OMNICRACK30K: A Benchmark for Crack Segmentation and the Reasonable Effectiveness of Transfer Learning

Christian Benz and Volker Rodehorst
Bauhaus-Universität
Weimar, Germany
christian.benz@uni-weimar.de

Abstract

Large benchmarking datasets, such as ImageNet, COCO, Cityscapes, or ScanNet, have enormously promoted research in computer vision. For the domain of crack segmentation, no such large and well-maintained benchmark exists. Crack segmentation is characterized by the decentralized creation of datasets, almost all of which have their specific right to existence. Each dataset covers a different aspect of the surprisingly complex landscape of materials, acquisition conditions, and appearances linked to crack segmentation. The OMNICRACK30K *dataset forms the first large-scale, systematic, and thorough approach to provide a sustainable basis for tracking methodical progress in the field of crack segmentation. It contains 30k samples from over 20 datasets summing up to 9 billion pixels in total. Featuring materials as diverse as asphalt, ceramic, concrete, masonry, and steel, it paves the road towards* universal crack segmentation*, a currently under-explored topic. Experiments indicate the effectiveness of transfer learning for crack segmentation:* nnU-Net *achieves a mean* $clIoU_{4px}$ *of 64% outperforming all other approaches by at least 10% points.*

1. Introduction

The societal impact of crack detection cannot be underestimated. Recurring bridge collapses underline the importance of regular and thorough structural inspection. Inspection campaigns, however, consume a significant amount of resources: they lead to route closure, require special technical equipment, and put human lives at risk. The technological state of emerging imaging platforms (*e.g.*, drones) has reached a maturity which renders large-scale image acquisition at structures possible.

The multitude and complexity of the captured images exceed the temporal and cognitive capacities of human experts. Automation approaches can, thus, productively support the experts in deriving valuable information from the data and thereby facilitating the inspection process. Cracks are among the most relevant defects to provide substantial insights into the conditional state of a structure. As a consequence image-based crack detection forms a key ingredient in enhancing the process of structural inspection.

Even though other approaches have been proposed, semantic segmentation is the prevailing paradigm for crack detection. It is commonly referred to as *crack segmentation*. Crack segmentation is an unsolved problem and comes with a number of challenges including robustness and universality. By no means is the relevance of crack segmentation restricted to the domain of civil engineering: computer vision has investigated line-like features usually as object boundaries, but rarely as objects themselves. Road detection in satellite imagery and the recognition of blood vessels in medical imaging are arguably the most analogous tasks to crack segmentation in that sense.

The contributions of this work are threefold: (1) the systematic and thorough review and analysis of available datasets for crack segmentation. (2) The composition and provision of the large-scale benchmarking OMNICRACK30K dataset for universal crack segmentation based on this analysis. (3) The detailed investigation and benchmarking of transfer learning for crack segmentation.

2. Related Work

[36] provide a survey on crack detection before artificial neural networks (ANN) became the dominant approach. Edge detection, morphological operations, filtering, and thresholding were among the most frequently used techniques [1, 30, 37, 41, 45, 48, 56]. The CrackTree approach [59] constructs a minimum spanning tree over previously identified crack seeds. An ensemble of decision trees called CrackForest is used by [42] for crack classification.

Since 2017, artificial neural networks (ANN) have emerged as the dominant approach for crack detection. [18] conducted a study comparing different training configura-

tions of AlexNet [26] with six edge detectors, including Sobel, LoG, and Butterworth. Experiments on the SDNET dataset [19] indicated the superiority of ANN and the effectiveness of transfer learning. Other approaches, proposed by [11, 12, 57], involve using a classification CNN combined with a sliding window to process larger images and/or improve localization. [54] popularized the transition to fully-convolutional networks (FCN) [34] for crack segmentation. Based on SegNet [3], DeepCrackZ was designed [60]: a separate fusion logic with individual, scale-wise losses supports preserving thin structures. The conceptually similar approach DeepCrackL[1] is suggested by [32]. In the style of deeply-supervised nets (DSN) [28], losses are computed for intermediate side-outputs to make use of fine details and anti-noise capabilities alike. The outputs undergo post-processing with guided filtering (GF) [23] and conditional random fields (CRF) [58]. A U-Net [40] with focal loss [29] is reported to perform superiorly compared to a simpler FCN design [33]. [52] propose the feature pyramid and hierarchical boosting network (FPHBN). It extends holistically-nested edge detection (HED) [51] by a feature pyramid module to incorporate and propagate context information to lower levels. The hierarchical boosting supports the inter-level communication within the FPHBN. [31] develop CrackFormer, which is a transformer-based approach to crack segmentation. For that purpose, the convolutional layers of VGG [44] are replaced by a self-attention logic. To increase the crack sharpness, a scaling-attention block is suggested. [6] propose a re-trained version of the hierarchical multi-scale attention network by [49] called HMA, which mitigates the scale sensitivity of cracks. The results are aggregated based on the attention to cracks on different levels of scales. In order to preserve the continuity of cracks, [38] suggest TOPO loss, which uses maximin paths to mitigate discontinuities between cracks. An oriented bounding box approach, named CrackDet, has recently been proposed by [15]. [8] and [27] emphasize the usefulness of transfer learning and compose smaller crack datasets into larger ones, Conglo, and CrackSeg9k. [27] compare a number of approaches, including Pix2Pix, SWIN, and MaskRCNN. DeepLabV3+ [14] with a ResNet-101 backbone outperformed the other methods. [8] confirm that DeepLabV3+ is an effective method for crack segmentation.

3. OmniCrack30k Dataset

The data culture in computer vision, as typically practiced in top-tier conferences, is centralized and top-down: major benchmarking datasets are monolithic products created by established research institutions, often in cooperation with renowned industry partners. Datasets for crack segmentation, on the other hand, typically emerge from a more democratic 'grassroots' movement in a decentralized and bottom-up fashion. Even though complicating benchmarking, the singular datasets usually have their right of existence by reflecting a specific challenge linked to crack segmentation. Composing these special cases into the overarching OmniCrack30k dataset reduces biases, enables benchmarking, and paves the road to a more general 'universal' crack segmentation.

[1] DeepCrackZ and DeepCrackL are used for disambiguation since both were originally called DeepCrack.

