Designing a Hybrid Neural System to Learn Real-world Crack Segmentation from Fractal-based Simulation

Achref Jaziri
Goethe University
Frankfurt am Main, Germany
Jaziri@em.uni-frankfurt.de

Martin Mundt
TU Darmstadt and hessian.AI
Darmstadt, Germany
martin.mundt@tu-darmstadt.de

Andres Fernandez Rodriguez
University of Tübingen
Tübingen, Germany
a.fernandez@uni-tuebingen.de

Visvanathan Ramesh
Goethe University and hessian.AI
Frankfurt am Main, Germany
vramesh@em.uni-frankfurt.de

Abstract

Identification of cracks is essential to assess the structural integrity of concrete infrastructure. However, robust crack segmentation remains a challenging task for computer vision systems due to the diverse appearance of concrete surfaces, variable lighting and weather conditions, and the overlapping of different defects. In particular recent data-driven methods struggle with the limited availability of data, the fine-grained and time-consuming nature of crack annotation, and face subsequent difficulty in generalizing to out-of-distribution samples. In this work, we move past these challenges in a two-fold way. We introduce a high-fidelity crack graphics simulator based on fractals and a corresponding fully-annotated crack dataset. We then complement the latter with a system that learns generalizable representations from simulation, by leveraging both a pointwise mutual information estimate along with adaptive instance normalization as inductive biases. Finally, we empirically highlight how different design choices are symbiotic in bridging the simulation to real gap, and ultimately demonstrate that our introduced system can effectively handle real-world crack segmentation.

1. Introduction

The process of structural monitoring and assessment of civil infrastructure is an important task to ensure safety and usability. Executed primarily by humans, the inspection process is time consuming and labor-intensive, as it needs to be carried out at the target location, is potentially dangerous and can lead to down-times in the infrastructure use. To alleviate these challenges, the deployment of robots with integrated computer vision systems is emerging as an exciting, safe and low cost addition to traditional inspection methods[29, 51].

In general, such a computer vision system should be robust and invariant to a variety of nuisance variables such as illumination, object scale or pose. Early works achieved these desiderata by stacking and combining quasi-invariant transformations specified by a domain expert to guarantee that the output remains unchanged for a range of transformations that are irrelevant to the application domain [7, 12]. In contrast, modern data-driven systems may learn these transformations by relying on large amounts of labeled data. In recent years, deep learning techniques in conjunction with labelled datasets were introduced for structural inspection tasks like crack identification [30, 14, 26].

However, gathering appropriate real-world data for training is tremendously challenging. Data acquisition is particularly tough in the case of cracks on concrete bridges, where defects tend to be located in difficult to capture areas and overlap with other defects like spalling, exposed metal bars etc. Moreover, data labeling is not only excessively time consuming, it is prone to errors due to the fine-grained nature of cracks and requires highly specialized experts (who may not end up agreeing) to provide precise ground-truth [3]. Previous works have thus proposed datasets addressing the crack identification challenge from a multi-target classification perspective [37], but similar real-world efforts are still needed for semantic segmentation in diverse contexts.

Faced with a lack of appropriately annotated and diverse

data, one may resort to physics based rendering, which has enabled the creation of photorealistic synthetic data for training and testing computer vision models [35, 17]. Compared to data-driven based generative models, the approach promises full control over the scenes and automatically generated ground truth maps. Alas, there is a typical statistical mismatch between simulated and real images due to modeling assumptions and computational approximations. Therefore, purely data-driven neural networks trained with synthetic images tend to suffer from performance degradation when applied to real images.

In this work, we seek to overcome existing limitations by combining the strengths of context specific modeling and data-driven designs (hybrid system), enabling us to fully leverage physics based rendering as an underlying source of data. For this purpose, we first introduce a fractal-based concrete crack simulation pipeline and investigate the use of synthetic images for learning in the context of crack segmentation. We also propose a model that can achieve better performance by leveraging image-based pointwise mutual information as well as style transfer techniques for better generalization. To empirically examine the latter, we annotate a subset of a prominent real-world concrete defect classification dataset [37] and thoroughly experimentally corroborate our proposed design choices in additional settings. In summary, our contributions are as follows:

- We propose the "Cracktal" high-fidelity simulator for cracks on concrete surfaces to generate pixel wise annotated data with depth and surface normal maps [1].
- We present an approach to close the gap between performance on simulated and real data through **C**onsistency enforced training between **A**daptive instance normalization and **P**ointwise mutual information, CAP-Net for short.
- We annotate real-world images of concrete bridges with cracks from the popular CODEBRIM dataset [37] to empirically corroborate our approach.
- We investigate the performance of different algorithms in the context of crack detection and empirically validate our approach on the annotated CODEBRIM images as well as other public crack segmentation benchmarks.

2. Related Work

We summarize related work for crack detection, the use of synthetic data for training neural networks and approaches to reduce the Sim2Real gap.

[1]Code as well as datasets are availabe at: https://github.com/achrefjaziri/SimCrack.

2.1. Crack Identification

Traditional works on crack recognition focus on using image processing algorithms like edge and boundary detection techniques [1], morphological operation based methods [57], principle component analysis [2], or automatic clustering for segmentation based on Canny and K-Means [31]. The work of Koch et. al [29] presents an exhaustive literature review on the common practices of assessing the state of concrete infrastructure and crack detection.

Recent works leverage data-driven approaches for crack identification using classification or semantic segmentation neural architectures [11, 14, 16]. The works of Cao et. al [8] and König et. al [30] provide a review of current data-driven crack detection approaches. However, one of the main limitations of current approaches is that the training data are mostly composed of simple and small datasets with uniform asphalt or concrete backgrounds [61, 20, 45, 5, 63], which hinders effective generalization of data-driven approaches in the context of precise semantic crack segmentation.

More recent works considered the generation of synthetic scenes for civil engineering tasks, including crack detection on pavement surfaces [58, 56, 60, 41]. These works collectively illustrate the importance of synthetic data generation with pixel-accurate labels in overcoming limitations posed by the availability of large annotated datasets.

In our work, we overcome the data hurdles by proposing a high-fidelity data simulator, which we can fully leverage by proposing a model that incorporates necessary inductive biases while enabling effective learning.

