MSC 68T07, 68T45, 68U10 **DOI: 10.14529/jcem260205**

ARTIFICIAL INTELLIGENCE TECHNOLOGIES FOR CRACK DETECTION IN CULTURAL HERITAGE OBJECTS

T. V. Karpeta[1], eroshkinatv@susu.ru,
E. V. Kazantseva[1], kazantsevaliza04@gmail.com
[1]South Ural State University, Chelyabinsk, Russian Federation

Driven by the need for systematic structural health monitoring of cultural heritage sites to ensure timely defect identification, this study employs neural network-based computer vision techniques. We details the development and annotation of a dataset comprising images of cracks in historical buildings. Particular emphasis is placed on data acquisition and preprocessing, as these stages significantly influence subsequent model performance metrics. The training results for both models are presented, followed by a comparative analysis. Finally, a cascaded pipeline integrating detection and segmentation is proposed to enhance the accuracy and reliability of defect identification.

Keywords: neural networks; object detection; image segmentation; cultural heritage objects; cracks; YOLOv11; U-Net.

Introduction

The preservation of cultural heritage objects (CHOs) is a priority task of modern society. Historic buildings and architectural monuments constitute an irreplaceable resource, the loss of which causes irreparable damage to cultural identity. One of the most common and hazardous types of deterioration is cracking, which occurs due to natural material aging, atmospheric exposure, temperature fluctuations, and foundation deformations.

Traditionally, technical condition assessment is performed by experts through visual inspection and manual analysis of photographs. This approach is labor-intensive, time-consuming, subject to evaluator bias, and difficult to scale. Therefore, there is an increasing demand for automated image analysis tools capable of detecting defects with high accuracy and minimal human involvement.

The rapid development of deep learning methods has opened fundamentally new possibilities for visual data analysis. Computer vision techniques based on convolutional neural networks (CNNs) have demonstrated high efficiency in classification, object detection, and image segmentation tasks [1].

1. Problem Statement

The objective of this study is the automatic detection of cracks in images of cultural heritage objects under two formulations: *object detection* and *semantic segmentation.*

Object Detection. The task is to determine the presence of a crack in an image and localize it using a bounding box. Formally,

$$f_{\text{det}} : X \to \{(b_i, c_i, p_i)\}_{i=1}^{N},$$

where $X \in \mathbb{R}^{H\times W\times 3}$ is the input image, H and W are its pixel dimensions, and 3 is the number of color channels (RGB); $b_i = (x_{\text{center}}, y_{\text{center}}, w, h)$ defines the bounding box

parameters (center coordinates, width, and height); $c_i \in \{\text{crack}, \text{web\_crack}\}$ indicates the defect category, distinguishing between 'crack' (a single longitudinal or transverse surface damage) and 'web_crack' (a complex network of intersecting cracks); $p_i \in [0, 1]$ denotes the confidence level of the prediction.

Semantic Segmentation. The task is to delineate crack regions at the pixel level:

$$f_{\text{seg}} : X \to Y,$$

where $X \in \mathbb{R}^{H \times W \times 3}$ is the input image; $Y \in \{0, 1, 2\}^{H \times W}$ is the semantic segmentation mask. Each pixel value in Y corresponds to a specific category: **0** represents the background (areas without defects); **1** (crack) denotes a single longitudinal or transverse surface breach; **2** (web_crack) signifies a complex network of multiple intersecting cracks. The separation into two defect classes is motivated by their different physical characteristics and restoration requirements.

2. Dataset Construction

A specialized dataset of cultural heritage object images containing crack defects was constructed for training the neural networks. The development of a custom dataset was necessary because existing public datasets mainly contain images of standard construction elements and do not reflect the specific features of cultural heritage sites.

2.1. Data Sources

The dataset was compiled from two independent sources.

1. Photographs provided by a client during professional inspections of CHOs. High-resolution images were divided into equal square fragments (Fig. 1), resulting in 3,080 images.

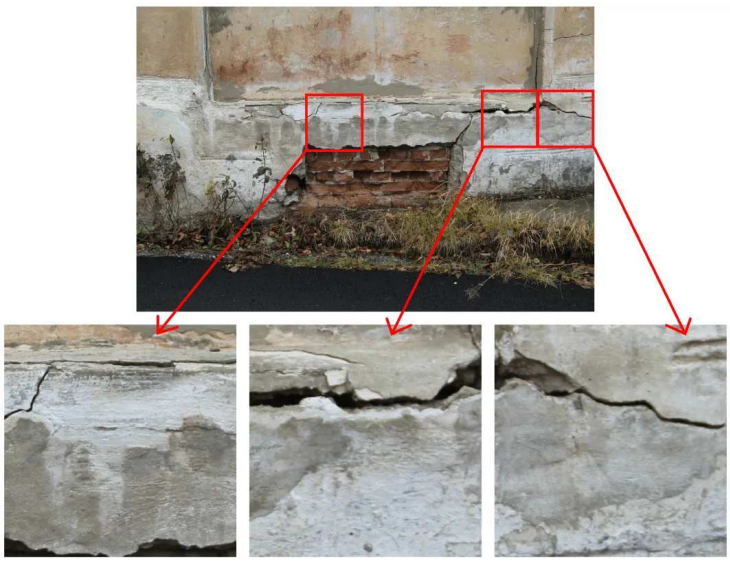

Fig. 1. An example of splitting an image into fragments

2. Photographs captured by the authors during field surveys of historical and residential buildings (5,858 images), representing diverse lighting conditions, distances, surface types, and viewpoints (Fig. 2).