Dataset	Image Sizes	Total Samples	Crack Coverage Original	Crack Coverage Centerline	Primary Reference
AEL	[462, 1000]×[311, 991]	58	0.67%	0.32%	[2]
BCL	256×256	11,000	2.24%	0.30%	[55]
Ceramic	256×256	100	2.05%	0.60%	[25]
CFD	320×480	118	1.62%	0.47%	[42]
Conglomerate	448×448	9,584	3.18%	0.31%	[8]
CRACK500	[1440, 1936]×[2560, 2592]	447	2.84%	0.11%	[57]
CrackLS315	512×512	315	0.25%	0.25%	[60]
CrackSeg9k	400×400	6,315	2.48%	0.21%	[27]
CrackTree200	600×800	206	0.32%	0.32%	[59]
CrackTree260	[600, 720]×[800, 960]	260	0.46%	0.45%	[60]
CRKWH100	512×512	100	0.36%	0.36%	[60]
CrSpEE	[81, 4160]×[116, 4608]	1,220	0.84%	0.11%	[4]
CSSC	768×768	195	2.42%	0.20%	[53]
DeepCrack	[384, 544]×[384, 544]	537	3.54%	0.37%	[32]
DIC	256×256	1,060	1.79%	0.41%	[39]
GAPS384	1080×1920	384	0.36%	0.05%	[52]
Kaggle11k	448×448	11,298	3.50%	0.75%	[35]
Khanh11k	448×448	11,298	3.50%	0.75%	[22]
LCW	[237, 4608]×[256, 5184]	3,773	0.02%	0.01%	[8]
Masonry	224×224	240	4.21%	0.41%	[17]
S2DS	1024×1024	743	0.26%	0.03%	[6]
Stone331	512×512	331	0.10%	0.10%	[60]
TopoDS	256×256	7,180	0.31%	0.31%	[38]
UAV75	512×512	75	1.42%	0.24%	[7]

Table 1. Overview of useful, relevant, and accessible datasets for crack segmentation as of Aug 16, 2023. The column 'crack coverage' represents the area covered by the pixel-wise crack labels across all images of the respective dataset. 'Centerline' derives from the transformation of the original labels to a centerline representation.

3.1. Available Datasets

[9] and [27] attempted to list or combine different datasets, in yet incomplete fashion. Tab. 1 provides a quantitative overview of available, relevant, and useful datasets for crack segmentation, briefly outlined in the following:

AEL: Published by [2], contained subsets: AIGLE_RN, ESAR, and LCMS. LCMS has low-quality images. AEL features images of variable size showing *asphalt* cracks.
BCL: *Bridge Crack Library* (BCL), published by [55]. Consists of patches from larger images. Cracks are shown in a macro-like fashion, which accounts for the blur visible in many samples. 2,036 samples show cracks in *steel*, 5,769 in *concrete* and *stone*, and 3,195 are negative samples for crack-like artifacts. No information on the data splits. Featuring only synthetic cracks, BCL 2.0 is disregarded.

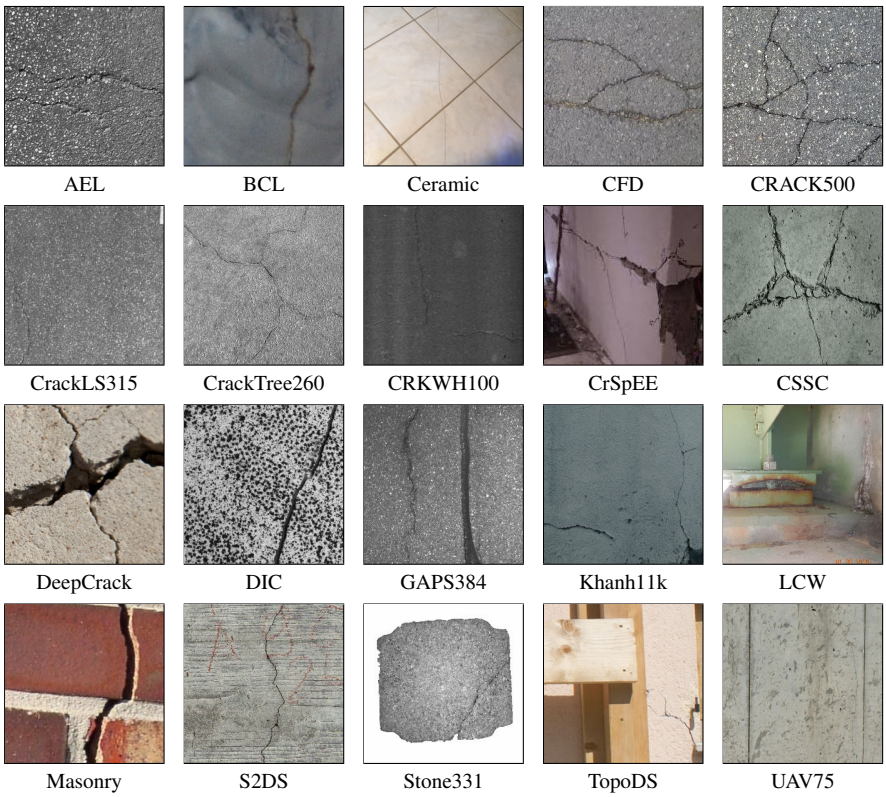

Figure 1. Visual impression of the crack segmentation datasets which are combined into the OmniCrack30k dataset. Images are cropped to square. Best viewed on screen.

Ceramic: Published by [25]. No labels provided for the test set. The images show cracks in a variety of *ceramic* tiles.

CFD: Short for *CrackForest* dataset [42]. The images show cracks almost exclusively in *asphalt* surfaces. Exact dataset splits are not reported; train/test share is 60%/40%.

Conglomerate: Compiled and used by [8]. Based on Khanh11k, it forms a collection of datasets, including CFD, CRACK500, CrackTree200, DeepCrack, GAPs, etc.

CRACK500: The dataset is provided by [52], its origins are traced to [57]. The images by [52] are smaller than the size reported by [57]. The images vary in size and exclusively show *pavement* cracks of variable widths.

CrackLS315: Published by [60]. The images show *asphalt* cracks with slight illumination inhomogeneities.

CrackSeg9k: [27] combined and transformed the Khanh-11k, Masonry, and Ceramic datasets. The downloadable dataset has ∼6k samples instead of 9k (Aug 16, 2023).

