DOI:10.22337/2587-9618-2020-16-4-38-49

A CRACK DETECTION SYSTEM FOR STRUCTURAL HEALTH MONITORING AIDED BY A CONVOLUTIONAL NEURAL NETWORK AND MAPREDUCE FRAMEWORK

Darya Filatova [1], Charles El-Nouty [2]

[1] École Pratique des Hautes Études, Paris, FRANCE
[2] Université Sorbonne Paris Nord, Paris, FRANCE

Abstract: The quickly expanded development of artificial intelligence offers alternative ways to solve numerous civil engineering problems. The work is devoted to the development of a computer-vision-based crack detection system capable to process big data related to pathology recognition. In this study, we discuss an automated crack type classification pipeline based on CNN deep learning algorithm and MapReduce framework. The results of numerical modeling illustrate the potential of the crack detection system.

Keywords: deep learning, ground truth segmentation, machine learning, structural health monitoring, crack detection, MapReduce

СИСТЕМА ОБНАРУЖЕНИЯ ТРЕЩИН ДЛЯ МОНИТОРИНГА СОСТОЯНИЯ КОНСТРУКЦИЙ С ПОМОЩЬЮ СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ И ФРЕЙМВОРКА MAPREDUCE

Дарья Филатова [1], Шарль Эль-Нути [2]

[1] Практическая школа высших исследований, г. Париж, ФРАНЦИЯ
[2] Университет Сорбонна Париж Север, г. Париж, ФРАНЦИЯ

Аннотация: Быстрое развитие искусственного интеллекта предлагает альтернативные способы решения многочисленных проблем гражданского строительства. Работа посвящена разработке системы обнаружения трещин на основе компьютерного зрения, способной обрабатывать большие данные, связанные с распознаванием патологии. В этом исследовании мы обсудили конвейер автоматической классификации типов трещин, основанный на алгоритме глубокого обучения CNN и инфраструктуре MapReduce. Результаты численного моделирования иллюстрируют потенциальные возможности системы обнаружения трещин.

Ключевые слова: глубокое обучение, разметка достоверных данных, машинное обучение, мониторинг состояния конструкции, обнаружение трещин, MapReduce

1. INTRODUCTION

Thinking on the main principles of orthogenesis, taking into account that some "driving force" of evolutionary progress is always applied to reach a goal, it becomes obvious that the large-scale trends in massive digital transformation increase informational complexity. Started in the 2010th, the explosion of large volumes of structured and non-structured data production (text, image, video, and audio) initiated the so-called “fourth wave” of data evolution. Only several years later the “4th Industrial Revolution” stimulated the development of new industrial relations. That is to say, to stay on the market each company has to adapt in real-time available resources (technical or human) and solutions with respect to the shrinking technological innovation cycle. This makes "data"-related questions (format, quantity and quality, availability, time variation, etc.) highly important. Being influenced by technological progress, civil engineering adapts to the new environmental conditions. Many companies handle with cloud tools, storage cloud spaces, third-party calculations to speed up the project realization on all its life-stages with respect to technological innovations. Since no human and no "ordinary computer" can evaluate the millions of variables concerning a real-world phenomenon new AI-based raw data processing platforms and frameworks have to be developed.

The main streams of AI applications development concern the data either deterministic (all factors are known and controlled) or probabilistic (all factors have a probability to happen). To illustrate these last we limit examples to genetic algorithms, swarm intelligence, and artificial neural networks, which are considered as a core subset of machine learning (ML) techniques (see Table 1). As it is possible to admit, the mentioned methods can be grouped as modeling, optimization, control, or forecast and are related to all the life-stages of the construction process from the concept to maintenance and demolition. Despite the variety of the solutions, the expertise related tasks as structural damage detection (SDD) still have need of a human-specialist final decision. So, widely used visual, ultrasonic, or leak testing as well as acoustic emission, optical or laser methods can be augmented by ML technology, which in-depth development (mostly it is based on a statistical representation of a phenomenon under study) makes possible to automate most of these evaluations due to "deep feature extraction" and "feature classification" [12]. The choice of ML technology depends on the domain of application

