DOI: 10.37943/WAGZ5973

Maksat Galiyev
PhD Candidate, Senior Lecturer, Department of Computer Science
galiev.maksat@gmail.com, orcid.org/0009-0006-2045-7907
SDU University, Kazakhstan
Timur Merembayev
PhD, Assistant Professor, Laboratory of Artificial Intelligence and Robotics
timur.merembayev@gmail.com, orcid.org/0000-0001-8185-235X
Institute of Information and Computational Technologies, Kazakhstan
Olzhas Toktassyn
PhD Student, Department of Machine Learning and Data Science
olzhastoktassyn@gmail.com , orcid.org/0009-0004-7226-650X
Satbayev University, Kazakhstan
Nurzhan Sagyndyk
PhD Student, Department of Machine Learning and Data Science
Nurzhan.sagyndyk99@gmail.com, orcid.org/0009-0008-4697-8844
Satbayev University, Kazakhstan

COMPARATIVE EVALUATION OF SEMANTIC SEGMENTATION MODELS FOR CRACK DETECTION

Abstract: Cracks on concrete and asphalt surfaces are an early warning that a dam, a bridge, or a road is starting to fail, and missing them can be expensive. Inspection is increasingly done with drones, which produce far more images than an engineer can check by hand, so the analysis has to be automated. In this paper we compare five semantic segmentation networks for crack detection and, unlike most such comparisons, we push all of them through exactly the same pipeline. Four are convolutional (U-Net, UNet++, DeepLabV3+, and a Feature Pyramid Network) and one is a transformer (SegFormer-B2). Every model sees the same data, the same 320 x 320 inputs, the same augmentation, the same Binary Cross-Entropy plus Dice loss, and the same 0.5 threshold, and all are trained with mixed precision on the public Crack500 benchmark and its official train, validation, and test split. We repeat each training five times with different seeds, report the mean and standard deviation on the test set, and check the gap between the two best models with a paired significance test. To see whether the ranking survives outside Crack500, we then run the trained networks, untouched, on a second dataset (DeepCrack). SegFormer-B2 was the most accurate model on Crack500 (IoU 0.595, F1 0.746), and the t-test shows its margin over UNet++ is well beyond chance. It stayed first on DeepCrack too (IoU 0.704, F1 0.826). We also report parameters, FLOPs, inference time, and memory, so the accuracy can be weighed against the cost of running each model.

Keywords: crack detection; semantic segmentation; deep learning; structural health monitoring; transformer models; SegFormer; convolutional neural networks; Crack500; cross-dataset validation; benchmark

Introduction

A crack on the surface of a dam, a bridge, or a road is often the first thing an inspector can actually see when a structure begins to fail. Small cracks grow, and one missed today can become a structural problem later, which is why early detection matters for both safety and cost [1-2]. The usual approach, sending an engineer to look, is slow and subjective, and some surfaces are simply hard to reach. Drones have changed the first half of the problem: they photograph large surfaces quickly and safely [3-4]. The second half, working through thousands of images, still has to be solved automatically.

Received: 12.06.2026 Accepted: 17.07.2026 Published: 30.06.2026

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 26, JUNE 2026

For that, deep learning is now the default. Convolutional neural networks (CNNs) learn features straight from pixels and have been used for defect detection for years [5-8]. U-Net, with its skip connections, made pixel-level segmentation practical [9], and UNet++ improved on it by adding nested, dense connections between the encoder and decoder [10]. DeepLabV3+ and Feature Pyramid Networks push the multi-scale side further [11-14]. Transformers arrived more recently: instead of local filters, SegFormer uses self-attention to relate distant parts of an image, and it does so efficiently [15].

Knowing which of these is actually best is harder than it sounds. Papers report their numbers on different datasets, at different resolutions, and with different augmentation, splits, metrics, and thresholds, so the numbers do not line up. The task is also unforgiving: cracks are thin and broken, they blend into the surrounding texture, and they cover only a sliver of the pixels, which leaves the training data badly imbalanced [5, 6, 16].

