 8:114892–114899, 2020.

[46] Xinyu He, Zhiwen Tang, Yubao Deng, Guoxiong Zhou, Yanfeng Wang, and Liujun Li. UAV-based road crack object-detection algorithm. *Automation in Construction*, 154(February):105014, 2023.

[47] Dimitrios Loverdos and Vasilis Sarhosis. Automatic image-based brick segmentation and crack detection of masonry walls using machine learning. *Automation in Construction*, 140(June):104389, 2022.

[48] Yuchuan Du, Shan Zhong, Hongyuan Fang, Niannian Wang, Chenglong Liu, Difei Wu, Yan Sun, and Mang Xiang. Modeling automatic pavement crack object detection and pixel-level segmentation. *Automation in Construction*, 150(March):104840, 2023.

[49] Qiwen Qiu and Denvid Lau. Real-time detection of cracks in tiled sidewalks using YOLO-based method applied to unmanned aerial vehicle (UAV) images. *Automation in Construction*, 147(January):104745, 2023.

[50] Jing Zhang, Yuan Yuan Cai, Dong Yang, Ye Yuan, Wen Yu He, and Yan Jia Wang. MobileNetV3-BLS: A broad learning approach for automatic concrete surface crack detection. *Construction and Building Materials*, 392(June):131941, 2023.

[51] Jiaxiu Dong, Niannian Wang, Hongyuan Fang, Qunfang Hu, Chao Zhang, Baosong Ma, Duo Ma, and Haobang Hu. Innovative method for pavement multiple damages segmentation and measurement by the Road-Seg-CapsNet of feature fusion. *Construction and Building Materials*, 324(January):126719, 2022.

[52] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic Routing Between Capsules. In *Neural Information Processing Systems*, 2017.

[53] Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4217–4228, 2021.

[54] Qin Zou, Zheng Zhang, Qingquan Li, Xianbiao Qi, Qian Wang, and Song Wang. DeepCrack: Learning hierarchical convolutional features for crack detection. *IEEE Transactions on Image Processing*, 28(3):1498–1512, 2019.

[55] Chengjia Han, Tao Ma, Ju Huyan, Xiaoming Huang, and Yanning Zhang. CrackW-Net: A Novel Pavement Crack Image Segmentation Convolutional Neural Network. *IEEE Transactions on Intelligent Transportation Systems*, PP:1–10, 2021.

[56] Erkai Li and Huiming Tang. A Novel Convolutional Neural Network for Pavement Crack Segmentation. In *Proceedings - 20th IEEE International Conference on Machine Learning and Applications, ICMLA 2021*, pages 95–99, 2021.

[57] Haotian Li, Hongyan Xu, Xiaodong Tian, Yi Wang, Huaiyu Cai, Kerang Cui, and Xiaodong Chen. Bridge crack detection based on SSENets. *Applied Sciences*, 10(12), 2020.

[58] Chao Xiang, Wei Wang, Lu Deng, Peng Shi, and Xuan Kong. Crack detection algorithm for concrete structures based on super-resolution reconstruction and segmentation network. *Automation in Construction*, 140(May):104346, 2022.

[59] Zhong Zhou, Longbin Yan, Junjie Zhang, Yidi Zheng, Chenjie Gong, Hao Yang, and E. Deng. Automatic segmentation of tunnel lining defects based on multiscale attention and context information enhancement. *Construction and Building Materials*, 387(February):131621, 2023.

[60] Huajun Liu, Xiangyu Miao, Christoph Mertz, Chengzhong Xu, and Hui Kong. CrackFormer: Transformer Network for Fine-Grained Crack Detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3763–3772, 2021.

[61] Feng Guo, Yu Qian, Jian Liu, and Huayang Yu. Pavement crack detection based on transformer network. *Automation in Construction*, 145(November 2022):104646, 2023.

[62] Elyas Asadi Shamsabadi, Chang Xu, Aravinda S. Rao, Tuan Nguyen, Tuan Ngo, and Daniel Dias-da Costa. Vision transformer-based autonomous crack detection on asphalt and concrete surfaces. *Automation in Construction*, 140(December 2021):104316, 2022.

[63] Domen Rački, Dejan Tomaževič, and Danijel Skočaj. A compact convolutional neural network for textured surface anomaly detection. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1331–1339, 2018.

[64] Jakob Božič, Domen Tabernik, and Danijel Skočaj. End-to-end training of a two-stage neural network for defect detection. In *International Conference on Pattern Recognition*, 2020.

[65] Qin Zou, Yu Cao, Qingquan Li, Qingzhou Mao, and Song Wang. CrackTree: Automatic crack detection from pavement images. *Pattern Recognition Letters*, 33(3):227–238, 2012.

[66] Markus Eisenbach, Ronny Stricker, Daniel Seichter, Karl Amende, Klaus Debes, Maximilian Sesselmann, Dirk Ebersbach, Ulrike Stoeckert, and Horst Michael Gross. How to get pavement distress detection ready for deep learning? A systematic approach. *Proceedings of the International Joint Conference on Neural Networks*, 2017-May:2039–2047, 2017.

[67] Myeongsuk Pak and Sanghoon Kim. Crack Detection Using Fully Convolutional Network in Wall-Climbing Robot. In James J Park, Simon James Fong, Yi Pan, and Yunsick Sung, editors, *Advances in Computer Science and Ubiquitous Computing*, pages 267–272, Singapore, 2021. Springer Singapore.

[68] Sattar Dorafshan, Robert J. Thomas, and Marc Maguire. SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks. *Data in Brief*, 21:1664–1668, 2018.