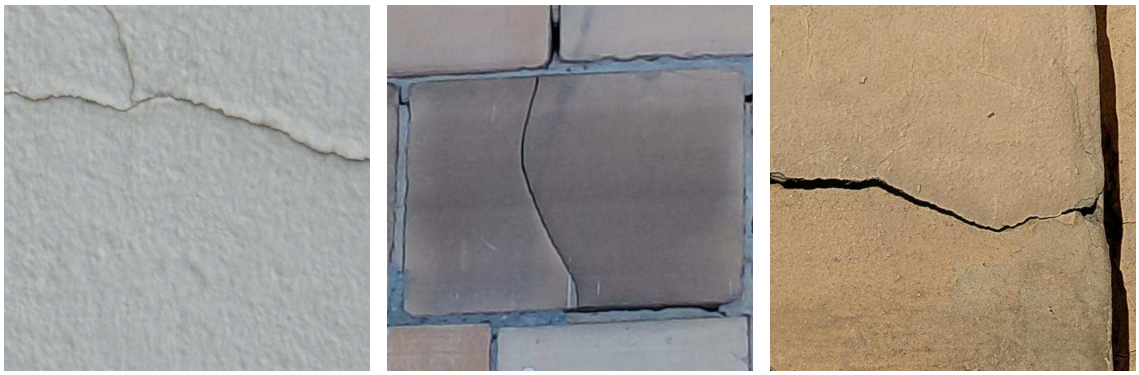

Fig. 2. An example of self-shooting

2.2. Data Annotation

The layout of the dataset was performed manually using the CVAT (Computer Vision Annotation Tool), an open platform for annotating visual data. The markup was carried out in two formats corresponding to the subtasks being solved.

Detection annotation was performed in YOLO (You Only Look Once) format. Each crack was enclosed by a minimal axis-aligned bounding box (Fig. 3). Each image corresponds to a text file in the format:

$$(\text{class}, x_{\text{center}}, y_{\text{center}}, \text{width}, \text{height}),$$

where class index 0 corresponds to crack and 1 to web_crack.

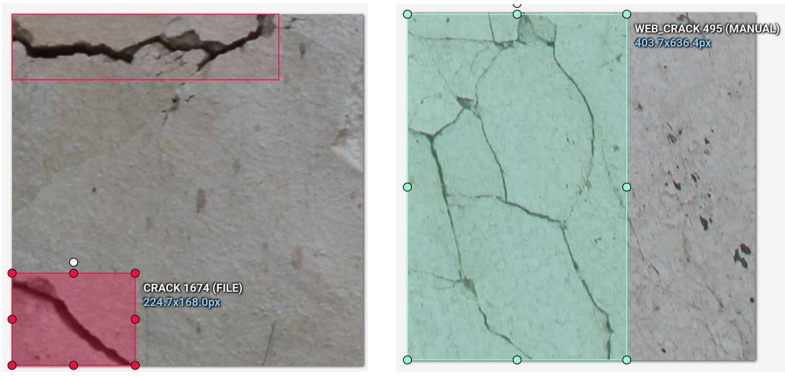

Fig. 3. Detection annotation: left — class crack, right — class web_crack

Segmentation annotation was performed as pixel-wise masks (0 — background, 1 — crack, 2 — web_crack). Polygon annotations were converted to raster masks (Fig. 4). Pixel-level annotation was the most labor-intensive stage due to the complex irregular shapes, thin branches, and blurred boundaries of cracks.

Fig. 4. Segmentation annotation: left — class crack, right — class web_crack

2.3. Data Preprocessing and Augmentation

To ensure compatibility with convolutional neural network architectures, all images in the dataset were resized to a fixed resolution of 640×640 pixels. Resizing was performed with preservation of the original aspect ratio. Any remaining areas were padded with a neutral background to obtain a square format, thereby preventing geometric distortions of crack shapes.

To artificially increase the size of the training set and improve the generalization capability of the models, data augmentation was applied. A geometric augmentation procedure was implemented, including rotations by angles that are multiples of 45°, horizontal and vertical flips, as well as their combinations. This approach enhances the model's invariance to object orientation within the scene and reduces the risk of overfitting. As a result of augmentation, the final dataset comprised **10 200 images**.

2.4. Dataset Split

The constructed dataset was divided into three disjoint samples (Table 1).

Table 1

Dataset split

Subset	**Purpose**	**Proportion**
Training	Model parameter optimization	70%
Validation	Hyperparameter tuning	15%
Test	Final performance evaluation	15%

The split was performed using a stratified sampling strategy to preserve class balance, while also ensuring diversity in surface types and imaging conditions within each subset.