2.2. Simulating Data and the Sim2Real Gap

In recent years, data-driven generative models have gained considerable popularity. Despite that, simulators based on physics-based rendering engines have maintained their significance. This is largely attributed to their ability to effortlessly produce pixel-accurate labels, thus reducing the burden of manual annotation. Furthermore, these simulators offer a unique advantage in generating data with controlled priors, enabling the generation of diverse datasets tailored to specific scenarios and applications. Proposed simulators in the literature include GTA5 [40], SYNTHIA [43] and endless runner for continual learning [21].

Whereas some works show promising results for the use of synthetic data in detection tasks [47, 36, 54], models trained with synthetic data are well-known to face difficulties in generalizing to real data, due to the statistical gap between synthetic images and real images [49, 48, 55]. Apart from improving the graphics rendering pipeline itself, this Sim2Real gap is typically reduced by seeking out domain adaptation (DA) or domain generalization (DG) techniques.

DA approaches focus on adapting the statistics of the synthetic data to that of the target domain, for instance by adversarially tuning the parameters of the generative mod-

els based on the statistics of the real data for better generalization [50, 4]. Others [28] make use of style transfer methods to adapt the training data. We refer to [62, 53] for comprehensive surveys.

In contrast, DG approaches seek to improve the robustness of DNNs to arbitrary unseen domains, see Wang et. al [52] for a detailed review. Approaches for learning domain-agnostic feature representations can leverage meta-learning [6, 19], adversarial training [33], instance normalization [39], selective whitening [13], style transfer [59].

Other works bias their models to focus on image features that are more realistic and transferable to improve generalization in a given application domain. For instance, features related to the image geometry can be more generalizable in the case of car detection [44]. Since geometry and semantics are naturally connected, [10, 25] propose to mitigate the limitations of synthetic data by leveraging the geometric information in a multi-task learning framework. In our work, we follow the spirit of these works, but note that most application contexts of synthetic data in the literature focus on objects with well defined shapes (e.g cars, buildings etc.). In contrast, we emphasize that cracks, like most defects, have highly irregular shapes. We design an extendable physics-based crack simulator and subsequently leverage specific crack sensitive quasi-invariant models to learn more generalizable representations to reduce the Sim2Real gap.

3. Cracktal: A Fractal-based Simulator for Cracked Concrete Surfaces

Cracktal is a physics-based simulator that generates images of cracked concrete surfaces along with their semantic ground truth, depth and surface normal maps. The overall rendering workflow consists of two main steps: scene and crack generation. A set of texture maps are used to set the scene based on physics based rendering rules in Blender Engine. A random crack is then generated using our fractal generator model, detailed below, and added to the scene's material. The full scene is then rendered using Blender Cycles PBR engine and corresponding ground truth maps are generated. Our simulator code is written using Blender's Python API with further details in the Appendix. Figure 1 illustrates examples of the synthetic images generated with a 2048×2048 resolution.

3.1. Physics-based Scene Generation

In the scene generation process, non relevant backgrounds (e.g sky, out of focus buildings etc.) are excluded, assuming an up-close camera. The base components of physics based rendering (PBR) workflows: albedo, normal, roughness, and height maps are applied to a plane mesh grid to generate a realistic looking concrete surface, defining its color, surface and subsurface scattering, and geometrical displacement respectively. The required PBR metallicity map is included but remains uniformly zero, as concrete is a dielectric. An optional ambient occlusion map can be included to introduce surface markings, e.g. graffiti. The textures used in this work were created from real concrete images from the CODEBRIM training dataset [37] and decomposed manually by the Substance B2M software.

The environment is illuminated utilizing a simulated natural sunlight source. The black body radiator possesses two key attributes: luminous intensity (L_I) and color temperature (L_T). Luminous intensity determines the amount of energy that the light source emits into the scene, whereas color temperature defines the chromaticity of the illuminant. In the datasets of our later study, a color temperature of $L_T = 5800$ Kelvin and intensity $L_I = 3.3$ were chosen. The rotation of the light source is parameterized by its Euler α, β γ angles as in common conventions. α and γ are fixed angles with values $\frac{\pi}{3}$ and 0. By varying the β angle, we simulate the change of the hour of day during which the image is captured. The β angle is randomly sampled from:

$$\beta \sim \mathscr{U}(\frac{-\pi}{6}, \frac{\pi}{6}) \tag{1}$$

3.2. Fractal-based Crack Generation

Cracks are highly irregular, but like many other patterns found in nature can be represented as fractals. In order to generate a crack pattern, we draw inspiration from a decades old model presented by [32] as a baseline. The authors suggest the use of a stochastic version of the Koch "snowflake" fractal in the generation of pavement distress features, e.g cracks on a road surface. A conventional Koch "snowflake" fractal can be generated through the iterative splitting of each straight line into three equal length segments. The middle segment is then replaced by two segments of equal length to form an equilateral triangle. These steps are repeated for each straight line to create a regular fractal until a desired subdivision depth is reached.

By modifying the displacement parameters at each step, it is feasible to generate non-uniform fractals that resemble cracks. Rather than dividing each line into identical segments to form an equilateral triangle, the position of the third point, which is determined by both the magnitude r and angle θ, is altered in each step. In our simulation, the angle is sampled from a Gaussian normal distribution with a mean of $\mu = 0$ and a standard deviation of $\sigma = 30$ degrees. The probability density of displacement magnitude r is given by $P(r) = \frac{2r}{p^2}$ where p is a hyper-parameter. For intuition, Figure 2 illustrates these steps for a "snowflake" and the stochastic version for crack generation.