CrackTree200: Also referred to as CrackTree206, published by [59]. Features *asphalt* cracks in relatively homogeneous conditions.

CrackTree260: Extension of CrackTree200, published by [60]. Images with lower quality and cracks with a sophisticated net-like structure were added. All images show *asphalt* cracks under relatively homogeneous conditions.

CRKWH100: Published by [60]. The images show *asphalt* cracks with slight illumination inhomogeneities. It contains five images with white cracks.

CrSpEE: The *Crack and Spalling Dataset in the Context of Extreme Events* (CrSpEE) is designed for instance segmentation of cracks and spalling [4]. It shows cracks (and spalling) 'in the wild', i.e., under challenging conditions with numerous distractors (people, context, etc.).

CSSC: The *Concrete Structure Spalling and Crack* dataset (CSSC) published by [53]. Duplicates resulting from flipping augmentations are removed. The images show cracks and spalling mostly in *concrete* and *stone*.

DeepCrack: Published with model DeepCrackL [32]. Features partly very wide cracks in different surfaces, e.g., concrete and stone. Some images are blurry.

DIC: Published by [39] to investigate digital image correlation (DIC) for deformable image matching. Features cracks on a *plastered* wallett (a specimen) with random black speckles under homogeneous, lab-like conditions.

GAPS384: [20], [46], and [47] published a series of datasets under the name GAPs (*German Asphalt Pavement Distress*) v1, v2, and 10m. GAPs v1 and GAPs v2 provide patch classification labels only. GAPs 10m contains very coarse segmentation masks. [52] created pixel-wise labels

for GAPs v1, released as GAPS384. The images show *asphalt* cracks under relatively homogeneous conditions.

LCW: The *Labeled Cracks in the Wild* (LCW) dataset features crack detection under challenging conditions [8]. Most images are of lower quality and show the structure's context, including support elements, vegetation, etc. A high number of negative samples is contained. To counteract disbalance, OMNICRACK30K uses positive samples only. The images almost exclusively show cracks in *concrete*.

Masonry: Features cracks in variable kinds of *masonry*, mostly brick walls [17]. The negative samples are not provided. The reconstructed train/test split is probably faulty.

Kaggle11k: A copy of Khanh11k without reference.

Khanh11k: The name refers to the repository owner. A collection of other datasets, including CrackTree200, CFD, CRACK500, DeepCrack, and parts of AEL. Labels were stored in JPG format, which probably accounts for artifacts in some labels. Many images were patched and anisotropically scaled, leading to distortions.

S2DS: *Structural Defects Dataset* (S2DS), published by [6]. Contains cracks alongside other classes representing real-world inspection scenarios of *concrete* structures. Due to the overall small number, also the negative samples are kept.

Stone331: Published by [60]. The images show cracks in *stone* under homogeneous conditions.

TopoDS: An 'in the wild' dataset with challenging images containing a multitude of distractors (other support structures, objects, etc.) [38]. The centerline annotations often are coarse and have offsets to the presumed centerline. Features cracks on various structures in a post-disaster scenario. Demolition edges of spallings are also labeled as crack.

UAV75: *Unmanned Aerial Vehicle* dataset (UAV75) [7]. Represents cracks in real-world, UAV-based inspection scenarios with fine, partially blurry cracks in *concrete* surfaces.

3.2. Overlap Analysis

It was observed that images from certain datasets frequently re-occur in others. For the overlap analysis all images are transformed into a perceptual embeddings of size $32\times32\times3$ with four rotational configurations. This preserves the image characteristics while abstracting from slight image manipulations. The *normalized overlap* score is computed as $|\{b : ||a-b|| < \tau\}|/|A|$, where a and b represent the perceptual embeddings and $|A|$ the number of samples in dataset A. In this case $\tau = 100$ rendered suitable.

Tab. 2 shows the results of the overlap analysis. The datasets BCL, CrackLS315, CRKWH100, DIC, LCW, Masonry, S2DS, Stone331, TopoDS, and UAV75 are not listed since they are mutually exclusive with all other datasets considered here. For CRACK500 and GAPS384, the patched version is used for overlap analysis. The overlapping properties between Kaggle11k and Khanh11k suggest that they are identical, which is the case. Furthermore, it can be observed that duplicates of the CFD images occur three times in Khanh11k. More problematically, they scatter over test and training, leaking information to the test set.

Dataset A \ Dataset B	AEL	Ceramic	CFD	Conglomerate	CRACK500	CrackSeg9k	CrackTree200	CrackTree260	CrSpEE	CSSC	DeepCrack	GAPS384	Kaggle11k	Khanh11k
	Normalized Overlap [%]													
AEL		–	–	–	–	–	–	–	–	–	–	–	52	52
Ceramic	–		–	–	–	–	–	–	–	1	–	–	–	–
CFD	–	–		100	–	–	–	–	–	–	–	–	300	300
Conglomerate	–	–	1		36	52	2	2	0	0	5	5	104	104
CRACK500	–	–	–	92		49	–	–	–	–	–	–	92	92
CrackSeg9k	–	–	–	79	29		–	–	–	–	–	6	103	103
CrackTree200	–	–	–	100	–	–		100	–	–	–	–	100	100
CrackTree260	–	–	–	79	–	–	79		–	–	–	–	79	79
CrSpEE	–	–	–	0	–	–	–	–		2	0	–	0	0
CSSC	–	0	–	1	–	–	–	–	1		1	–	1	1
DeepCrack	–	–	–	97	–	–	–	–	0	1		–	97	97
GAPS384	–	–	–	100	–	75	–	–	–	–	–		100	100
Kaggle11k	0	–	1	86	31	58	2	2	0	0	5	5		102
Khanh11k	0	–	1	86	31	58	2	2	0	0	5	5	102	

Table 2. Result of the overlay analysis. The share of dataset A contained in dataset B in %. Numbers of more than 100% indicate a duplication of samples.

It can be concluded that there are only a handful of genuine datasets that created images and distributed segmentation labels. The most prominent ones include CFD, CRACK500, CrackTree200, DeepCrack, and GAPs. Other genuine but less visible datasets are AEL, Ceramic. On the other hand, Conglomerate, CrackSeg9k, Kaggle11k, and parts of Khanh11k are mere data collections with disputable scientific value.