We do not introduce a new network or a new loss. What we try to do is compare the existing ones properly. All five models are trained and tested on identical data, and the operations, loss, and metrics are written out as equations so the setup is unambiguous. Three things separate this from the usual comparison paper. First, we train each model five times and run a paired significance test, so a 0.01 gap is not reported as if it were meaningful. Second, we take the trained models to a different dataset and measure how they do there, which shows whether a ranking generalises or is just tuned to one benchmark. Third, we record the computational cost of each model next to its accuracy. With all three in place, the transformer SegFormer-B2 comes out on top, significantly ahead of the CNNs on Crack500, and still ahead on the second dataset.

Literature review and problem statement

Monitoring of large structures has become its own research area, mostly because failures are expensive. The literature keeps pointing to early defect detection, and cracks in particular, as the key [1-2]. Drones help with data collection, capturing detailed images from a distance [3-4], and in doing so they move the difficulty downstream, to interpreting all that imagery reliably.

CNNs dominate that interpretation step because they learn features directly from the images [5-6]. U-Net's skip connections became the standard recipe for pixel-level segmentation [9], and UNet++ tightened the fusion across scales with nested connections [10]. Crack-specific work followed: Xia et al. added attention to DeepLabV3+ for pavement cracks [11], Zhou et al. built a multi-scale detector for bridge cracks [12], and transformer and hybrid models such as SegFormer [15] and convolution-transformer combinations [17] have appeared.

Crack500 [18] is a good example of why comparing across papers does not work. The dataset has 500 pavement photos of about 2000 x 1500 pixels, each cut into sixteen tiles for roughly 3,368 labelled crops, and different studies then feed these same crops in at 256 x 256, 448 x 448, or 360 x 640, with their own splits. Table 1 lists a few representative studies; the dataset, input size, split, metric, and threshold change from row to row, so the reported scores cannot be read side by side.

Table 1. Heterogeneity of experimental setups across representative crack-segmentation studies (as reported in the cited works)

Study	Dataset / domain	Input size	Train / val / test split	Metrics; threshold
[9] Ronneberger	ISBI EM; biomedical	512 x 512	single split, heavy augmentation	IoU, warping error; 0.5
[10] Zhou	Multiple medical sets	dataset-specific	dataset-specific	IoU, Dice; not standardized
[18] Yang (Crack500)	Crack500; pavement	~360 x 640	1896 / 348 / 1124	F1, precision, recall; 0.5
[11] Xia	Own pavement set	not standardized	own split	mIoU; not reported
[12] Zhou	Aerial concrete bridge	not standardized	own split	mAP, precision; detection-based
[16] Zhou	Pipe-gallery cracks	not standardized	own split	IoU, F1; 0.5

DOI: 10.37943/WAGZ5973

The newest work, from 2024 onward, moves toward foundation models, mask-classification transformers, and state-space networks. A cluster of papers builds on the Segment Anything Model (SAM): Owor et al. retrained only its mask decoder, using 180 images, for prompt-based pavement segmentation [19]; Ge et al. fine-tuned a vision foundation model for cracks [20]; and Ye et al. used SAM for instance-level damage detection [21]. Yang bolted a local-perception module and extra convolutions onto Mask2Former to keep the thin crack shapes that global attention loses [22]. Hybrids keep coming, such as Wang et al. and their dual-path network [23] and the Hybrid-Segmentor of Goo et al. [24], and Han et al. swapped attention for a visual Mamba backbone to save computation [25]. All of this is moving fast, but the papers rarely test against strong CNNs in one place. That head-to-head is what we provide, with SegFormer-B2 standing in for the transformer side against four established CNNs.

So the gap we address is the absence of a single controlled comparison of CNN and transformer segmentation models for cracks, one that reports variability, significance, cross-dataset behaviour, and cost together rather than one at a time.

The aim and objectives of the study

Our aim is a controlled, repeatable comparison of current segmentation networks for crack detection, run under one setup, so that the best model can be identified and its ranking checked on data it has not seen [1, 2, 5, 6].