3. Training Results

3.1. YOLOv11 Detection Model

For the crack detection task, the YOLOv11 architecture was employed – a single-stage object detector from the YOLO family. YOLOv11 predicts bounding boxes and object classes in a single forward pass of the image through the neural network, ensuring high processing speed while maintaining strong accuracy.

Model configuration. The YOLOv11n variant was used with an input resolution of 640×640×3. The model generates predictions at three scale levels (80×80, 40×40, 20×20), enabling the detection of cracks of various sizes, ranging from fine hairline fractures to large extended defects.

Training parameters. The model was trained for 100 epochs with a mini-batch size of 16 and a learning rate of 5×10^{-4}. Stochastic Gradient Descent with momentum (SGD with Momentum) was employed as the optimizer. The weight update rule for θ is defined as follows:

$$v_{t+1} = \mu v_t + \nabla_\theta J(\theta_t), \quad \theta_{t+1} = \theta_t - \eta v_{t+1},$$

where v denotes the velocity vector, $\mu = 0.937$ is the momentum coefficient, and $\nabla_\theta J(\theta)$ represents the gradient of the loss function.

Training was performed using a composite loss function typical of recent YOLO architectures:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{CIoU}} + \lambda_2 \mathcal{L}_{\text{BCE}} + \lambda_3 \mathcal{L}_{\text{DFL}},$$

The loss components are defined as follows:

1. **CIoU Loss** (Complete Intersection over Union) for bounding box regression:

$$\mathcal{L}_{\text{CIoU}} = 1 - \text{IoU} + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v,$$

where $\rho(\cdot)$ denotes the Euclidean distance between the centers of the predicted and ground-truth boxes, c is the diagonal length of the smallest enclosing box covering both regions, and αv represents a term accounting for the aspect ratio consistency.

2. **Binary Cross-Entropy (BCE)** for object classification:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{n} \sum_{i=1}^{n} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)],$$

where y_i denotes the ground-truth class label and $\hat{y}_i$represents the predicted probability.

3. **DFL** (Distribution Focal Loss) for refining bounding box boundaries under conditions of indistinct edges:

$$\mathcal{L}_{\text{DFL}}(P_i, P_{i+1}) = -((y_{i+1} - y) \log(P_i) + (y - y_i) \log(P_{i+1})),$$

where y s the ground-truth distance from the center to the box boundary, while y_i и y_{i+1} are the nearest integer values such that $(y_i \leq y \leq y_{i+1})$. The terms P_i и P_{i+1} correspond to the probabilities predicted by the model for these respective values.

To assess the performance of the model, standard detection metrics were employed, based on the following fundamental definitions:

- **TP** (True Positives): the number of correctly identified and classified defects where the Intersection over Union (IoU) value exceeds a predefined threshold.

- **FP** (False Positives): instances where the model incorrectly predicts the presence of a defect in a background region, or where the overlap between the predicted bounding box and the ground truth is insufficient (below the threshold).
- **FN** (False Negatives): the number of ground-truth defects that the model failed to detect.

Based on these definitions, the following primary performance metrics are computed.

- **Precision**: $P = \frac{TP}{TP+FP}$ — the proportion of correctly predicted defects out of all detections predicted by the model.
- **Recall**: $R = \frac{TP}{TP+FN}$ — the proportion of identified defects out of the total number of actual defects present in the images.
- **mAP@0.5** (Mean Average Precision): he mean precision calculated at an Intersection over Union (IoU) threshold of 0.5. This metric is defined as the arithmetic mean of the Area Under the Precision-Recall (PR) Curve across all classes:

$$mAP = \frac{1}{C}\sum_{i=1}^{C} AP_i$$

where $C$is the number of classes and AP_i denotes the Average Precision for the i-th class.

Training results. Analysis of the loss curves confirmed stable convergence and the absence of overfitting. All components of the loss function decreased monotonically throughout training, and the gap between the training and validation curves remained minimal (Fig. 5).