3.3. OmniCrack30k

The dataset used in this work is henceforth called OMNICRACK30K. The prefix 'Omni' is borrowed by [10], who proposed the Omni3D dataset, a combination of multiple dataset for 3D object detection. The suffix '30k' follows the establishing naming convention to inform about the approximate number of samples in the dataset.

Tab. 3 provides an overview of the OMNICRACK30K dataset. There are roughly 30k samples in total, with 22k for training, 3.3k for validation, and 4.6k for testing, resulting in a train/val/test split of roughly 75/10/15. BCL contributes the largest share of 11k samples, accounting for one third of all samples. The images in BCL, however, are comparatively small, leading to a less than 10% contribution in terms of pixels. On the other hand, CRACK500 and LCW consist of relatively large images, contributing about 20% each in terms of pixels. The Ceramic and Masonry subsets have the lowest share in pixels with 7M and 12M respectively, representing together less than 1% of all pixels. The CrackTree260 is not included in the test set since it was traditionally used for training only, *e.g.*, [60].

	Samples				Pixels (in million)			
	Train	Val	Test	Total	Train	Val	Test	Total
AEL	–	–	58	58	–	–	20.8	20.8
BCL	8,910	990	1,100	11,000	583.9	64.9	72.1	720.9
Ceramic	70	15	15	100	4.6	1.0	1.0	6.6
CFD	61	10	47	118	9.4	1.5	7.2	18.1
CRACK500	250	50	199	499	934.9	188.3	915.3	2,038.5
CrackLS315	–	–	315	315	–	–	82.6	82.6
CrackTree260	234	26	–	260	122.2	14.0	–	136.2
CRKWH100	–	–	100	100	–	–	26.2	26.2
CrSpEE	981	109	130	1,220	617.6	68.5	72.2	758.2
CSSC	101	12	73	186	59.6	7.1	43.1	109.7
DeepCrack	263	30	236	529	54.9	6.3	49.3	110.5
DIC	301	129	100	530	19.7	8.5	6.6	34.7
GAPS384	353	4	27	384	732.0	8.3	56.0	796.3
Khanh11k	3,750	417	737	4,904	752.6	83.7	147.9	984.3
LCW	781	87	377	1,245	778.8	91.1	947.4	1,817.3
Masonry	130	14	96	240	6.5	0.7	4.8	12.0
S2DS	563	87	93	743	590.3	91.2	97.5	779.1
Stone331	–	–	331	331	–	–	86.8	86.8
TopoDS	5,360	1,287	533	7,180	351.3	84.3	34.9	470.5
UAV75	50	10	15	75	13.1	2.6	3.9	19.7
Total	22,158	3,277	4,582	30,017	5,631.6	721.9	2,675.5	9,029.0

Table 3. Overview of the OMNICRACK30K dataset for crack segmentation. It consists of 20 subsets. The number of samples and pixels are shown as well as the dataset splits into train, validation, and test set.

The under- and overrepresentation of subsets is inherent to the landscape of crack segmentation. Unlike others [27], no attempt is made to unify the subsets in terms of size or other properties. The subsets are deliberately kept as unmodified as possible, especially in the case of the test sets. The goal is to maintain benchmarking comparability of these specific subsets independent of OMNICRACK30K.

The following modifications are performed. The duplicates identified in the overlap analysis are removed from derivative datasets. Therefore, some datasets from Tab. 3 show fewer samples than in Tab. 1. As pointed out by [27], the labels of the Khanh11k dataset are fixed by morphological processing. For those datasets without validation set, a fraction (roughly 10%) of the training set is split off and held out for validation. To counteract the imbalance and to avoid class inconsistencies, the negative samples from the LCW training set are removed.

4. Benchmarking

4.1. Metrics

It is convincingly argued by [6] that standard *intersection-over-union* (IoU) is suited for blob-like objects, while cracks form rather elongated, line-like structures. Accordingly, the *centerline intersection-over-union* (clIoU) metric[2] is proposed for evaluating crack segmentation. It is defined as:

$$\text{clIoU}_\tau = \frac{|\text{TP}_{\text{cl}}|}{|\text{TP}_{\text{cl}}| + |\text{FP}_{\text{cl}}| + |\text{FN}_{\text{cl}}|} \tag{1}$$

$$\text{TP}_{\text{cl}} = S_T \bigcap \left[S_P \oplus K_\tau\right]$$

$$\text{FP}_{\text{cl}} = S_P \setminus \left[S_P \bigcap \left[S_T \oplus K_\tau\right]\right]$$

$$\text{FN}_{\text{cl}} = S_T \setminus \text{TP}_{\text{cl}}$$

TP_{cl}, FP_{cl}, and FN_{cl} denote the true positives, false positives, and false negatives for clIoU. The skeleton of the ground truth is denoted S_T, the skeleton of the prediction S_P. Skeletonization or thinning can be performed with off-the-shelf methods, here [21] is used[3]. K_τ refers to the dilation kernel, for instance a circle with radius τ. The symbol $\oplus$ denotes the morphological operator for dilation. Dilation implements the positional tolerance of the clIoU. TP_{cl} is the intersection of the true skeleton with the dilated skeleton of the predictions, FP_{cl} the skeleton of the predictions without the intersection of predicted skeletons and the dilated true skeletons. FN_{cl} is computed from the true skeleton without the TP_{cl}. The clDice metric, proposed by [43], is calculated in similar fashion.

4.2. Baselines

Two groups of baselines are selected for benchmarking. The results of six domain-specific models regularly used for crack segmentation are presented. These models are specifically tuned on one or more crack segmentation datasets, thereby assumed to have learned powerful crack-specific representations. The models include DeepCrackZ [60], DeepCrackL [32], CrackFormer [31], HMA [6], Conglo [8], and TOPO [38]. For more details, see Sec. 2. Furthermore, transfer learning is applied to four general-purpose approaches widely used for semantic segmentation. The models are trained on OMNICRACK30K.

The self-configuring approach nnU-Net [24] won multiple challenges in medical imaging. Based on a set of fixed, rule-based, and empirical parameters, a suitable configuration of the architecture, data processing, and training parameters is inferred. The patch size, 256×256 for OMNICRACK30K, is deduced from the dataset and based on the median image size. U-Net [40] serves as architectural template, which is adapted to the inferred patch size. For OMNICRACK30K, the encoder consists of six convolutional blocks that transform the input into a $512 \times 4 \times 4$ feature representation. Pooling is performed by convolutional layers of stride two. The decoder features six complementary convolutional blocks, each starting with a transposed convolution to upsample the input. Skip connections link intermediate encoder outputs to the respective decoder blocks.