Finally, before adding the crack to the rendered scene, the generated crack is randomly translated and rotated, and a Gaussian blur is applied in order to introduce width to the

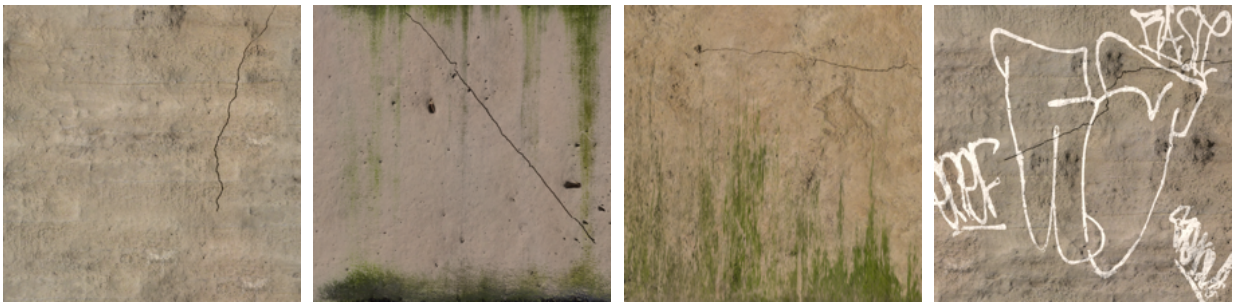

Figure 1. Cracktal examples with texture variety and presence of other perturbations/ anomalies like moss and graffiti. Images were generated at 2048×2048 resolution and are heavily down-sampled for view in pdf at loss of quality.

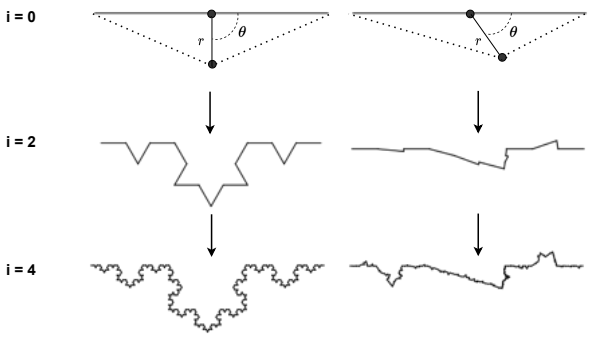

Figure 2. Example Koch fractal (left) and stochastic version for cracks (right). While the magnitude r and the angle θ of the displacement are typically fixed, they are randomized for each displacement iteration (i) when generating the irregular crack shape.

crack.

3.3. Annotation of Real World Data

In conjunction with simulated data, we require real world images to validate systems trained with synthetic images. Ideally, the chosen real world image should offer additional challenges that make generalization less straightforward. For these reasons, we semantically annotated images provided in the CODEBRIM dataset [37] on a pixel basis.

We have chosen this dataset as its development has been motivated by the need for a concrete visual inspection dataset that contains other overlapping defects and features various levels of deterioration, defect severity and surface appearance. Previous works [61, 20, 45, 5, 63], focused on data where cracks are the only visible defect, and they are usually centered in the image, making them unrealistically easy to segment. Most of them also show pavement cracks, which may differ in appearance from concrete cracks. In many CODEBRIM images, other defects like exposed reinforcement bars, spallation, corrosion and calcium leaching are present. In particular the latter share visual similarities with cracks, which makes the prediction more challenging.

We selected image patches containing visible cracks and annotated them using GIMP. In this way, Multiple annotators semantically annotated images of 1500×844 resolution, each containing at least one crack. We consolidated consistent annotations into a set of 420 examples for our real-world test set.

4. CAP-Net: A Hybrid Neural Approach for Crack Segmentation

To fully leverage our simulator and bridge the Sim2Real gap, we introduce **CAP**-Net, a hybrid neural model based on **C**onsistency enforced training between **A**daptive instance normalization and **P**ointwise mutual information. It is composed of two parallel network branches, each one based on a U-Net architecture [42]. During training, we input the RGB image stylized by the AdaIN module. The second network is equipped with a PMI module to extract representations that are projected into a quasi-invariant feature space that helps with the domain transfer. Both networks are connected with a consistency loss to enforce common representations across the different domains. We train our pipeline end to end with the help of the synthetic images and their ground truths generated by Cracktal, as depicted in Figure 3.

4.1. Pointwise Mutual Information

Cracks can be viewed as anomalies in a textured surface. Pointwise mutual information (PMI), computed in a local neighborhood, is a measure of deviation of the gray-level co-occurrence statistics in the neighborhood relative to marginal statistics of gray-levels globally. Thus, the PMI measure flags boundaries between dominating texture patterns in the image. The resulting output is an indicator of texture anomalies and directly relates to hypothesized cracks and other boundary structures.

Drawing inspiration from prior work [24], natural objects produce probability density functions that are well clustered. These clusters can be discovered in an unsupervised manner, and fitted by kernel density estimation (KDE). The obtained density functions can then further be

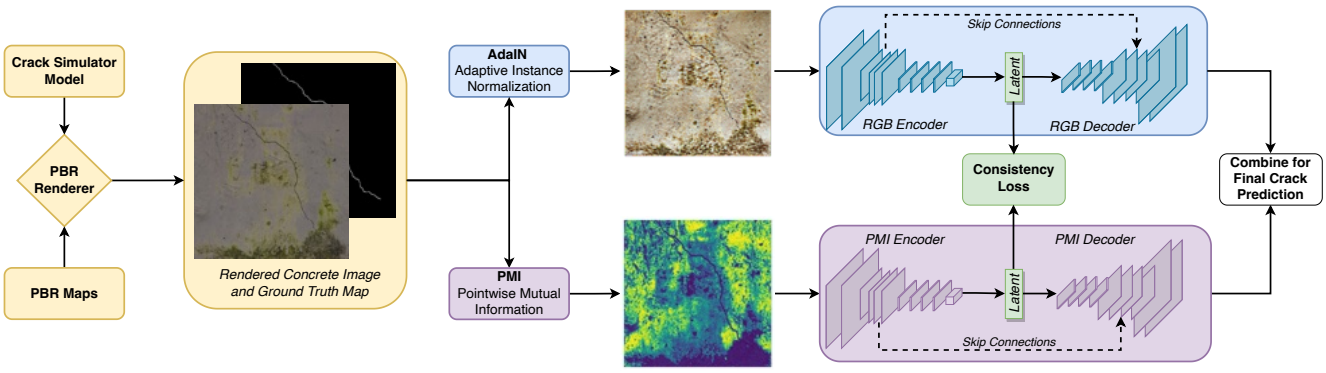

Figure 3. Schematic of the Cracktal to CAP-Net training pipeline. Based on simulated images and their automatic annotation (yellow shading), consistency enforced training (green) is performed between two networks extended with adaptive instance normalization (AdaIN, blue) and pointwise mutual information (PMI, purple) respectively. For final inference, the AdaIN style-transfer module is dropped.

leveraged to distinguish between common pixel pairs (belonging to the background texture) from less common pairs (belonging to anomalies in the texture or edges). To compute the PMI scores between two pixels, we first need to estimate the joint distribution and marginal distributions for image pixels. For the marginal distribution $P(A)$, we sample pixels randomly from the image to perform the KDE. To estimate the joint distribution $P(x_i, x_j)$, we sample pairs of pixels at various distances and perform KDE.