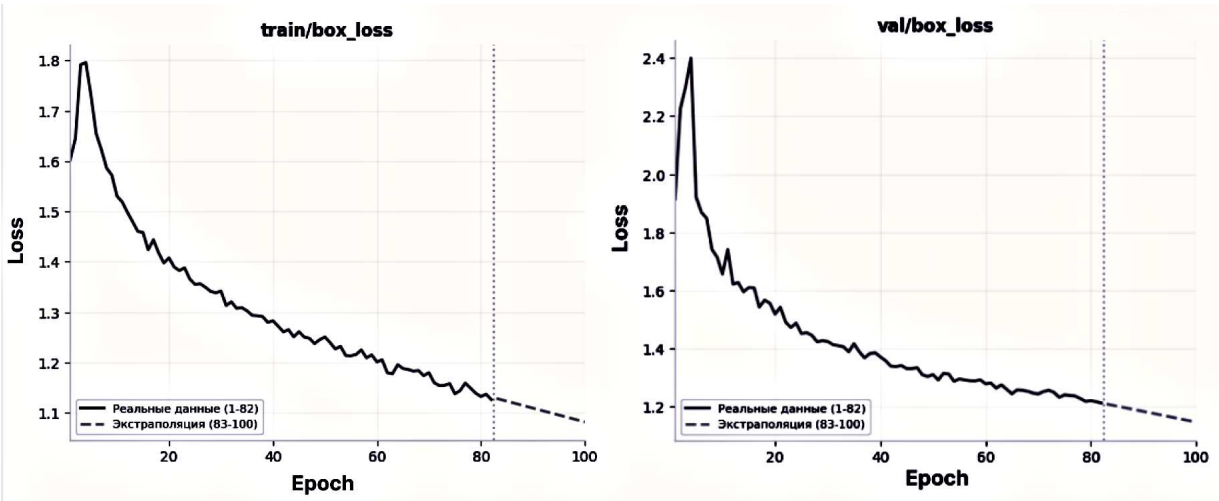

Fig. 5. Graphs of the YOLO loss function

Performance metrics on the test set are presented in Table 2.
The obtained results indicate the following:

Table 2

YOLOv11 model quality metrics based on a test sample

Class	Precision	Recall	mAP@0.5
crack	0.88	0.91	0.89
web_crack	0.82	0.84	0.83
Average	**0.85**	**0.88**	**0.86**

- the value of mAP@0.5 = 0.86 exceeds the target threshold of 0.70, confirming that the required level of performance has been achieved;
- the model demonstrates high Recall values, which is critically important for cultural heritage monitoring tasks, as missed defects may result in delayed restoration measures;
- the Precision values indicate that the model rarely misclassifies intact surface areas as cracks, reflecting a low false positive rate.

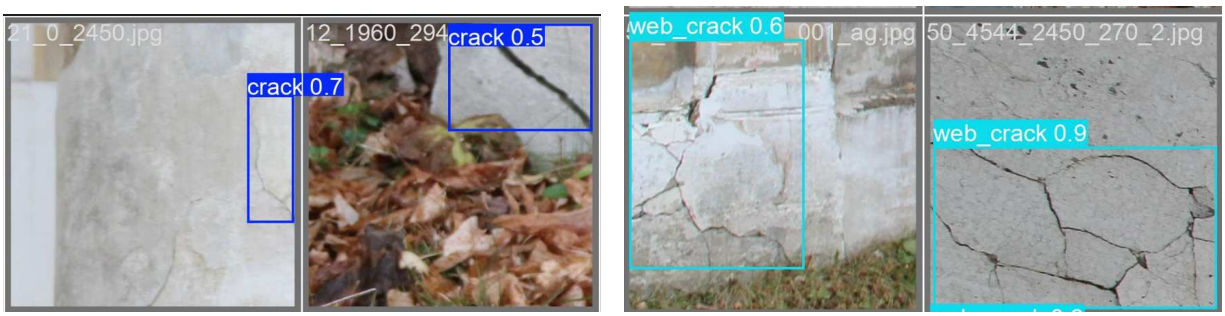

Fig. 6. Detection Model predictions: left — class crack, right — class web_crack

3.2. Training of the U-Net Segmentation Model

For the semantic segmentation of cracks, the U-Net architecture was employed. U-Net has a U-shaped structure consisting of two symmetrical parts: an encoder and a decoder, connected by skip connections. These skip connections transfer fine-grained spatial information from the encoder to the decoder, which is critically important for the accurate segmentation of thin linear crack structures.

Model configuration. The model takes as input an image of size $640 \times 640 \times 3$ and produces a prediction tensor of the same spatial resolution, $640 \times 640 \times 3$, where each pixel contains class membership probabilities for three classes (background, crack, web_crack). The encoder consists of four levels of convolutional blocks with channel dimensions increasing as $64 \to 128 \to 256 \to 512$. The decoder restores spatial resolution using transposed convolution operations.

Training parameters. The model was trained for 100 epochs with a mini-batch size of 16 and a learning rate of 5×10^{-4}. The Adam (Adaptive Moment Estimation) algorithm

was employed as the optimizer. It adaptively adjusts the learning rate for each individual parameter by utilizing estimates of the first and second moments of the gradients:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \quad v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2,$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t,$$

where g_t is the gradient of the loss function, m_t and v_t enote the estimates of the mean (the first moment) and the uncentered variance (the second raw moment) of the gradients, respectively, with $\beta_1 = 0.9, \beta_2 = 0.999$.

Given the significant class imbalance, where background pixels vastly outnumber crack pixels, a composite loss function was employed:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{CCE}} + \lambda_2 \mathcal{L}_{\text{Dice}},$$

The individual components are defined as follows:

1. **Categorical Cross-Entropy (CCE)** – the standard pixel-wise cross-entropy for multiclass classification:

$$\mathcal{L}_{\text{CCE}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=0}^{2} y_{i,c} \log(\hat{y}_{i,c})$$

   where N denotes the total number of pixels, and $y_{i,c}$ is a binary indicator variable signifying whether pixel i belongs to class c.