Table 1. Some examples of AI techniques application in civil engineering

Technique	Method	Purpose	Application
Genetic algorithm (GA) [1–4]	• unsupervised and nonparametric • selective evolutionary • multi-objective fuzzy-genetic control	• to solve ill-posed large-scale optimization problem • to determine decision boundary • to monitor the over-time changes • to complete the sensitivity analysis	• structural damage detection • reduction of negative impact of environment on structures • shape and cross-section optimization of truss structures • cable tension changes control • vibration reduction • heat exchange
Swarm intelligence (SI) [5–9]	• particle swarm optimization • ant colony algorithm • bee colony algorithm • krill herd algorithm	• to solve global optimization problem in the hyper imensional space • to select reasonably design parameters • to update the dynamic models	• potential damage reduction (i.e. seismic safety of building, bridge vibrations) • slope stability analysis • optimal selection of geomechanical parameters • diverse design problems (tubular column, three-bar truss, helical compression string and etc.)
Artificial neural networks (ANN) [10–13]	• supervised • semi-supervised • non-supervised • reinforcement	• to solve highly complex ill-posed problems • to predict and forecast processes of different nature • to classify structural damages, predictive scoring, predictive maintenance • to detect abnormalities	• structural damage detection • reliability analysis • reduction of construction waste • structural optimization and control • health monitoring • damage detection

and of the area of engineering research. Let us consider the main advantages of ML-based SDD for some abstract civil construction.

The life cycle of any civil structure depends on cause-damage internal and external environmental factors (temperature, humidity, pressure, creep, corrosion, shrinkage, etc.). Therefore to prevent damages in the early stages, continuous evaluation of the structure is required. Despite the popularity, mostly referred to the surface inspection (fissures, cracks, leaks, etc.) the visual-based methods of damage detection become laborious and time-consuming for relatively large and complex structures demanding highly-skilled-trained experts. The main reasons are difficulties of access to certain parts of constructions as well as quantity and quality of structured and non-structured information to analyze.

Taking into account the progress being made in the field, the aim of this study is the development of an automated crack-detection system aided by a high-throughput deep learning algorithm for complex features extraction.

To reach the objective, we organize the work in the following manner. In Section 2, we analyze the existing ML-techniques based on computer vision and commonly used features for cracks detection. Section 3 deals with the methodological aspects of the cracks detection and classification system based on the semantic segmentation. Next, in Section 4, we evaluate the performances of the designed system and compare it with the other well-performed methods. Section 5 contains concluding remarks on advantages and limitations as well as recommendations for further development.

2. METHODS FOR DETECTING CRACKS

2.1. Cracks: naïve conception

Cracks and fissures refer to the phenomenon of the surface splitting without breaking it apart. Usually, they are observed as “lines” or “broken curves” of different width, length, and spatial orientation. Through empirical observation for specified surface (were-used materials, physical or chemical characteristics, a period of exploitation, environmental conditions, etc.), the reason of cracks appearance can be concluded. An introduction of cracks classification serves to estimate damage they would or already have caused. In this work, we would pass on only the principles of cracks detection. Hence the further considerations concern only concrete surfaces in the wide sense.