[2] Originally called ltIoU for *line-based tolerant IoU* [6].

[3] https://scikit-image.org/docs/stable/api/skimage.morphology.html#skimage.morphology.thin

	Trainset	AEL	BCL	Ceramic	CFD	CRACK500	CrackLS315	CRKWH100	CrSpEE	CSSC	DeepCrack	DIC	GAPS384	Khanh11k	LCW	Masonry	S2DS	Stone331	TopoDS	UAV75	*Average*
# Test Samples		58	1100	15	47	199	315	100	130	73	236	100	27	737	377	96	93	331	533	15	241
DeepCrackZ	CrackTree260	61.8	40.9	19.2	83.8	3.2	70.6	82.8	3.3	3.7	37.8	20.3	40.8	44.3	1.0	16.9	4.3	31.7	13.9	19.1	*31.5*
DeepCrackL	DeepCrack	19.5	22.9	19.7	31.5	3.0	7.1	6.2	2.7	2.5	51.3	7.9	1.7	27.0	0.2	34.2	7.9	11.5	6.8	10.9	*14.4*
CrackFormer	CrackTree260	44.6	15.9	18.1	76.4	3.8	43.6	70.8	3.1	1.7	27.4	19.9	5.1	20.2	0.1	18.2	2.7	13.8	7.5	23.4	*21.9*
HMA	S2DS	30.2	44.6	16.6	73.5	15.9	17.2	45.3	10.1	43.1	58.5	0.0	21.6	70.0	4.0	22.5	85.7	8.3	18.6	41.3	*33.0*
Conglo	Conglomerate	58.0	56.5	29.1	86.6	43.8	57.5	80.3	14.6	26.6	84.3	47.4	59.8	94.4	5.6	36.2	45.8	43.2	16.5	55.7	*49.6*
TOPO	TopoDS	38.2	53.4	45.5	65.9	17.4	34.3	60.8	17.8	29.8	60.3	22.9	32.1	68.0	6.2	35.6	42.8	45.9	38.1	53.8	*40.5*
nnU-Net	OmniCrack30k	79.3	82.3	50.8	88.7	45.3	70.7	88.1	24.3	43.6	82.4	89.0	64.2	91.0	11.8	78.2	73.5	69.2	5.8	72.8	*63.7*
DeepLabV3	OmniCrack30k	51.5	55.5	20.4	76.3	21.2	34.0	51.9	10.3	32.7	67.9	57.0	31.2	76.8	2.9	19.8	47.4	46.5	24.9	38.7	*40.4*
BEiT	OmniCrack30k	62.6	63.8	21.9	82.6	28.3	46.9	60.0	10.6	32.0	73.5	40.7	41.5	73.4	3.9	25.4	49.5	49.3	23.9	31.1	*43.2*
Mask2Former	OmniCrack30k	76.3	61.4	52.4	84.8	22.4	67.2	78.5	15.5	38.1	75.9	55.0	37.5	82.8	5.5	34.3	54.2	62.5	30.7	69.9	*52.9*
Average		*52.2*	*49.7*	*29.4*	*75.0*	*20.4*	*44.9*	*62.5*	*11.2*	*25.4*	*61.9*	*36.0*	*33.6*	*64.8*	*4.1*	*32.1*	*41.4*	*38.2*	*18.7*	*41.7*	*39.1*
Max		*79.3*	*82.3*	*52.4*	*88.7*	*45.3*	*70.7*	*88.1*	*24.3*	*43.6*	*84.3*	*89.0*	*64.2*	*94.4*	*11.8*	*78.2*	*85.7*	*69.2*	*38.1*	*72.8*	*66.4*

Table 4. Performance measured in $\text{clIoU}_{4\text{px}}$ in % with tolerance radius $\tau = 4$px on the test subsets. 'Average' refers to the unweighted mean over the datasets and approaches respectively. 'Max' represents the maximum performance reached by the compared approaches.

	Tolerance τ							
	0 px	1 px	2 px	4 px	8 px	16 px	32 px	64 px	128 px
DeepCrackZ	3.2	7.4	9.6	12.0	14.0	15.4	16.7	18.7	22.3
DeepCrackL	1.6	4.4	6.4	8.6	10.8	12.9	15.1	18.2	22.9
CrackFormer	1.6	4.2	5.8	7.9	10.1	12.4	15.0	17.3	21.5
HMA	8.0	21.8	29.2	35.3	39.1	41.9	44.6	48.0	52.5
Conglo	7.5	24.2	38.7	52.3	59.3	64.3	68.4	72.2	76.3
TOPO	5.3	16.6	26.8	40.8	51.0	56.9	62.3	69.3	77.5
nnU-Net	16.2	38.2	49.4	58.5	64.3	67.9	70.2	72.1	74.0
DeepLabV3	5.4	16.8	26.8	40.6	53.4	61.9	68.1	74.2	80.5
BEiT	5.7	18.0	29.1	43.1	54.6	62.0	67.7	73.2	78.5
Mask2Former	5.7	18.2	29.9	46.0	59.5	68.8	74.3	79.0	83.4

Table 5. The clIoU metric in % computed with different radii of tolerance τ on the validation set.

Throughout the model, leaky ReLU activations, instance normalization, and dropout are used. The loss is based on a combination of cross-entropy and Dice loss. Five models of the same configuration are trained on different splits of the training set in five-fold cross-validation. Based on the fold performances measured by the Dice coefficient, nnU-Net finally derives a powerful ensemble.

DeepLabV3 [13] is an established baseline that makes elaborate use of atrous convolutions. BEiT [5] is a vision transformer pre-trained by masked image modeling. BEiT uses the UPerNet [50] framework for semantic segmentation which implements a feature pyramid network with a pyramid pooling module. Mask2Former [16] is an approach to unify the recognition tasks of object detection, instance segmentation, and semantic segmentation.

5. Results

In this section, the results of the benchmarking approaches on OMNICRACK30K and its subsets are presented.