2. **Dice Loss** based on the Sorensen–Dice coefficient, this loss measures the similarity between the predicted mask and the ground truth:

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2 \sum_{c=1}^{2} \sum_{i=1}^{N} y_{i,c} \hat{y}_{i,c} + \varepsilon}{\sum_{c=1}^{2} \sum_{i=1}^{N} (y_{i,c} + \hat{y}_{i,c}) + \varepsilon}$$

   The use of Dice Loss helps mitigate the impact of class imbalance by focusing on the intersection between predicted and ground-truth defect pixels, effectively disregarding the predominant background pixels.

To evaluate the pixel-wise segmentation performance, the following metrics were employed, calculated based on the fundamental counts of true positive TP_c false positive FP_c, and false negative FN_c pixels for each class $c \in \{1, 2\}$:

- **Precision** measures the proportion of pixels correctly assigned to a defect class relative to the total number of pixels classified by the model as belonging to that class:

$$P_c = \frac{TP_c}{TP_c + FP_c}$$

- **Recall** represents the proportion of correctly identified defect pixels relative to the total number of ground-truth pixels for that defect category:

$$R_c = \frac{TP_c}{TP_c + FN_c}$$

- **IoU** (Intersection over Union) per class:

$$IoU_c = \frac{TP_c}{TP_c + FP_c + FN_c}$$

- **mIoU** (Mean Intersection over Union): the average segmentation quality across all target defect classes, characterizing the extent of spatial overlap between the predicted regions and the ground-truth masks:

$$mIoU = \frac{1}{C}\sum_{c=1}^{C} IoU_c$$

where $C = 2$ denotes the number of defect classes (crack and web_crack).

Training results. Both components of the loss function demonstrate stable convergence throughout the training process (Fig. 7). The Dice Loss component ensured effective optimization under conditions of significant class imbalance. The divergence between the training and validation curves is minimal, indicating the absence of critical overfitting.

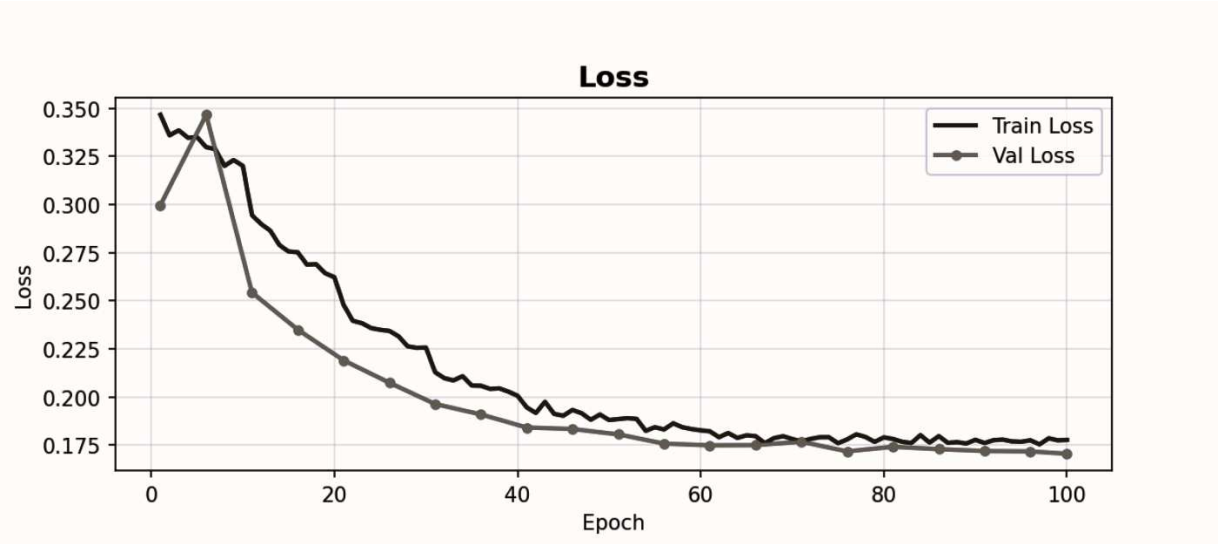

Fig. 7. Graphs of the U-Net loss function

Performance metrics on the test set are presented in Table 3.

The obtained results indicate that the mean IoU across defect classes exceeds the target threshold of 0.65, confirming the model's capability for sufficiently accurate pixel-level crack segmentation. The high Pixel Accuracy values are partially explained by the substantial predominance of background pixels; therefore, IoU and the Dice coefficient, which are more robust to class imbalance, are considered the primary evaluation metrics. The model successfully reconstructs the geometry of tortuous linear cracks as well as regions of web-like damage.