The objectives are the following:

1) to define a unified, mathematically specified experimental framework, with a fixed data partition, preprocessing, augmentation, loss, and threshold, for training and testing crack-segmentation models on the public Crack500 benchmark [5, 6, 18];
2) to evaluate four convolutional architectures, namely U-Net, UNet++, DeepLabV3+, and a Feature Pyramid Network [9-10, 13-14];
3) to evaluate the transformer based SegFormer-B2 model on the same task [15];
4) to measure performance on an independent test set with Intersection over Union, F1-score, precision, and recall, reported as mean and standard deviation over five random seeds with a paired significance test [26-27];
5) to assess generalization through cross-dataset external validation, by evaluating the trained models on a second, unseen dataset (DeepCrack) [28];
6) to analyse the computational cost of each model (parameters, floating point operations, inference time, memory) and its detection behaviour through precision and recall; and
7) to give evidence-based guidance for selecting a segmentation architecture in automated crack-inspection systems [1, 5].

Materials and methods

This section describes the datasets, the preprocessing, the networks, the equations behind them, the training and evaluation, and the cost measurements. The point throughout was to keep one protocol and apply it to every model without change [5-6, 26]. Table A1 in the appendix lists the full configuration.

Datasets and class imbalance.

Most of the experiments use Crack500 [18], a standard benchmark with real, hand-drawn pixel masks. It holds 500 pavement images of roughly 2000 x 1500 pixels, split into sixteen tiles each, giving about 3,368 crops divided into fixed sets of 1,896 training, 348 validation, and 1,124 test crops. We kept these official sets and touched the test set only at the end. The crops are around 360 x 640 pixels and were resized as described below. In every mask, crack pixels are a small minority, the class imbalance the Dice term is there to handle. For the external test we use DeepCrack [28], an unrelated set of 537 annotated crack images with different surfaces and lighting; its 237-image test split, never used in training, is our unseen data.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 26, JUNE 2026

Preprocessing and augmentation.

Images were resized to 320 x 320 and normalised with the ImageNet mean and standard deviation, as expected for ImageNet-pretrained encoders [9-10]. Masks were resized with nearest-neighbour interpolation and thresholded to 1 for crack and 0 for background. On the training split only, we applied random horizontal and vertical flips and random 90-degree rotations. Every model got the same pipeline, so none had a data advantage.

Segmentation architectures.

Five networks were compared. U-Net is the encoder-decoder baseline with skip connections [9]. UNet++ adds nested, dense skips for better multi-scale fusion [10]. DeepLabV3+ combines atrous spatial pyramid pooling with an encoder-decoder [13]. FPN builds a feature hierarchy and was originally a detection network [14]. SegFormer-B2 uses a hierarchical Mix-Transformer (MiT-B2) encoder and a small MLP decoder, with self-attention for global context [15]. U-Net, UNet++, and FPN ran on a ResNet-34 encoder and DeepLabV3+ on ResNet-50, all ImageNet-pretrained, as was SegFormer-B2. We used segmentation-models-pytorch for the CNNs and HuggingFace Transformers for SegFormer-B2. Parameter counts are in Table 2 and Table 4.

Mathematical formulation.

The CNNs are built on the discrete two-dimensional convolution, which slides a kernel K across an input map I and reads off a response at each position (i, j):

$$(I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n) + b \tag{1}$$

with b a bias. Stacking these with non-linearities gives the feature hierarchy that U-Net, UNet++, DeepLabV3+, and FPN rely on. SegFormer-B2 works differently, through scaled dot-product self-attention, which ties every position to every other one via queries Q, keys K, and values V:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

with d_k the key dimension. This is what lets the transformer track a long crack from one side of the image to the other. Training used a loss that adds a Dice term to Binary Cross-Entropy. For N pixels with labels y_i and predictions p_i:

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i log p_i + (1-y_i) log(1-p_i)] \tag{3}$$

$$L_{Dice} = 1 - \frac{2\sum_i p_i y_i + \varepsilon}{\sum_i p_i + \sum_i y_i + \varepsilon} \tag{4}$$

$$L = L_{BCE} + L_{Dice} \tag{5}$$

where epsilon keeps the Dice term from dividing by zero. Cross-entropy handles per-pixel classification; the Dice term pushes on region overlap and, because it is not swamped by the background, it takes the pressure off the imbalance.

Training strategy.