5.1. Positional Tolerance

To assess the effects of different positional tolerances of the evaluation metric, the clIoU is computed with radii zero and $\tau = 2^n$ where $n \in \{0, 1, \ldots, 6, 7\}$. Tab. 5 shows the results. It is observed that the transition from 2px to 4px yields a significant leap in values, while for larger tolerances, smaller increases are detected. This transition indicates that at a 4px tolerance, the models are capable of operating a major part of their performance. This corresponds to the tolerance of $\tau = 4$px proposed by [6]. It is considered a plausible compromise between positional accuracy and classification slack: 4px is still rigid enough to allow for subsequent processing, such as crack width estimation, while reasonably accounting for inaccuracies induced in the annotation process. The selection of a specific tolerance value, however, is an empirical question and a matter of debate.

5.2. Subset-Specific Performance

Tab. 4 shows the results of the benchmarking approaches on the test subsets of OMNICRACK30K. Note that only the general models are trained on OMNICRACK30K, while the crack-specific approaches used their specific datasets. Averaged over all subsets, nnU-Net performs the best by achieving a $\text{clIoU}_{4\text{px}}$ of 64%. For 14 out of 19 subsets, it outperforms all other approaches. Furthermore, it is relatively close to the 66% achieved by a hypothetical optimal ensemble over all models. Mask2Former achieves the

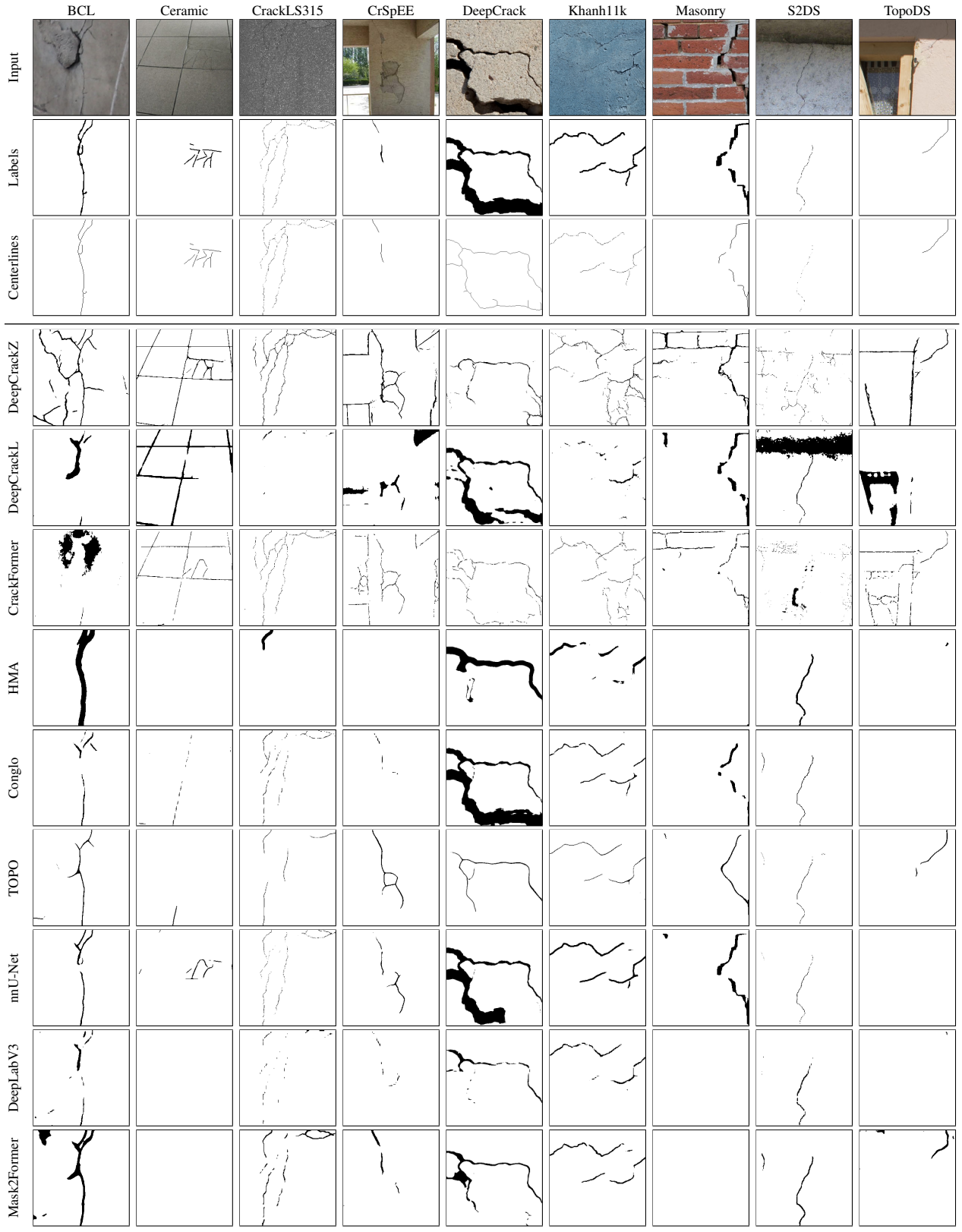

Figure 2. Qualitative results of benchmarking approaches on representative images from selected test subsets.

	Asphalt	Ceramic	Concrete	Steel	Stone	'In the Wild'
DeepCrackZ	68.6	19.2	4.9	40.9	31.7	7.1
DeepCrackL	7.3	19.7	8.0	22.9	11.5	4.0
CrackFormer	31.1	18.1	3.3	15.9	13.8	5.0
HMA	29.9	16.6	78.0	44.6	8.3	16.5
Conglo	64.9	29.1	46.8	56.5	43.2	15.8
TOPO	42.9	45.5	44.0	53.4	45.9	29.8
nnU-Net	76.5	50.8	73.4	82.3	69.2	12.8
DeepLabV3	43.4	20.4	46.2	55.5	46.5	21.1
UPerNet	54.4	21.9	47.0	63.8	49.3	20.5
Mask2Former	69.5	52.4	56.2	61.4	62.5	26.4
Average	*48.8*	*29.4*	*40.8*	*49.7*	*38.2*	*15.9*
Max	*76.5*	*52.4*	*78.0*	*82.3*	*69.2*	*29.8*

Table 6. Performance on different surface materials resp. 'in the wild' measured in $\text{clIoU}_{4\text{px}}$ in % with tolerance radius of $\tau = 4\text{px}$ on the test set.

second-best result of 53%, lagging more than 10% points behind nnU-Net. HMA and TOPO perform the best on their datasets, S2DS and TopoDS, respectively. Mask2Former outperforms nnU-Net on the Ceramic dataset by a small margin. Conglo, a finetuned DeepLabV3+, achieves top performance on DeepCrack and Khanh11k. LCW, CrSpEE, and TopoDS are the most challenging datasets across all approaches, while Khanh11k and CRKWH100 are the easiest.