Table 3

U-Net model quality metrics based on the test sample

Class	IoU	Dice	Pixel Accuracy
crack	0.72	0.84	0.95
web_crack	0.68	0.81	0.93
background	0.98	0.99	0.99
mIoU	**0.79**	**0.88**	**0.96**

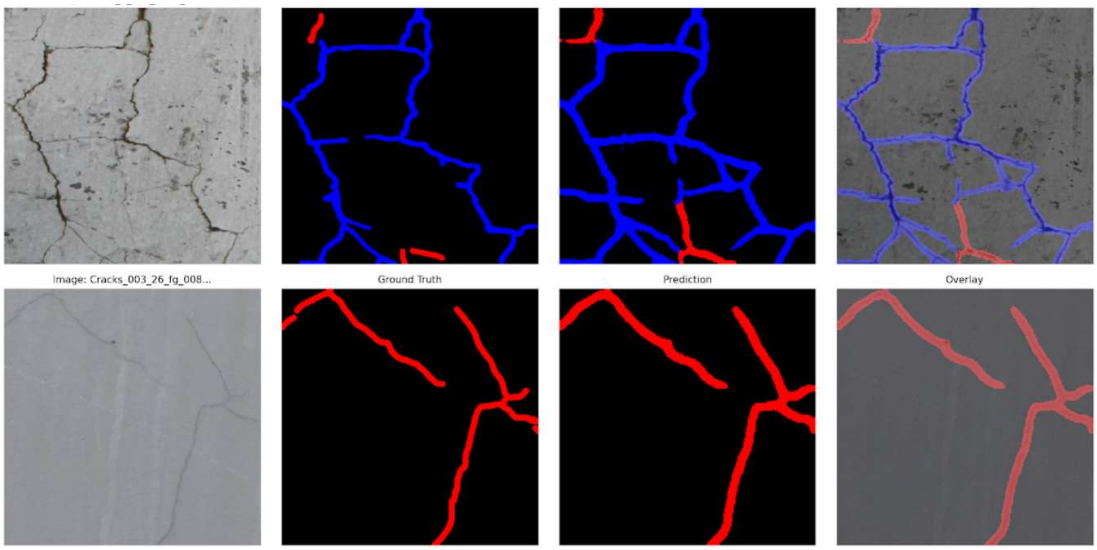

Fig. 8. Segmentation model predictions

3.3. Development of a Cascaded Crack Detection Pipeline

To address the challenge of comprehensive defect analysis, an approach based on the sequential integration of YOLOv11 detection and U-Net segmentation models is proposed. The architectural schematic of the algorithm is illustrated in Figure 9.

The core principle of the proposed approach lies in the sequential application of models designed for distinct tasks. In the first stage, YOLOv11 performs rapid localization of potential defects across the entire image, generating bounding boxes with associated class labels and confidence scores. Following confidence-based filtering, image patches corresponding to each detected region are extracted. In the second stage, each patch is resized to a resolution of 640×640 pixels and fed into the U-Net model, which produces a precise pixel-level segmentation mask of the defect. The resulting masks are then rescaled to their original dimensions and projected back onto the source image. The final output comprises the region coordinates, class label, confidence score, and a detailed pixel-wise mask delineating the exact crack boundaries.

The cascade approach provides several advantages. First, computational cost is reduced, as the segmentation model is applied only to regions containing potential defects rather than to the entire image. Second, the number of false positives is decreased through a two-stage verification process, which lowers the probability of erroneous defect detection. Third, informational completeness is achieved: rapid localization enables efficient screening