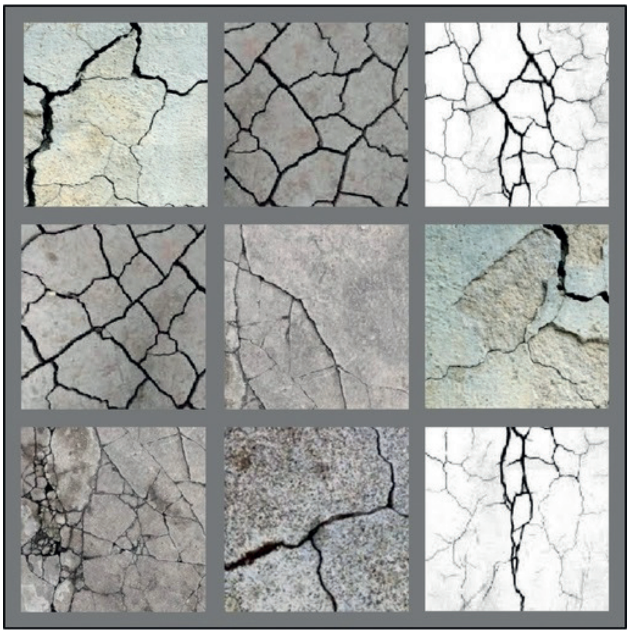

Figure 1. Some examples of the crack-damaged concrete surfaces

Table 2. Crack classification according opening (*a*, *b*, and *c* are the threshold values taken into account by an expert)

Severity class	Opening [mm]	Description
0	$< a$	out of consideration
1	$[a, b)$	an aesthetic nature
2	$[b, c)$	low risk of non-returnable surface damage
3	$\geq c$	elevated risk of non-returnable surface damage

To create an automated crack detection system, first, the classification criteria have to be settled, allowing experts to conclude on the damage state and to take the actions. Using visual analysis (see Fig. 1), cracks can be classified according to the *opening size* or *severity* (usually graduated measures in mm), to the *spatial orientation* (longitudinal, transversal, and miscellaneous), and to *expert-based characteristics* (plastic shrinkage, expansion, heaving, settling, overloading, crazing or crusting caused by premature drying, etc.). The diverse combinations of these features refer to multi-class classification problem, where classes and subclasses hierarchy has to be introduced by an expert with respect to the inspected structure. The example of the possible severity-based classification presents Table 2. As it is possible to notice this classification deal only with four levels of severity, other subdivision can be used without losing generalities.

We have also to admit that identification of spatial orientation as well as of expert-added characteristics depend on the architecture of the automated detection system and computer-vision techniques. Let us discuss some of them.

2.2. Computer-vision detection techniques

An automatic computer vision-based inspection of continuous surfaces (and, in a consequence, the defaults detection) supports a process of several stages, namely: image acquisition (surface lighting, camera configuration, synchronization, information storage), damage detection (preprocessing, localization, classification), and result exploitation (displaying and reporting information, decision making). The performance of the decision making depends on both hardware and software used in the detection system. It can be improved by increasing the quantity and quality of processed information subjected to decreasing processing time and controlling that repository of images fit in memory. In our opinion, the core element of the automatic damage detection system is the damage detection algorithm (see Fig. 2). Once its characteristics are available, the rest of the solution can be adapted for obtaining the desired parameters of the system as a whole.

The starting point of any damage detection algorithm supported by computer vision techniques is the digital image representation. The initial image is converted to a gray-scale one under selected resolution. Hence, the $n_1 - by - n_2$ pixels grayscale digital image corresponds to the surface associated with the bounded closed set

$$D = [0, n_1 - 1] \times [0, n_2 - 1] \subset R_+^2, \qquad (1)$$

where each pixel $(x, y) \in D$ is characterized by an intensity

$$f_{(x,y)} \in F,$$

where $F = \{f_{\min}, ..., f_{\max}\} \subset R_+$ is an ordered final set. Further, the values of intensities are sequentially transformed to find regions of suspected damage. The detection techniques, called the semantic segmentation, can be generalized as follows.

The fact that the pixel frequencies related to cracks are usually shifted to “black color,” such that the surrounding neighborhood seems to be lighter, is used to detect edges of cracks. Therefore choosing some threshold value one can point out damages.