For a pixel pair (x_i, x_j) the PMI score is computed as:

$$PMI(x_i, x_j) = log \frac{P(x_i, x_j)^\tau}{P(x_i) \cdot P(x_j)} \tag{2}$$

The parameter τ boosts the scores of common pairs and addresses the bias of PMI towards low-frequency events (i.e when the marginal distributions are small). When $\tau = 1$, $PMI(x_i, x_j)$ specifically compares the likelihood of observing the pixel x_i near x_j to the overall probability of observing x_i and x_j in the image. The final affinity score for each pixel is computed using the PMI score with the neighboring pixels. We define the set of neighbouring pixels of x_i as N_i. The PMI scores between a pixel and its neighbors are exponentiated and summed to estimate an affinity score for each pixel in the original image, indicating if this pixel belongs to the dominant background texture or is an anomaly:

$$Affinity(x_i) = \sum_{x_j \in N_i} e^{PMI(x_i, x_j)} \tag{3}$$

The scores are then passed to a neural network for crack prediction. Note that the exponential is important to obtain more stable affinity scores, which helps with learning.

4.2. Style Transfer

In addition to the use PMI as an inductive bias, we further reduce the Sim2Real gap from the data-driven angle by performing style transfer operations based on the adaptive instance normalization (AdaIn) [22], which aligns the mean and variance of the content features with those of the style features. The content features are obtained by encoding an image generated by Cracktal simulator using a VGG network pretrained on ImageNet [18]. Similarly, the style features are encoded from a texture image. The AdaIN layer is used to perform style transfer in the feature space and aligning the features of the content and style images. A decoder is learned to invert the AdaIN output to the image spaces.

We sample texture from the describable textures dataset [15] to perform the style transfer on Cracktal images. This way, we can augment the synthetic training data and increase the texture variety while at the same time keeping the semantic content of the original image and more specifically the crack intact. We note that style transfer is only performed during training with probability of 0.5 and is completely dropped during testing.

4.3. Consistency Loss

Finally, to get the best of both worlds, we add a consistency loss between the network trained with RGB images and the network trained with PMI based affinity scores. We postulate that ensuring consistency of the latent space representations across projected subspaces of the outputs of two networks will lead to robust features that will enable better transfer to real data. For training image X_i, The consistency loss is imposed as follows:

$$\mathcal{L}_{CL}(X_i) = (f_1(enc_{rgb}(X_i)) - f_2(enc_{pmi}(X_i)))^2 \tag{4}$$

where enc_{rgb} and enc_{pmi} are the encoding functions for the RGB and PMI networks respectively. The obtained latent encodings are then passed to projection heads (f_1 and f_2) before contrasting them.

5. Experiments

Our empirical investigation follows four key questions:
(Q1) Are Cracktal assumptions plausible? To corroborate the plausibility of our modelling assumptions and the utility of synthetic data for crack detection, we contrast the performance of a U-Net trained with real world data with a U-Net trained with our synthetic data.
(Q2) Do simulated auxiliary tasks improve generalization? In the spirit of prior works [10, 25], we further consider how the addition of auxiliary tasks can improve generalization performance in the context of crack segmentation. More specifically, crack patterns have local geometric variations, e.g. surface normal distribution variation, and depth variations relative to the geometry and depth in the surrounding context. Similarly, PMI maps address estimation of an auxiliary task of appearance anomaly extraction.
(Q3) Does our approach of CAP-Net reduce the Sim2Real gap? We empirically corroborate that our proposed method outperforms existing baselines, even when the latter are trained on real-data, effectively demonstrating how our design choices along with a domain specific simulation can lead to more robust crack segmentation models.
(Q4) Are all design choices for CAP-Net meaningful? We ablate each component of our hybrid CAP-Net to showcase that each proposed element has meaningful impact towards the overall CAP-Net performance.

5.1. Baselines and Additional Evaluation Datasets

In addition to SegCODEBRIM, we evaluate our models on a collection of the following public datasets: CRACK500 [61], GAPs384 [20], CFD [45], AEL [5], Cracktree200 [63]. We merge these into 950 images of cracks captured under various conditions. We refer to the experiments using these datasets collectively as the multi-source set. For consistency, we downsample all images to 256×256. As intuitive baselines, we consider the following models: A U-Net trained with synthetic data (U-Net), A U-Net trained with collection of multi-source data (U-Net(MultiSet)), A U-Net trained with real and synthetic data (U-Net(Sim+Real)). In addition, we compare to the attention based U-Net variant (Attn-U-Net) [38] and to TransU-Net [9]. For analysis of the multi-task training in Q2, we further construct Multi-U-Net architectures, based on a single joint encoder and one separate decoder per modality, in the spirit of prior segmentation works [10, 25].

5.2. Evaluation Metrics

Evaluating binary semantic segmentation maps with common overlap based scores such as Dice or Intersection over Union (IOU) comes with limitations. For cracks, connectivity is important but slight over- or under- segmentation of crack pixels can be tolerated, especially knowing that the ground truth maps are usually annotated by humans using different annotation tools with varied settings. For these reasons, we take inspiration from the medical imaging literature and adapt various metrics [34, 27].

Hausdorff based Metrics [23]: For two point sets X and Y, the one-sided Hausdorff Distance from X to Y is:

$$hd(X,\, Y) = \max_{x \in X} \min_{y \in Y} dist(x, y) \tag{5}$$

where $dist$ is a distance measuring function between pixels x and y. The bidirectional Hausdorff Distance is then:

$$HDF(X,\, Y) = \max(hd(X, Y), hd(Y, X)) \tag{6}$$

We use both the euclidean distance and radial basis function (RBF) as a distance measure between pixels. RBF, also known as the squared exponential kernel, is defined as:

$$RBF(x, y) = exp(-\frac{d(x, y)^2}{2l^2}) \tag{7}$$

where d is the euclidean distance between x and y. A main advantage of using RBF as a distance measure is that it decreases gradually the further the prediction is from the ground truth
clDice: The authors of [46] introduce a similarity measure centerlineDice (clDice), calculated by comparing the intersection of the prediction and ground truth masks and their morphological skeleta. See Appendix for further details.
$\boldsymbol{F1_\theta}$**:** We also consider a F1 scores with tolerance measure. In the experiments in the main body, we set $\theta = 10$.