All models used the AdamW optimizer [29] with a weight decay of 1 x 10^{-4}. The learning rate was 1 x 10^{-4} for the convolutional models and 5 x 10^{-5} for SegFormer-B2, which needs a gentler rate. Mixed precision cut memory and time. Each run went up to 30 epochs at batch size 8, with early stopping after 7 epochs of no improvement, and we kept the checkpoint with the best validation IoU. For variability we trained every model with five seeds (eight for UNet++), all on one NVIDIA RTX 4090.

DOI: 10.37943/WAGZ5973

Evaluation and statistical analysis.

Models were scored on the test set with Intersection over Union, F1-score, precision, and recall at a 0.5 threshold. With TP, FP, and FN counted per pixel:

$$IoU = \frac{TP}{TP + FP + FN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{8}$$

Intersection over Union and F1-score are the informative ones under imbalance, and we report them as mean and standard deviation across the five seeds. To check the gap between the top two models we compared their per-seed scores with a paired t-test and a Wilcoxon signed-rank test at 0.05 [27]. Generalisation was measured by running the trained models on the DeepCrack test set, with no further training, and computing the same metrics.

Computational profiling.

For each model we recorded parameters, floating point operations at 320 x 320 (via ptflops), the mean inference time per image over fifty warm forward passes, and peak GPU memory, all on the same machine, so accuracy could be read next to cost.

Results

This section covers the Crack500 results, the significance test, the DeepCrack results, the cost, and a short error analysis [5, 6, 15].

Table 2 has the Crack500 test scores as mean and standard deviation. SegFormer-B2 led on every measure, at 0.5951 IoU and 0.7462 F1, with the best precision (0.7222) and small standard deviations that point to stable training. The four CNNs sat close together: FPN, U-Net, and UNet++ landed between 0.5791 and 0.5808 IoU, and DeepLabV3+ was last at 0.5751. Figure 1 plots IoU and F1 with the standard deviation as error bars.

Table 2. Crack500 test-set performance (mean ± standard deviation over five seeds; UNet++ over eight), ordered by Intersection over Union

Model	Params (M)	IoU (mean ± std)	F1 (mean ± std)	Precision	Recall
SegFormer-B2	27.3	0.5951 ± 0.0023	0.7462 ± 0.0018	0.7222	0.7722
FPN (ResNet34)	23.2	0.5808 ± 0.0035	0.7348 ± 0.0028	0.7094	0.7630
U-Net (ResNet34)	24.4	0.5791 ± 0.0059	0.7334 ± 0.0048	0.6977	0.7731
UNet++ (ResNet34)	26.1	0.5791 ± 0.0019	0.7334 ± 0.0015	0.7128	0.7563
DeepLabV3+ (ResNet50)	26.7	0.5751 ± 0.0050	0.7302 ± 0.0040	0.6927	0.7725

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 26, JUNE 2026

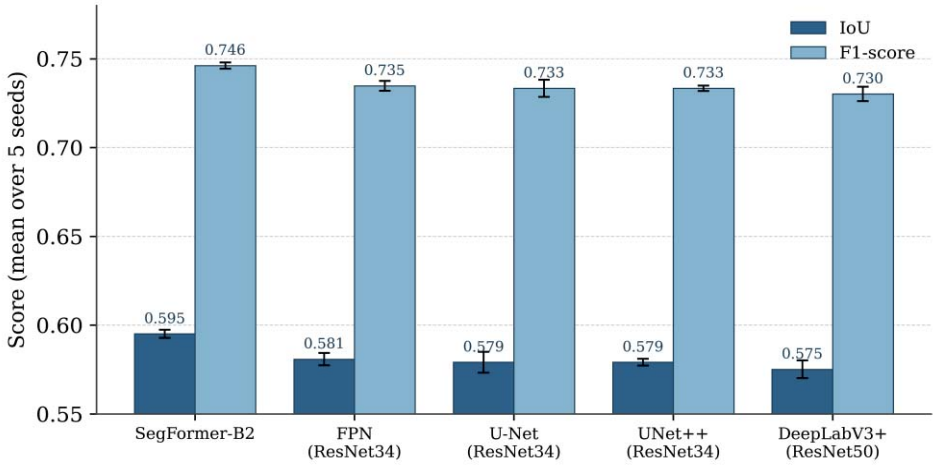

Figure 1. Crack500 test-set Intersection over Union and F1-score (mean over seeds; error bars show standard deviation).