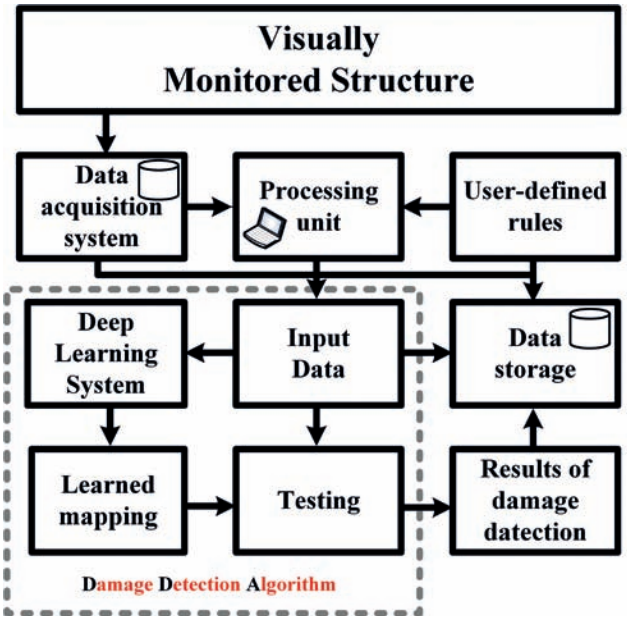

Figure 2. The scheme of the automatic damage detection system

However, if both regions have similar brightness, this kind of separation is hardly possible without additional pre-processed and post-processed transformations. This idea is supported by ML technique, which requires:

- preprocessing data (to eliminate anomalies related to acquisition system and to remove significant variations in initial intensities);
- segmentation (to detect desired features);
- classification (to group extracted features according to defined rules);
- post-processing (to refine the segmented and classified structures);
- evaluation (to analyze segmentation and classification quality).

As it is was indicated in [12–14], the accuracy of the damage detection algorithm mostly depends on strengths and weaknesses of the selected segmentation method (threshold-based, region-based, active contour model, mean-shift, K-means, Otsu, etc.). It can be significantly improved by deep learning methods using complex network architectures with significant volume of initial data. The CNN-based five-level architecture, developed in [13, 14] and called CrackNet, can be named as an example of automation of crack detection (see Fig. 3).

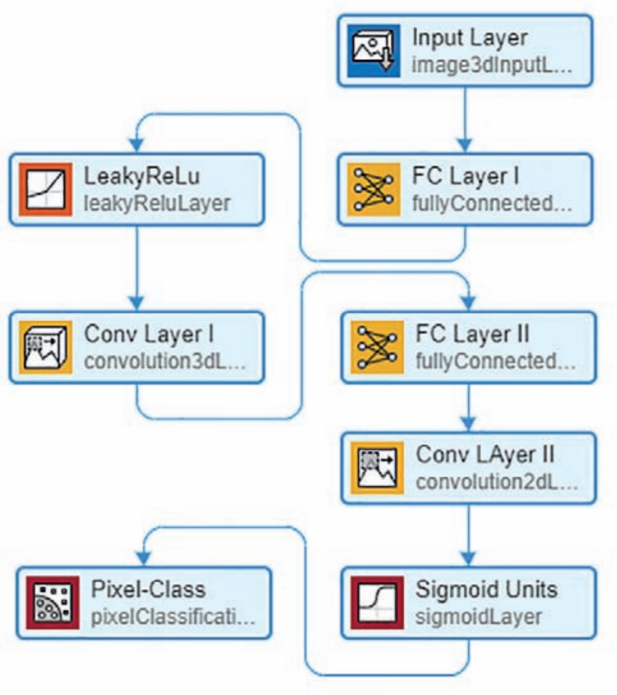

Figure 3. Sketch of overall architecture for crack detection – CrackNet [13]

To ensure desired pixel-wise accuracy authors had used more than million parameters. Despite the reduction of parameter space by introduction of more hidden layers, these methods are very time-consuming methods [15, 16].

To speed up the performances of the detection method some authors propose to treat the crack detection phase and crack characterization phase separately, reducing the number of processing operations (it can be done by Big Data technologies) and consistently getting rid of unnecessary information [17, 18].