5.3. Results and Discussion

(Q1) The modelling assumptions in Cracktal are plausible: The top half of table 1 shows the performance of models when evaluated on SegCODEBRIM. Despite training with real-world data and annotations of the multi-source dataset, U-Net (MultiSet) achieves an F1 score of 25.6%, which is worse than the performance of U-Net trained with synthetic Cracktal data. A similar trend can be observed on all metrics except $F1_{\theta=10}$, where U-Net (MultiSet) outperforms the baseline U-Net only marginally. We hypothesize that the overall worse performance achieved by U-Net (MultiSet) can be explained by the fact that the used training dataset comes from a variety of sources that tend to feature inconsistent annotation styles. More generally, U-Net (MultiSet) achieves a higher number of false positives and detects other anomalies present on concrete surfaces compared to U-Net, as evidenced by clDice and Hausdorff distance scores. These results underscore the significance of accurate labeling, which is guaranteed in simulation. Thus,

	Model	$F1(\uparrow)$	$F1_{\theta=10}(\uparrow)$	$clDice(\uparrow)$	$HDF_{Euc}(\downarrow)$	$HDF_{RBF}(\downarrow)$
SegCODEBRIM	U-Net [42]	29.6 ± 3.1	35.8 ± 1.8	38.5 ± 0.9	40.2 ± 13.2	53.6 ± 7.1
	U-Net (MultiSet)	25.6 ± 1.7	37.2 ± 0.9	28.5 ± 3.7	53.9 ± 10.7	66.5 ± 1.2
	U-Net (Sim+Real)	33.4 ± 3.8	35.5 ± 4.4	51.3 ± 4.2	$\mathbf{23.5 \pm 3.6}$	$\mathbf{42.7 \pm 5.3}$
	Attn-U-Net [38]	31.2 ± 1.1	37.5 ± 4.5	43.1 ± 6.2	34.7 ± 10.6	51.7 ± 5.5
	TransU-Net [9]	28.4 ± 1.8	31.3 ± 2.3	43.6 ± 2.0	25.6 ± 1.1	46.8 ± 2.2
	Multi-U-Net (D-SN)	31.9 ± 1.3	36.4 ± 0.9	45.3 ± 1.2	30.5 ± 7.1	46.4 ± 4.2
	Multi-U-Net (PMI)	32.7 ± 0.8	37.1 ± 1.1	47.2 ± 1.7	25.7 ± 6.4	43.9 ± 3.3
	CAP-Net	$\mathbf{37.3 \pm 1.5}$	$\mathbf{40.4 \pm 1.8}$	$\mathbf{53.6 \pm 1.5}$	$\mathbf{23.3 \pm 1.0}$	$\mathbf{42.6 \pm 1.8}$
Multi-source Set	U-Net [42]	42.6 ± 0.2	44.5 ± 0.2	71.5 ± 0.5	13.7 ± 1.3	33.5 ± 1.0
	U-Net (Sim+Real)	41.4 ± 3.7	48.5 ± 3.7	57.5 ± 6.9	45.1 ± 5.4	49.6 ± 4.1
	Attn-U-Net [38]	41.2 ± 2.2	43.2 ± 2.6	69.4 ± 1.7	14.9 ± 1.0	35.3 ± 2.7
	TransU-Net [9]	28.4 ± 1.8	31.3 ± 2.3	43.6 ± 2.0	22.0 ± 1.1	46.9 ± 2.2
	Multi-U-Net (D-SN)	42.5 ± 2.9	44.9 ± 2.5	68.9 ± 3.8	13.5 ± 1.8	33.6 ± 1.2
	Multi-U-Net (PMI)	41.0 ± 2.1	43.1 ± 1.5	66.9 ± 5.8	14.1 ± 1.4	32.7 ± 1.2
	CAP-Net	$\mathbf{44.9 \pm 1.5}$	$\mathbf{46.9 \pm 1.8}$	$\mathbf{72.30 \pm 1.5}$	$\mathbf{13.2 \pm 1.0}$	$\mathbf{31.5 \pm 1.8}$
	U-Net (MultiSet) $*$	$\mathbf{68.4 \pm 0.3}$	$\mathbf{82.8 \pm 0.6}$	$\mathbf{88.4 \pm 1.1}$	$\mathbf{5.7 \pm 1.4}$	$\mathbf{13.9 \pm 0.8}$

Table 1. Performance comparison of different models on SegCODEBRIM (top half) and multi-source set (bottom half). The best performing models on each dataset are highlighted in bold, where multiple values are highlighted if they lie within statistical deviations. CAP-Net outperforms *all* baselines on the real-world SegCODEBRIM, despite only being trained on simulated Cracktal data. It even outperforms the MultiSet trained on real multi-source data. Note that we also mark both CAP-Net and U-Net (MultiSet) for the multi-source dataset, as the latter is trained on real-world in-domain data and provides an expectation of what could be achieved, which we mark with a $*$. Apart from this upper-bound, CAP-Net beats all other simulation based baseliness.

we find the plausibility of our modelling assumptions in the Cracktal simulator to be well supported.

(Q2) Auxiliary simulated tasks improve the generalization: We consider two auxiliary tasks: depth and surface normals prediction and estimation of pointwise mutual information, denoted by the trained Multi-U-Net (D-SN) and Multi-U-Net (PMI) in table 1 respectively. Both models outperform the baseline U-Net significantly, improving clDice by 7+ and the Euclidean Hausdorff distance measure by 10+ on SegCODEBRIM dataset. Similar trends can be observed for the other metrics. Clearly, the depth and surface normal maps predicted by Multi-U-Net (D-SN) provide valuable information about the 3D spatial structure and layout of the scene, thus improving generalization on real data. Similarly, estimating geometry information can be seen as an inductive bias; Multi-U-Net (PMI) learns representation that focus on the anomalies in the images.