To confirm the lead was real, we compared the per-seed test IoU of SegFormer-B2 and UNet++. The paired t-test came back strongly significant (t = 21.75, p = 2.6 x 10^-5), and the Wilcoxon test hit its lowest possible value for five pairs (p = 0.0625). With the standard deviations in Table 2 not overlapping, SegFormer-B2 is clearly ahead and not just lucky. Going from three seeds to five dropped the t-test p-value by more than two orders of magnitude, so the extra seeds sharpened the conclusion rather than muddying it.

Cross-dataset external validation.

Table 3 shows the trained models on the DeepCrack test set, with no extra training. The Crack500 order held: SegFormer-B2 again first at 0.7035 IoU and 0.8260 F1, then U-Net, and DeepLabV3+ last. The scores are higher here than on Crack500 because DeepCrack cracks are cleaner and higher in contrast; what matters is not the level but that the order is the same on two separate datasets, which suggests the advantage is in the architecture and not fitted to one benchmark. SegFormer-B2 also had the best recall on DeepCrack (0.8861), so it caught the most true crack pixels on unseen data.

Table 3. Cross-dataset external validation on the unseen DeepCrack test set (models trained on Crack500), ordered by Intersection over Union.

Model	IoU	F1	Precision	Recall
SegFormer-B2	0.7035	0.8260	0.7735	0.8861
U-Net (ResNet34)	0.6980	0.8221	0.8017	0.8436
UNet++ (ResNet34)	0.6646	0.7985	0.7670	0.8327
FPN (ResNet34)	0.6570	0.7930	0.7541	0.8360
DeepLabV3+ (ResNet50)	0.6265	0.7704	0.7874	0.7541

Table 4 lists the cost. All five models sit in a narrow 23 to 27 million parameter range, so capacity is not what separates them. Speed and memory differ more. U-Net is quickest (about 8 ms at 24.5 GFLOPs), UNet++ is the heaviest (about 37 ms at 57.7 GFLOPs) and still not the most accurate, and SegFormer-B2 gets the best accuracy at a middling cost (about 27 ms at 45.5 GFLOPs), a reasonable trade when accuracy is what you care about.

DOI: 10.37943/WAGZ5973

Table 4. Computational cost of the segmentation models at 320 x 320 input, measured on an NVIDIA RTX 4090 GPU.

Model	Params (M)	GFLOPs @320	Inference (ms/img)	Peak mem (MB)
SegFormer-B2	27.3	45.5	27.1	503
UNet++ (ResNet34)	26.1	57.7	37.5	282
DeepLabV3+ (ResNet50)	26.7	28.8	10.5	535
U-Net (ResNet34)	24.4	24.5	8.0	427
FPN (ResNet34)	23.2	21.5	10.9	624

Behavioural error analysis.

Precision and recall in Table 2 say a lot about behaviour. SegFormer-B2 combined the best precision (0.7222) with high recall (0.7722), which is why it topped IoU and F1, and it fits the idea that self-attention helps separate cracks from look-alike texture. U-Net matched it on recall (0.7731) but had the worst precision (0.6977) and the widest spread across seeds (0.0059), so it caught many crack pixels while also raising more false alarms and training less consistently. DeepLabV3+ had both the lowest precision (0.6927) and the lowest IoU, which reads as over-predicting. UNet++ was the steadiest model (0.0019) but had the lowest recall (0.7563), and FPN sat in the middle. Figure 2 shows some examples, with the input, the ground truth, and each model output; everyone finds the main crack, and the differences are at the edges and along thin branches, where SegFormer-B2 masks are cleanest.