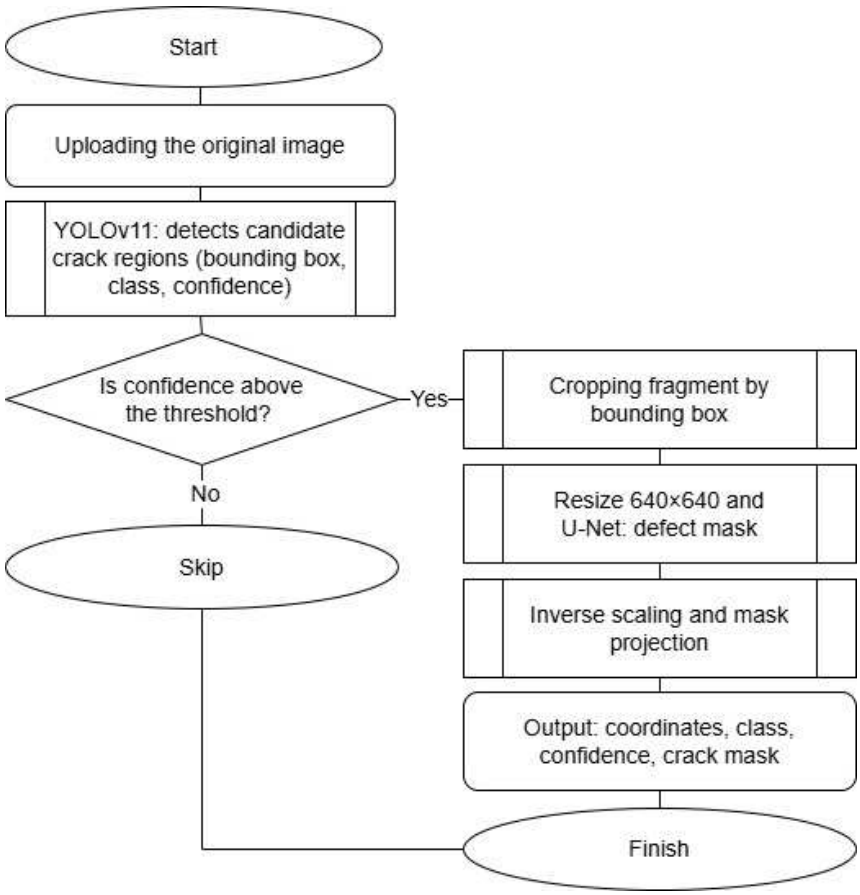

Fig. 9. The principle of cascade conveyor operation

of large image datasets, while detailed geometric characterization of defects supports engineering analysis and restoration planning.

3.4. Comparative Analysis of Models

A comparative analysis of the detection and segmentation models demonstrated their complementary nature (Table 4).

The YOLOv11 model provides high processing speed and is optimal for preliminary screening tasks, namely the rapid identification of images containing defects and the estimation of their approximate locations. However, the rectangular shape of the bounding box does not allow for an accurate representation of the geometry of tortuous cracks.

The U-Net model ensures significantly higher boundary delineation accuracy. The pixel-wise mask enables the computation of quantitative crack characteristics, including damaged area, length, average and maximum width, and propagation direction. This information is essential for assessing defect severity and planning the scope of restoration work.

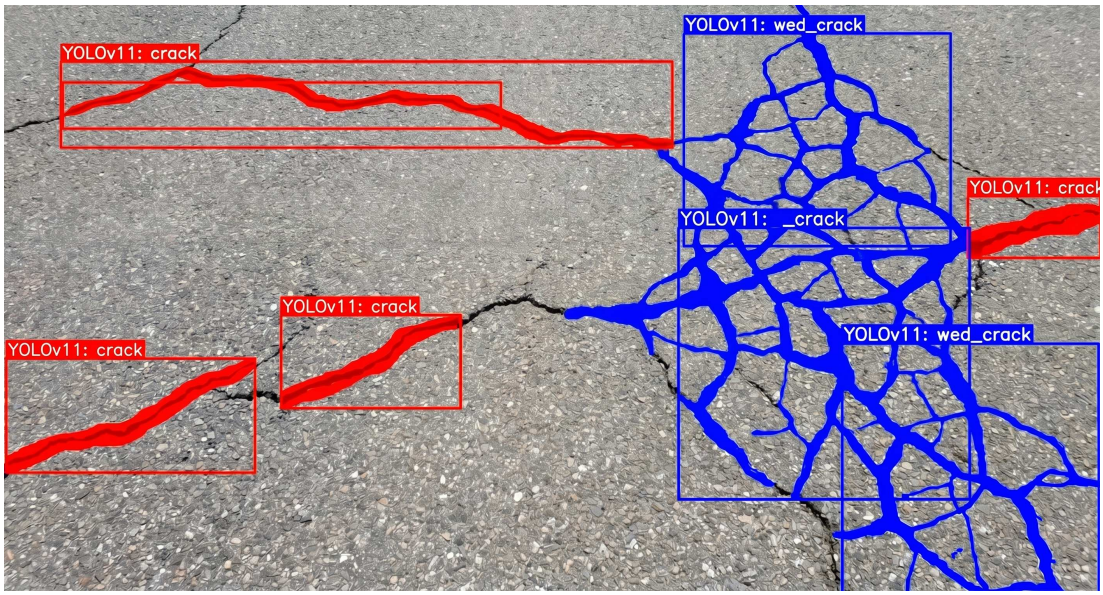

Fig. 10. Cascade pipeline output

Table 4

Comparison of crack detection approaches

Criterion	**YOLOv11**	**U-Net**	**Cascade**
Output data	Bounding box + class	Pixel mask	Both
Boundary accuracy	Low	High	High
Processing speed	High	Medium	Medium
Defect geometry	No	Yes	Yes
Large-scale screening	Yes	No	Yes
Detailed analysis	No	Yes	Yes

Conclusion

In this study, methods for the automatic detection of cracks in images of cultural heritage objects were developed and investigated using the YOLOv11 and U-Net neural network architectures.

A specialized dataset consisting of 10,200 images reflecting the specific characteristics of cultural heritage objects was constructed. The dataset was annotated in two formats (YOLO for detection and pixel-wise masks for segmentation) and divided into training, validation, and test subsets.

The YOLOv11 model achieved an mAP@0.5 score of 0.86, exceeding the target threshold of 0.70. The U-Net model achieved an mIoU value of 0.79, surpassing the target threshold of 0.65. Both models demonstrated the ability to accurately detect and classify cracks of both types.

The comparative analysis revealed the complementary nature of detection and segmentation approaches. The proposed cascade pipeline ensures an optimal balance between processing speed, accuracy, and informational completeness, representing a practically significant solution for monitoring the condition of cultural heritage objects.

Future work should includes expanding the dataset with additional defect classes, investigating transformer-based architectures, and developing a mobile application for field inspections.