However, both of these models are less robust than our CAP-Net, highlighted also by the fact that the Multi-U-Nets do not significantly outperform the baseline U-Net when evaluated on the multi-source data (bottom half of Table 1).

(Q3) CAP-Net's hybrid modelling effectively reduces the Sim2Real gap: Revisiting Table 1, CAP-Net clearly outperforms all approaches on SegCODEBRIM (top half of table). For instance, we observe an improvement of 7% in F1 and 11 in Hausdorff distance with RBF kernel compared to U-Net. Similarly, our model performs better than all the baselines on multi-source set (bottom half of table), except U-Net(Multiset), which has been trained with in-distribution training data. Overall, training on the real multi-source data is only beneficial when deploying in a closely related context, whereas the modelling of CAP-Net in conjunction with the Cracktal simulator provides a robust solution for widely applicable crack segmentation by adapting from purely synthetic data.

(Q4) All CAP-Net design choices contribute to performance improvements: The ablations in Table 2 shows the performance of different sub-modules of our system on SegCODEBRIM and multi-source set.

First, the style transfer provided by AdaIN improves the generalization to real world data compared to a simple U-Net on SegCODEBRIM (U-Net(AdaIN) vs CAP-Net w/o AdaIn), but leads to statistically insignificant performance change on multi-source data. Second, the addition of a second encoder branch (bottom half of figure 3) that receives affinity scores based on PMI as input further increases the performance on most metrics, even when the branches are not contrasted (CAP-Net w/o CL). For instance, we obtain a 1.5% improvement in F1 and 3 on the RBF Hausdorff distance on SegCODEBRIM. Third, the subsequent addition of the consistency loss leads to consolidated segmentation maps between both encoders and improves performance on various metrics (CAP-Net w/o CL vs. "full" CAP-Net). We obtain a 5% improvement in F1 and decrease of 6 in Haus-

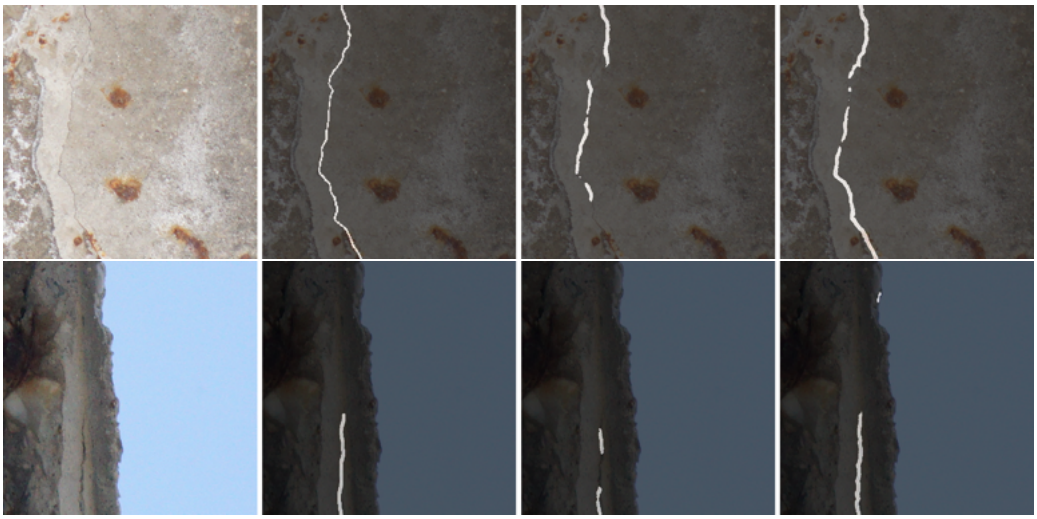

Figure 4. Qualitative examples on SegCODEBRIM from left to right: input image, ground-truth, U-Net, CAP-Net (ours). Images are compressed for view in pdf. Further examples can be found in the appendix.

	Model	$F1(\uparrow)$	$F1_{\theta=10}(\uparrow)$	$clDice(\uparrow)$	$HDF_{Euc}(\downarrow)$	$HDF_{RBF}(\downarrow)$
SegCODEBRIM	U-Net	29.5 ± 3.1	35.7 ± 1.8	38.5 ± 0.9	40.2 ± 13.2	53.5 ± 7.0
	U-Net (PMI)	28.4 ± 1.6	30.7 ± 2.5	44.8 ± 1.9	31.2 ± 2.1	53.3 ± 2.7
	U-Net (AdaIn)	31.4 ± 4.2	33.8 ± 4.3	48.4 ± 5.1	26.4 ± 5.2	48.5 ± 6.6
	CAP-Net w/o CL	32.6 ± 2.8	35.4 ± 1.3	48.7 ± 1.8	23.8 ± 11.5	46.0 ± 1.2
	CAP-Net w/o AdaIn	31.9 ± 4.1	40.1 ± 1.9	40.0 ± 1.7	41.0 ± 14.7	52.2 ± 0.8
	CAP-Net	$\mathbf{37.3 \pm 1.5}$	$\mathbf{40.4 \pm 1.8}$	$\mathbf{53.6 \pm 1.5}$	$\mathbf{21.7 \pm 1.0}$	$\mathbf{40.4 \pm 1.8}$
Multi-source Set	U-Net	42.6 ± 0.2	44.5 ± 0.2	71.4 ± 0.5	13.7 ± 1.3	33.5 ± 1.0
	U-Net (PMI)	40.6 ± 0.9	42.4 ± 1.1	68.4 ± 0.5	18.3 ± 0.5	37.8 ± 1.1
	U-Net (AdaIn)	40.6 ± 1.6	42.0 ± 1.5	70.7 ± 1.2	14.1 ± 1.4	36.1 ± 2.7
	CAP-Net w/o CL	42.0 ± 8.1	43.9 ± 8.6	69.5 ± 7.7	14.4 ± 3.9	34.6 ± 9.1
	CAP-Net w/o AdaIn	$\mathbf{45.2 \pm 2.4}$	$\mathbf{47.1 \pm 1.6}$	$\mathbf{72.6 \pm 0.7}$	$\mathbf{13.1 \pm 0.6}$	$\mathbf{30.5 \pm 2.3}$
	CAP-Net	$\mathbf{44.9 \pm 1.5}$	$\mathbf{46.9 \pm 1.8}$	$\mathbf{72.3 \pm 1.5}$	$\mathbf{13.2 \pm 1.0}$	$\mathbf{31.5 \pm 1.8}$

Table 2. Ablation study on SegCODEBRIM (top half) and multi-source set (bottom half). The best performing models on each dataset are highlighted in bold, where multiple values are highlighted if they lie within statistical deviations. Here, U-Net (PMI) and U-Net (AdaIn) refer to only the bottom and top parts of figure 3 respectively, whereas w/o CL and AdaIn denote the omission of the contrastive term and style transfer module. The "full" CAP-Net demonstrates that each designed component is crucial for SegCODEBRIM, whereas some components like AdaIn may be optional to perform well on the multi-source data.

dorff distance respectively on SegCODEBRIM.