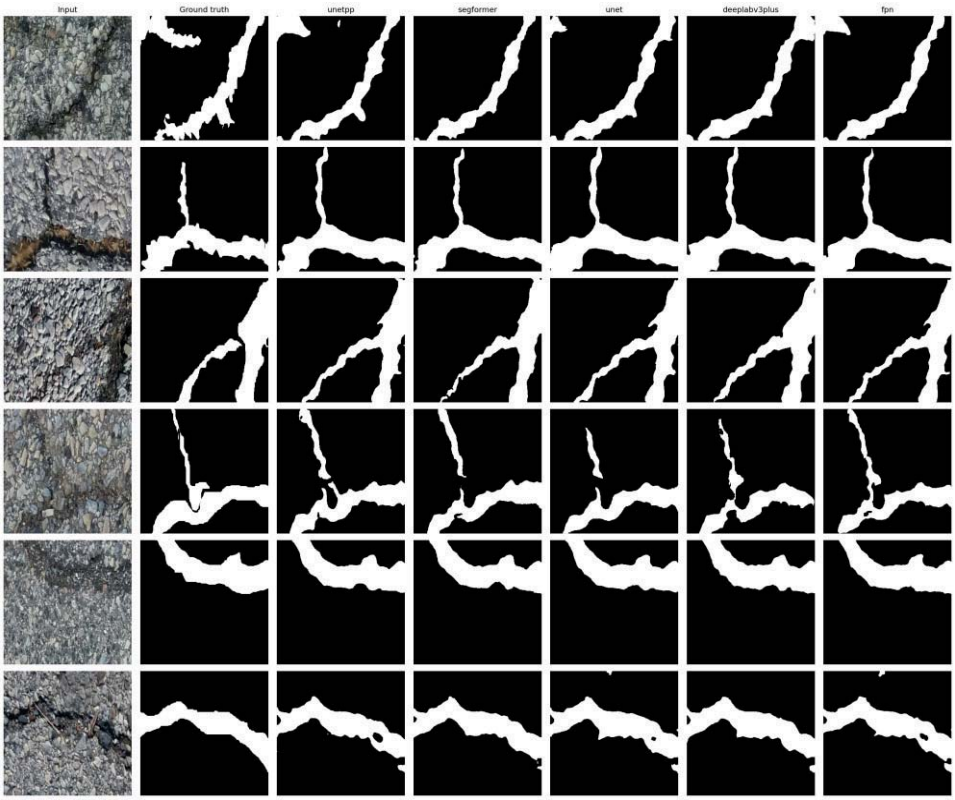

Figure 2. Qualitative results on Crack500 test crops: input, ground truth, and predictions of each model.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 26, JUNE 2026

Discussion

Under one shared protocol, SegFormer-B2 produced the most accurate segmentation, and the margin over the CNNs was significant rather than cosmetic [15]. Self-attention is the likely reason: it connects far-apart parts of the image, which suits the long, thin, broken shape of a crack. The finding runs a little against expectation, since UNet++ is often quoted as the strongest encoder-decoder, yet here it did not lead and the four CNNs bunched together. That is a reminder that the comparison protocol matters as much as the model.

The external test backs this up. Run unchanged on DeepCrack, the models kept the Crack500 order and SegFormer-B2 was again first. Leading on one benchmark can be luck or over-fitting; leading on a second, independent one is harder to dismiss. The higher DeepCrack scores just reflect easier, higher-contrast cracks, so it is the consistent order, not the absolute number, that carries the point.

On cost, since the models are close in parameters, the accuracy gaps are about design, not size. Speed and memory are where they part ways: U-Net is lean but less accurate, UNet++ is expensive for what it returns, and SegFormer-B2 buys the best accuracy at a moderate price. In a real inspection system, SegFormer-B2 makes sense when accuracy leads, and a light CNN like U-Net makes sense when speed or memory is tight.

There are limits. Both datasets are road and pavement, so while the cross-dataset test shows the models transfer between them, we still have not tried application-specific data such as UAV shots of dam surfaces, and that is next. We used one fixed 0.5 threshold; tuning it per model would move the precision-recall balance. And we compared established networks, not the newest foundation-model and mask-classification methods from the review [19-25], which would be the natural thing to add.

Future work, then, is to bring in dam-surface imagery, add the recent SAM-based, Mask2Former, and state-space models to the same benchmark, and look at convolution-transformer hybrids and crack-connectivity constraints for cleaner masks [15, 17, 23, 24].