References

1. Sirota A.A., Dryuchenko M.A., Ivankov A.Y. Steganalysis of Digital Images Using Shallow and Deep Machine Learning Methods: Known Approaches and New Solutions. *Proceedings of VSU, Series: Systems Analysis and Information Technologies,* 2021, no. 1, pp. 33–52. DOI: 10.17308/sait.2021.1/3369. (in Russian)
2. Krylov S.A., Kashevarova G.G. Automation of the Process of Assessing the Technical Condition of Exterior Walls of Brick Buildings Using Machine Learning. *Herald of Daghestan State Technical University. Technical Sciences*, 2025, vol. 52, no. 3, pp. 61–70. DOI: 10.21822/2073-6185-2025-52-3-61-70. (in Russian)
3. Koul A., Ganju S., Kasam M., *Practical Deep Learning for Cloud, Mobile, and Edge*. Sebastopol, O'Reilly Media, 2019.
4. Ronneberger O., Fischer P., Brox T., U-Net: Convolutional Networks for Biomedical Image Segmentation. *Lecture Notes in Computer Science*, 2015, vol. 9351, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28
5. Protopapadakis E., Voulodimos A., Doulamis A., Doulamis N., Stathaki T. Automatic Crack Detection for Tunnel Inspection Using Deep Learning and Heuristic Image Post-Processing. *Applied Intelligence*, 2019, vol. 49, no. 7, pp. 2793–2806. DOI: 10.1007/s10489-018-01396-y.
6. Ultralytics YOLO Documentation. – https://docs.ultralytics.com/.

Tatyana V. Karpeta, PhD (Physics and Math), Associate Professor, South Ural State University (Chelyabinsk, Russian Federation), eroshkinatv@susu.ru

Elizabeth V. Kazantseva, Bachelor of Mathematics, Department of Mathematical and Programming, South Ural State University (Chelyabinsk, Russian Federation), kazantsevaliza04@gmail.com

Received May 17, 2026

УДК 004.896:004.932.72'1 DOI: 10.14529/jcem260205

ТЕХНОЛОГИИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ ОБНАРУЖЕНИЯ ТРЕЩИН НА ОБЪЕКТАХ КУЛЬТУРНОГО НАСЛЕДИЯ

Т. В. Карпета[1]***, Е. В. Казанцева***[1]
[1]Южно-Уральский государственный университет, г. Челябинск, Российская Федерация

В связи с потребностью в регулярном мониторинге технического состояния объектов культурного наследия для своевременного выявления дефектов в данной работе были использованы нейросетевые методы компьютерного зрения. В статье описан процесс формирования и разметки датасета, содержащего изображения трещин на исторических зданиях. Особое внимание уделено описанию сбора и подготовки данных для обучения, которые оказывают большое влияние на последующие метрики качества модели. Приведены результаты обучения обеих моделей и выполнен их сравнительный анализ. Предложен каскадный конвейер, объединяющий детекцию и сегментацию для повышения точности и надёжности обнаружения дефектов.

Ключевые слова: нейронные сети; детекция объектов; сегментация изображений; объекты культурного наследия; трещины; YOLOv11; U-Net.

Литература

1. Сирота, А.А. Стегоанализ цифровых изображений с использованием методов поверхностного и глубокого машинного обучения: известные подходы и новые решения / А.А. Сирота, М.А. Дрюченко, А.Ю. Иванков // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. – 2021. – № 1. – С. 33–52. – DOI: 10.17308/sait.2021.2/3482.

2. Крылов, С.А. Автоматизация процесса оценки технического состояния наружных стен кирпичных зданий с использованием технологии машинного обучения / С.А. Крылов, Г.Г. Кашеварова // Вестник Дагестанского государственного технического университета. Технические науки. – 2025. – Т. 52, № 3. – С. 61–70.

3. Коул, А. Искусственный интеллект и компьютерное зрение. Реальные проекты на Python / А. Коул, С. Ганджу, М. Казам; перевод с английского Е.А. Матвеевой. – СПб: Питер, 2021. – 560 с.

4. Ronneberger, O. U-Net: Convolutional networks for biomedical image segmentation / O. Ronneberger, P. Fischer, T. Brox // Lecture Notes in Computer Science. – 2015. – V. 9351. – P. 234–241.

5. Protopapadakis, E. Automatic crack detection for tunnel inspection using deep learning and heuristic image post-processing / E. Protopapadakis, A. Voulodimos, A. Doulamis, N. Doulamis, T. Stathaki // Applied Intelligence. – 2019. – V. 49, №7. – P. 2793–2806.

6. Ultralytics YOLO Documentation. – https://docs.ultralytics.com/.

Карпета Татьяна Васильевна, кандидат физико-математических наук, доцент кафедры прикладной математики и программирования, Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), eroshkinatv@susu.ru

Казанцева Елизавета Викторовна, бакалавр математики, кафедра прикладная математика и программирование, Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), kazantsevaliza04@gmail.com

Поступила в редакцию 17 мая 2026 г.