Fig. 2 provides a comparative overview of qualitative results for all approaches on representative images of selected subsets. BEiT was excluded due to space constraints and limited informative value. DeepCrackZ is very sensitive to fine, line-like features, performing best on CrackLS315 but oversegmenting other images. DeepCrackL shows a strong response to darker features, such as the shadow in S2DS. CrackFormer tends to oversegmentation. HMA produces thick masks and misses finer cracks, especially in unknown materials. Conglo shows a balanced result but struggles with Ceramic and 'in the wild' data. TOPO produces connected cracks while missing finer cracks in Ceramic and CrackLS315. DeepLabV3 underperforms for thin cracks, on Masonry, and 'in the wild'. The performance of nnU-Net appears relatively balanced, partially segmenting even the small Ceramic cracks. However, nnU-Net fails for TopoDS. Mask2Former misses thin cracks and struggles with Ceramic and Masonry. Many approaches have problems segmenting the thick crack in DeepCrack, for which only DeepCrackL, Conglo, and nnU-Net show (partial) success.

5.3. Surface Dependency

To assess performance under diverse conditions, Tab. 6 reports $\text{clIoU}_{4\text{px}}$ on five different surface materials as well as 'in the wild'. *Asphalt* is represented by AEL, CFD, CrackLS315, CRKWH100, and GAPS384, *ceramic* by Ceramic, *Concrete* by S2DS and UAV75, *steel* by BCL, and *stone* by the Stone331 dataset. CrSpEE and TopoDS represent 'in the wild' conditions. Observations show that the models perform reasonably well on the materials they were trained on: HMA achieves the top $\text{clIoU}_{4\text{px}}$ on concrete, and TOPO excels in 'in the wild'. For asphalt, steel, and stone, however, nnU-Net takes the lead. Mask2Former outperforms nnU-Net on ceramic by a small margin. Despite nnU-Net's overall strong performance, it distinctly lags behind in 'in the wild'; all other general-purpose models exceed nnU-Net by at least 7

In all approaches, 'in the wild' proves to be the toughest scenario. Steel appears to be the easiest, although this can be attributed to the relative ease of the BCL dataset. The average performance on asphalt is also decent, likely because many crack-specific approaches were trained on asphalt data.

6. Conclusion

The decentralized emergence of datasets for crack segmentation presents a considerable challenge in tracking methodical progress in the field. The compilation of relevant datasets into the OMNICRACK30K dataset marks the first systematic and comprehensive approach to facilitate sustainable benchmarking for crack segmentation. The analyses reveal significant overlap among available datasets, sometimes even resulting in information leakage from the training set into the test set. These shortcomings are addressed to establish a scientifically sound basis for benchmarking. Emphasis is placed on preserving the original datasets as unmodified as possible to maintain comparability for past and future evaluations on the individual datasets.

Beyond crack-specific approaches developed in recent years, transfer learning is applied to SOTA models designed for semantic segmentation. Despite the promotion of tailored architectures and losses in crack-specific approaches, general-purpose models demonstrate effective performance in crack segmentation. In some cases, they even outperform dedicated crack-specific models, highlighting the effectiveness of these general-purpose models.

The standout model is nnU-Net, relying on the basic U-Net architecture. The self-configuration of architectural, data, and training parameters indicates that architectural modifications are of subordinate relevance. The performance on OMNICRACK30K positions nnU-Net as a powerful starting point for universal crack segmentation. While past approaches often focused on singular datasets, the field of crack segmentation has matured, prompting the logical next step toward more challenging environments and diverse surfaces. Although nnU-Net performs well under controlled circumstances, it exhibits significant deficits in crack segmentation 'in the wild.' These deficits create opportunities for future domain-specific adaptations and genuine methodical contributions in the field of crack segmentation.

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. 1

[2] R Amhaz, S Chambon, J Idier, and V Baltazart. Automatic crack detection on 2d pavement images: An algorithm based on minimal path selection, accepted to ieee trans. *Intell. Transp. Syst*, 2015. 2

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 2

[4] Yongsheng Bai, Halil Sezen, and Alper Yilmaz. Detecting cracks and spalling automatically in extreme events by end-to-end deep learning frameworks. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2:161–168, 2021. 2, 3

[5] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 6

[6] Christian Benz and Volker Rodehorst. Image-based detection of structural defects using hierarchical multi-scale attention. In *Pattern Recognition: 44th DAGM German Conference, DAGM GCPR 2022, Konstanz, Germany, September 27–30, 2022, Proceedings*, pages 337–353. Springer, 2022. 2, 4, 5, 6

[7] Christian Benz, Paul Debus, Huy Khanh Ha, and Volker Rodehorst. Crack segmentation on uas-based imagery using transfer learning. In *2019 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pages 1–6. IEEE, 2019. 2, 4

[8] Eric Bianchi and Matthew Hebdon. Development of extendable open-source structural inspection datasets. *Journal of Computing in Civil Engineering*, 36(6):04022039, 2022. 2, 3, 4, 5

[9] Eric Bianchi and Matthew Hebdon. Visual structural inspection datasets. *Automation in Construction*, 139:104299, 2022. 2

[10] Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3d: A large benchmark and model for 3d object detection in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13154–13164, 2023. 4

[11] Young-Jin Cha, Wooram Choi, and Oral Büyüköztürk. Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, 32(5):361–378, 2017. 2

[12] Fu-Chen Chen and Mohammad R Jahanshahi. Nb-cnn: Deep learning-based crack detection using convolutional neural network and naïve bayes data fusion. *IEEE Transactions on Industrial Electronics*, 65(5):4392–4400, 2017. 2

[13] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 6

[14] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 2

[15] Zhuangzhuang Chen, Jin Zhang, Zhuonan Lai, Guanming Zhu, Zun Liu, Jie Chen, and Jianqiang Li. The devil is in the crack orientation: A new perspective for crack detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6653–6663, 2023. 2