Conclusion

We compared four CNNs (U-Net, UNet++, DeepLabV3+, FPN) against the transformer SegFormer-B2 on crack segmentation, under one protocol written out in equations and with attention to statistical and cross-dataset validity.

On Crack500, SegFormer-B2 was most accurate (IoU 0.595, F1 0.746), and the paired t-test put its lead over the best CNN well past chance. The same order held on the unseen DeepCrack set (IoU 0.704), so the advantage is not tied to one dataset. Costs showed the gaps come from architecture, not size, with SegFormer-B2 giving the best accuracy at a moderate price.

The contribution is the method: one reproducible benchmark that puts CNNs and a transformer on equal footing and reports variability, significance, generalisation, and cost together. For anyone building automated crack inspection, that turns the choice between SegFormer-B2 accuracy and a lighter CNN speed into an informed one.

Acknowledgments

This study is funded by the Committee of Science of the Ministry of Higher Education and Science of the Republic of Kazakhstan under Grant No. AP23488281, “Development of an automatic drone charging system based on Reinforcement Learning technology for continuous monitoring of large-scale territories.”

This crack-detection work feeds directly into that project: it lets imagery from visual inspection of infrastructure be processed automatically, and the networks tested here can turn that imagery into reliable crack maps.

References

[1] Albertoni, V. (2019). Structural health monitoring of concrete dams: A review and a case study (Master of Science Thesis). Politecnico di Milano, Faculty of Civil, Environmental and Land Management Engineering, Milan, Italy. https://hdl.handle.net/10589/145613

[2] Oliveira, S., Rodrigues, J., Campos Costa, A., & Mendes, P. (2026). Damage detection in concrete dams using dynamic continuous monitoring and numerical models. In Proceedings of the International Conference on Structural Health Monitoring of Intelligent Infrastructure (pp. 435–442). Taylor & Francis. https://doi.org/10.1201/9781003762553-53

[3] Qi, Y., Lin, P., Yang, G., & Liang, T. (2025). Crack detection and 3D visualization of crack distribution for UAV-based bridge inspection using efficient approaches. Structures, 78, 109075. https://doi.org/10.1016/j.istruc.2025.109075

[4] Ding, W., Yang, H., Yu, K., & Shu, J. (2023). Crack detection and quantification for concrete structures using UAV and transformer. Automation in Construction, 152, 104929. https://doi.org/10.1016/j.autcon.2023.104929

[5] Nguyen, C. K., Kawamura, K., & Nakamura, H. (2023). Deep learning-based crack detection and classification for concrete structures inspection. In Proceedings of the 17th East Asian-Pacific Conference on Structural Engineering and Construction (EASEC 2022) (pp. 710–717). Springer. https://doi.org/10.1007/978-981-19-7331-4_58

[6] Maalej, M., Cherif, R., Yaddaden, Y., & Khoumsi, A. (2022). Automatic crack detection on concrete structure using a deep convolutional neural network and transfer learning. In 2022 2nd International Conference on Advanced Electrical Engineering (ICAEE) (pp. 1–6). IEEE. https://doi.org/10.1109/ICAEE53772.2022.9962029

[7] Merembayev, T., & Amanbek, Y. (2025). Natural fracture network model using Gaussian simulation and machine learning algorithms. Applied Computing and Geosciences, 100258. https://doi.org/10.1016/j.acags.2025.100258

[8] Merembayev, T., & Amanbek, Y. (2023). Natural fracture network model using machine learning approach. In International Conference on Computational Science and Its Applications (pp. 384–397). Springer. https://doi.org/10.1007/978-3-031-37114-1_26

[9] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 234–241). https://doi.org/10.1007/978-3-319-24574-4_28

[10] Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J. (2018). UNet++: A nested U-Net architecture for medical image segmentation. In Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support (pp. 3–11). https://doi.org/10.1007/978-3-030-00889-5_1

[11] Xia, X., Su, J., Wang, Y., Liu, Y., & Li, M. (2024). Lightweight pavement crack detection model based on DeepLabv3+. Laser & Optoelectronics Progress, 61(8), 0812001. https://doi.org/10.3788/LOP231323