The results of Table 2 empirically corroborate the efficacy of our design choices. The incorporation of PMI-based modeling approaches and purely data-driven U-Net style learning, augmented by consistency loss improves the appearance invariance of our model and thus leads to better generalization to out of distribution data, even when only trained on synthetic data.

6. Conclusion

In this paper, we introduced Cracktal, a flexible simulator for generating synthetic cracked concrete surface data with ground truth labels. Additionally, we proposed a hybrid design that combines data-driven models with single-image statistical estimation models, to fully leverage synthetic data. Our empirical validation demonstrates that this approach reduces the Sim2Real gap. Our work emphasizes the importance of fusing expert-based inductive biases with learning from simulated data and provide new domain to explore domain generalization and adaptation methods.

7. Acknowledgments

We acknowledge funding from the EU H2020 Research and Innovation Programme under grant agreement number 769066. This work was also supported by the Artificial Intelligence Systems Engineering Laboratory (AISEL) project under funding number 01IS19062, funded by the German Federal Ministry of Education and Research.

References

[1] Ikhlas Abdel-Qader, Osama Abudayyeh, and Michael E Kelly. Analysis of edge-detection techniques for crack identification in bridges. *Journal of Computing in Civil Engineering*, 17(4):255–263, 2003. 2

[2] Ikhlas Abdel-Qader, Sara Pashaie-Rad, Osama Abudayyeh, and Sherif Yehia. PCA-based algorithm for unsupervised bridge crack detection. *Advances in Engineering Software*, 37(12):771–778, 2006. 2

[3] Murad Al Qurishee, Weidong Wu, Babatunde Atolagbe, Joseph Owino, Ignatius Fomunung, and Mbakisya Onyango. Creating a dataset to boost civil engineering deep learning research and application. *Engineering*, 12(3):151–165, 2020. 1

[4] Hassan Abu Alhaija, Siva Karthik Mustikovela, Andreas Geiger, and Carsten Rother. Geometric image synthesis. In *Asian Conference on Computer Vision*, pages 85–100. Springer, 2018. 3

[5] Rabih Amhaz, Sylvie Chambon, Jérôme Idier, and Vincent Baltazart. Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection. *IEEE Transactions on Intelligent Transportation Systems*, 17(10):2718–2729, 2016. 2, 4, 6

[6] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. volume 31, 2018. 3

[7] Thomas O Binford and Tod S Levitt. Quasi-invariants: Theory and exploitation. In *DARPA Image Understanding Workshop*, pages 819–829, 1993. 1

[8] Wenming Cao, Qifan Liu, and Zhiquan He. Review of pavement defect detection methods. *Ieee Access*, 8:14531–14544, 2020. 2

[9] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021. 6, 7

[10] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1841–1850, 2019. 3, 6

[11] Jierong Cheng, Wei Xiong, Wenyu Chen, Ying Gu, and Yusha Li. Pixel-level crack detection using u-net. In *TENCON 2018-2018 IEEE region 10 conference*, pages 0462–0466. IEEE, 2018. 2

[12] Roland T Chin and Charles R Dyer. Model-based recognition in robot vision. *ACM Computing Surveys (CSUR)*, 18(1):67–108, 1986. 1

[13] Sungha Choi, Sanghun Jung, Huiwon Yun, Joanne T Kim, Seungryong Kim, and Jaegul Choo. Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11580–11590, 2021. 3

[14] Wooram Choi and Young-Jin Cha. Sddnet: Real-time crack segmentation. *IEEE Transactions on Industrial Electronics*, 67(9):8016–8025, 2019. 1, 2

[15] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014. 5

[16] Dimitris Dais, Ihsan Engin Bal, Eleni Smyrou, and Vasilis Sarhosis. Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning. *Automation in Construction*, 125:103606, 2021. 2

[17] Celso M de Melo, Antonio Torralba, Leonidas Guibas, James DiCarlo, Rama Chellappa, and Jessica Hodgins. Next-generation deep learning based on simulators and synthetic data. *Trends in Cognitive Sciences*, 2021. 2

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009. 5

[19] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. volume 32, 2019. 3

[20] 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. In *International Joint Conference on Neural Networks (IJCNN)*, pages 2039–2047, 2017. 2, 4, 6

[21] Timm Hess, Martin Mundt, Iuliia Pliushch, and Visvanathan Ramesh. A procedural world generation framework for systematic evaluation of continual learning. In *Advances in Neural Information Processing Systems*, 2021. 2

[22] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017. 5

[23] Daniel P Huttenlocher, Gregory A. Klanderman, and William J Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993. 6

[24] Phillip Isola, Daniel Zoran, Dilip Krishnan, and Edward H Adelson. Crisp boundary detection using pointwise mutual information. In *European Conference on Computer Vision*, pages 799–814. Springer, 2014. 4

[25] Chenfanfu Jiang, Yixin Zhu, Siyuan Qi, Siyuan Huang, Jenny Lin, Xiongwen Guo, Lap-Fai Yu, Demetri Terzopoulos, and Song-Chun Zhu. Configurable, photorealistic image rendering and ground truth synthesis by sampling stochastic grammars representing indoor scenes. *arXiv preprint arXiv:1704.00112*, 2, 2017. 3, 6