[16] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 6

[17] 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, 4

[18] Sattar Dorafshan, Robert J Thomas, and Marc Maguire. Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Construction and Building Materials*, 186:1031–1045, 2018. 1

[19] 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. 2

[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 *2017 international joint conference on neural networks (IJCNN)*, pages 2039–2047. IEEE, 2017. 3

[21] Zicheng Guo and Richard W Hall. Parallel thinning with two-subiteration algorithms. *Communications of the ACM*, 32(3):359–373, 1989. 5

[22] Huy Khanh Ha. Crack segmentation, 2019. 2

[23] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence*, 35(6):1397–1409, 2012. 2

[24] Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021. 5

[25] Gerivan Santos Junior, Janderson Ferreira, Cristian Millán-Arias, Ramiro Daniel, Alberto Casado Junior, and Bruno JT Fernandes. Ceramic cracks segmentation with deep learning. *Applied Sciences*, 11(13):6017, 2021. 2, 3

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 2

[27] Shreyas Kulkarni, Shreyas Singh, Dhananjay Balakrishnan, Siddharth Sharma, Saipraneeth Devunuri, and Sai Chowdeswara Rao Korlapati. Crackseg9k: A collection and benchmark for crack segmentation datasets and frameworks. *arXiv preprint arXiv:2208.13054*, 2022. 2, 3, 5

[28] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial intelligence and statistics*, pages 562–570. Pmlr, 2015. 2

[29] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2

[30] Romulo Gonçalves Lins and Sidney N Givigi. Automatic crack detection and measurement based on image analysis. *IEEE Transactions on Instrumentation and Measurement*, 65 (3):583–590, 2016. 1

[31] Huajun Liu, Xiangyu Miao, Christoph Mertz, Chengzhong Xu, and Hui Kong. Crackformer: Transformer network for fine-grained crack detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3783–3792, 2021. 2, 5

[32] 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. 2, 3, 5

[33] Zhenqing Liu, Yiwen Cao, Yize Wang, and Wei Wang. Computer vision-based concrete crack detection using u-net fully convolutional networks. *Automation in Construction*, 104: 129–139, 2019. 2

[34] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2

[35] Lakshay Middha. Crack segmentation dataset, 2020. 2

[36] Arun Mohan and Sumathi Poobal. Crack detection using image processing: A critical review and analysis. *Alexandria Engineering Journal*, 57(2):787–798, 2018. 1

[37] Henrique Oliveira and Paulo Lobato Correia. Automatic road crack detection and characterization. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):155–168, 2012. 1

[38] Bryan G Pantoja-Rosero, D Oner, Mateusz Kozinski, Radhakrishna Achanta, Pascal Fua, Fernando Pérez-Cruz, and K Beyer. Topo-loss for continuity-preserving crack detection using deep learning. *Construction and Building Materials*, 344:128264, 2022. 2, 4, 5

[39] Amir Rezaie, Radhakrishna Achanta, Michele Godio, and Katrin Beyer. Comparison of crack segmentation using digital image correlation measurements and deep learning. *Construction and Building Materials*, 261:120474, 2020. 2, 3

[40] 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. 2, 5

[41] Muhammad Salman, Senthan Mathavan, Khurram Kamal, and Mujib Rahman. Pavement crack detection using the gabor filter. In *16th international IEEE conference on intelligent transportation systems (ITSC 2013)*, pages 2039–2044. IEEE, 2013. 1

[42] 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. 1, 2, 3

[43] 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. 5

[44] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[45] Sunil K Sinha and Paul W Fieguth. Automated detection of cracks in buried concrete pipe images. *Automation in construction*, 15(1):58–72, 2006. 1

[46] Ronny Stricker, Markus Eisenbach, Maximilian Sesselmann, Klaus Debes, and Horst-Michael Gross. Improving visual road condition assessment by extensive experiments on the extended gaps dataset. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019. 3

[47] Ronny Stricker, Dustin Aganian, Maximilian Sesselmann, Daniel Seichter, Marius Engelhardt, Roland Spielhofer, Matthias Hahn, Astrid Hautz, Klaus Debes, and Horst-Michael Gross. Road surface segmentation-pixel-perfect distress and object detection for road assessment. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pages 1789–1796. IEEE, 2021. 3

[48] Ahmed Mahgoub Ahmed Talab, Zhangcan Huang, Fan Xi, and Liu HaiMing. Detection crack in image using otsu method and multiple filtering in image processing techniques. *Optik*, 127(3):1030–1033, 2016. 1

[49] Andrew Tao, Karan Sapra, and Bryan Catanzaro. Hierarchical multi-scale attention for semantic segmentation. *arXiv preprint arXiv:2005.10821*, 2020. 2

[50] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018. 6

[51] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015. 2

[52] Fan Yang, Lei Zhang, Sijia Yu, Danil Prokhorov, Xue Mei, and Haibin Ling. Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, 21(4):1525–1535, 2019. 2, 3

[53] Liang Yang, Bing Li, Wei Li, Zhaoming Liu, Guoyong Yang, and Jizhong Xiao. Deep concrete inspection using unmanned aerial vehicle towards cssc database. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, pages 24–28, 2017. 2, 3

[54] Xincong Yang, Heng Li, Yantao Yu, Xiaochun Luo, Ting Huang, and Xu Yang. Automatic pixel-level crack detection and measurement using fully convolutional network. *Computer-Aided Civil and Infrastructure Engineering*, 33 (12):1090–1109, 2018. 2

[55] Xiao-Wei Ye, T Jin, ZX Li, SY Ma, Y Ding, and YH Ou. Structural crack detection from benchmark data sets using pruned fully convolutional networks. *Journal of Structural Engineering*, 147(11):04721008, 2021. 2

[56] Chul Min Yeum and Shirley J Dyke. Vision-based automated crack detection for bridge inspection. *Computer-Aided Civil and Infrastructure Engineering*, 30(10):759–770, 2015. 1

[57] Lei Zhang, Fan Yang, Yimin Daniel Zhang, and Ying Julie Zhu. Road crack detection using deep convolutional neural network. In *2016 IEEE international conference on image processing (ICIP)*, pages 3708–3712. IEEE, 2016. 2, 3

[58] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1529–1537, 2015. 2

[59] 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. 1, 2, 3

[60] 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, 2018. 2, 3, 4, 5