[12] Zhou, L., Jia, H., Jiang, S., & Xu, F. (2025). Multi-scale crack detection and quantification of concrete bridges based on aerial photography and improved object detection network. Buildings, 15(7), 1117. https://doi.org/10.3390/buildings15071117

[13] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (pp. 833–851). https://doi.org/10.1007/978-3-030-01234-2_49

[14] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 936–944). https://doi.org/10.1109/CVPR.2017.106

[15] Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021). SegFormer: Simple and efficient design for semantic segmentation with transformers. Advances in Neural Information Processing Systems, 34, 12077–12090. https://arxiv.org/abs/2105.15203

[16] Zhou, Y., Yang, Z., Bai, X., & Li, C. (2024). Semantic segmentation of surface cracks in urban comprehensive pipe galleries based on global attention. Sensors, 24(3), 1005. https://doi.org/10.3390/s24031005

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 26, JUNE 2026

[17] Noh, J. (2025). Crack segmentation using U-Net and transformer for surface defect detection. Applied Sciences, 15, 10737. https://doi.org/10.3390/app151910737

[18] Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X., & Ling, H. (2019). Feature pyramid and hierarchical boosting network for pavement crack detection. IEEE Transactions on Intelligent Transportation Systems, 21(4), 1525–1535. https://doi.org/10.1109/TITS.2019.2910595

[19] Owor, N. J., Adu-Gyamfi, Y., Aboah, A., & Amo-Boateng, M. (2025). PaveSAM – Segment anything for pavement distress. Road Materials and Pavement Design, 26(3), 593–617. https://doi.org/10.1080/14680629.2024.2374863

[20] Ge, K., Wang, C., Guo, Y. T., Tang, Y. S., Hu, Z. Z., & Chen, H. B. (2024). Fine-tuning vision foundation model for crack segmentation in civil infrastructures. Construction and Building Materials, 431, 136573. https://doi.org/10.1016/j.conbuildmat.2024.136573

[21] Ye, Z., Lovell, L., Faramarzi, A., & Ninić, J. (2024). SAM-based instance segmentation models for the automation of structural damage detection. Advanced Engineering Informatics, 62, 102826. https://doi.org/10.1016/j.aei.2024.102826

[22] Yang, Y. (2025). A transformer-based pavement crack segmentation model with local perception and auxiliary convolution layers. Electronics, 14(14), 2834. https://doi.org/10.3390/electronics14142834

[23] Wang, J., Zeng, Z., Sharma, P. K., Alfarraj, O., Tolba, A., Zhang, J., & Wang, L. (2024). Dual-path network combining CNN and transformer for pavement crack segmentation. Automation in Construction, 158, 105217. https://doi.org/10.1016/j.autcon.2023.105217

[24] Goo, J. M., Milidonis, X., Artusi, A., Boehm, J., & Ciliberto, C. (2025). Hybrid-Segmentor: A hybrid approach for automated fine-grained crack segmentation in civil infrastructure. Automation in Construction, 170, 105960. https://doi.org/10.1016/j.autcon.2024.105960

[25] Han, C., Yang, H., & Yang, Y. (2024). Enhancing pixel-level crack segmentation with visual Mamba and convolutional networks. Automation in Construction, 168, 105770. https://doi.org/10.1016/j.autcon.2024.105770

[26] Jadon, S. (2020). A survey of loss functions for semantic segmentation. In 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB) (pp. 1–7). IEEE. https://doi.org/10.1109/CIBCB48159.2020.9277638

[27] Benavoli, A., Corani, G., Demšar, J., & Zaffalon, M. (2017). Time for a change: A tutorial for comparing multiple classifiers through Bayesian analysis. Journal of Machine Learning Research, 18(77). https://www.jmlr.org/papers/v18/16-305.html

[28] Liu, Y., Yao, J., Lu, X., Xie, R., & Li, L. (2019). DeepCrack: A deep hierarchical feature learning architecture for crack segmentation. Neurocomputing, 338, 139–153. https://doi.org/10.1016/j.neucom.2019.01.036

[29] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1412.6980