[26] Dongho Kang, Sukhpreet S Benipal, Dharshan L Gopal, and Young-Jin Cha. Hybrid pixel-level concrete crack segmentation and quantification across complex backgrounds using deep learning. *Automation in Construction*, 118:103291, 2020. 1

[27] Davood Karimi and Septimiu E Salcudean. Reducing the hausdorff distance in medical image segmentation with convolutional neural networks. *IEEE Transactions on Medical Imaging*, 39(2):499–513, 2019. 6

[28] Myeongjin Kim and Hyeran Byun. Learning texture invariant representation for domain adaptation of semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12975–12984, 2020. 3

[29] 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. 1, 2

[30] Jacob König, Mark Jenkins, Mike Mannion, Peter Barrie, and Gordon Morison. What's cracking? a review and analysis of deep learning methods for structural crack segmentation, detection and quantification. *arXiv preprint arXiv:2202.03714*, 2022. 1, 2

[31] David Lattanzi and Gregory R Miller. Robust automated concrete damage detection algorithms for field applications. *Journal of Computing in Civil Engineering*, 28(2):253–262, 2014. 2

[32] JEFFREY LeBlanc, Michael A Gennert, Norman Wittels, and David Gosselin. Analysis and generation of pavement distress images using fractals. *Transportation Research Record*, 1311:158–165, 1991. 3

[33] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2018. 3

[34] Jun Ma, Jianan Chen, Matthew Ng, Rui Huang, Yu Li, Chen Li, Xiaoping Yang, and Anne L Martel. Loss odyssey in medical image segmentation. *Medical Image Analysis*, 71:102035, 2021. 6

[35] Keith Man and Javaan Chahl. A review of synthetic image data and its use in computer vision. *Journal of Imaging*, 8(11):310, 2022. 2

[36] Jiteng Mu, Weichao Qiu, Gregory D Hager, and Alan L Yuille. Learning from synthetic animals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12386–12395, 2020. 2

[37] Martin Mundt, Sagnik Majumder, Sreenivas Murali, Panagiotis Panetsos, and Visvanathan Ramesh. Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11196–11205, 2019. 1, 2, 3, 4

[38] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018. 6, 7

[39] Duo Peng, Yinjie Lei, Munawar Hayat, Yulan Guo, and Wen Li. Semantic-aware domain generalized segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2594–2605, 2022. 3

[40] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision*, pages 102–118. Springer, 2016. 2

[41] Rodrigo Rill-García, Eva Dokladalova, and Petr Dokládal. Syncrack: Improving pavement and concrete crack detection through synthetic data generation. In *17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP'22)*, 2022. 2

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241. Springer, 2015. 4, 7

[43] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 3234–3243, 2016. 2

[44] Fatemeh Sadat Saleh, Mohammad Sadegh Aliakbarian, Mathieu Salzmann, Lars Petersson, and Jose M Alvarez. Effective use of synthetic data for urban scene semantic segmentation. In *European Conference on Computer Vision*, pages 86–103. Springer, 2018. 3

[45] 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. 2, 4, 6

[46] Suprosanna Shit, Johannes C Paetzold, Anjany Sekuboyina, Ivan Ezhov, Alexander Unger, Andrey Zhylka, Josien PW Pluim, Ulrich Bauer, and Bjoern H Menze. cldice-a novel topology-preserving loss function for tubular structure segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16560–16569, 2021. 6

[47] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 109–117, 2017. 2

[48] Tobi Vaudrey, Clemens Rabe, Reinhard Klette, and James Milburn. Differences between stereo and motion behaviour on synthetic and real-world stereo sequences. In *International Conference Image and Vision Computing*, pages 1–6, 2008. 2

[49] David Vazquez, Antonio M Lopez, Javier Marin, Daniel Ponsa, and David Geronimo. Virtual and real world adaptation for pedestrian detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(4):797–809, 2013. 2

[50] VSR Veeravasarapu, Constantin Rothkopf, and Ramesh Visvanathan. Adversarially tuned scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2587–2595, 2017. 3

[51] Sanjeev Kumar Verma, Sudhir Singh Bhadauria, and Saleem Akhtar. Review of nondestructive testing methods for condi-

tion monitoring of concrete structures. *Journal of Construction Engineering*, 2013(2008):1–11, 2013. 1

[52] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022. 3

[53] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 3

[54] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. Learning from synthetic data for crowd counting in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8198–8207, 2019. 2

[55] Jiaolong Xu, Sebastian Ramos, David Vázquez, and Antonio M López. Domain adaptation of deformable part-based models. *IEEE Transactions on Pattern Analysis and Machine intelligence*, 36(12):2367–2380, 2014. 2

[56] Jia Xu, Cheng Yuan, Jiaxuan Gu, Jian Liu, Jiong An, and Qingzhao Kong. Innovative synthetic data augmentation for dam crack detection, segmentation, and quantification. *Structural Health Monitoring*, 22(4):2402–2426, 2023. 2

[57] Tomoyuki Yamaguchi, Shingo Nakamura, Ryo Saegusa, and Shuji Hashimoto. Image-based crack detection for real concrete surfaces. *IEEJ Transactions on Electrical and Electronic Engineering*, 3(1):128–135, 2008. 2

[58] Huaquan Ying, Rafael Sacks, and Amir Degani. Synthetic image data generation using bim and computer graphics for building scene understanding. *Automation in Construction*, 154:105016, 2023. 2

[59] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2100–2110, 2019. 3

[60] Guanghao Zhai, Yasutaka Narazaki, Shuo Wang, Shaik Althaf V Shajihan, and Billie F Spencer Jr. Synthetic data augmentation for pixel-wise steel fatigue crack identification using fully convolutional networks. *Smart Struct Syst*, 29(1):237–250, 2022. 2

[61] Lei Zhang, Fan Yang, Yimin Daniel Zhang, and Ying Julie Zhu. Road crack detection using deep convolutional neural network. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 3708–3712. IEEE, 2016. 2, 4, 6

[62] Sicheng Zhao, Xiangyu Yue, Shanghang Zhang, Bo Li, Han Zhao, Bichen Wu, Ravi Krishna, Joseph E Gonzalez, Alberto L Sangiovanni-Vincentelli, Sanjit A Seshia, et al. A review of single-source deep unsupervised visual domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):473–493, 2020. 3

[63] 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. 2, 4, 6