3. RESEARCH METHODOLOGY

3.1. Method overview

The damage detection algorithm usually contains two stages:

- "*separation*", where regions of interest (ROI) are separated from the background, and
- "*classification*", where ROIs are subdivided into subsets according to some selected criteria.

Both stages refer to classification problems (binary and multi-class) and can be solved either by non-supervised or by supervised learning methods. In the case of supervised learning, the patterns to be recognized are compared with the labeled ones. In this work, we will apply the supervised methods to both stages each time solving the optimization tasks. Analysis of the patterns, where cracks were presented, shows that the damaged surface is much smaller than the background surface. This causes a problem of miss-balanced classification on the first stage and as a consequence miss-classification on the second one. To reduce miss-classification error a kind of regularization has been included to the optimization criteria.

Let I^{tag} be the labeled image, which corresponds to the image I defined by (1), where total number of pixels is $M = n_1 \cdot n_2$. We will call a pair (I, I^{tag}) as a training example. Labels of I^{tag} form M elements ground truth set T subdivided on K disjoint subsets T_k (classes). Each subset T_k, $k \in K$, contains m_k elements t_{kj}, $j = \{1, ..., m_k\}$, such that $\sum_k m_k = M, 0 \leq m_k \leq M$. To compensate the influence of larger classes, each subset is weighted as

$$w_k = \frac{1}{(m_k)^2 + \varepsilon},$$

where ε is a compensation term settled as a machine precision value to avoid division by zero, if T_k is an empty set.

We denote the results of prediction for some possible segmentation $Q = \{Q_k, 1 \leq k \leq K\}$ of the initial image I by q_{kj}, $k = \{1, ..., K\}, j = \{1, ..., m_k\}$. The wise-element loss between one image I and the corresponding ground truth image I^{tag} can be measured by the generalized Dice loss function:

$$\mathrm{L}(q,t) = 1 - \frac{\left(2\sum_{k=1}^{K} w_k \sum_{j=1}^{m_k} q_{kj} t_{kj}\right) + \varepsilon}{\sum_{k=1}^{K} w_k \sum_{j=1}^{m_k} \left(\left(q_{kj}\right)^2 + \left(t_{kj}\right)^2\right) + \varepsilon}, \qquad (2)$$

where the ε term is used to avoid division by zero, if Q_k or T_k are empty sets.

Hence, the classification turns to the solution of the optimization problem:

$$\mathrm{J}\ (q,t) = \underset{q \in Q}{\arg\min}\, \mathrm{L}(q,t). \qquad (3)$$

In the supervised learning, the problem (3) is repetitively solved N times on the set of training examples (I_{tr}, I_{tr}^{tag}):

$$\mathrm{J}^{\,*}(q,t) = \underset{\substack{q^{(i)} \in Q_{tr} \\ 1 \leq i \leq N}}{\arg\min} \tfrac{1}{N} \sum_{i=1}^{N} \mathrm{L}\left(q^{(i)}, t^{(i)}\right), \qquad (4)$$

where the index “tr” stands for the training set. The same optimization criteria is applied in case of severity class outlining. The severity-class labels are determined according to the method proposed in [17]. The crack opening is calculated on pixel level as average width along the crack skeleton:

$$\bar{u} = \frac{u_{cr}}{u_{sk}} r_s, \qquad (5)$$

where u_{cr} and u_{sk} are the numbers of pixels associated with the detected crack and its skeleton, whereas r_s is the spatial resolution of the acquisition device.

In both classification cases, the solution of (4) is considered as a predictive model. Once it have been evaluated on the validation set of examples (I_{val}, I_{val}^{tag}), each new image can be examined to determined damages. As it is possible to conclude, the crack detection and classification algorithm has two main phases: the training and the testing. The training phase requires the samples selection, preprocessing of initial data, the definition of the neural network architecture to create the predictive model. During testing phase the cracks are detected and classified, the results are evaluated by means of the performances metrics (see Fig. 4).

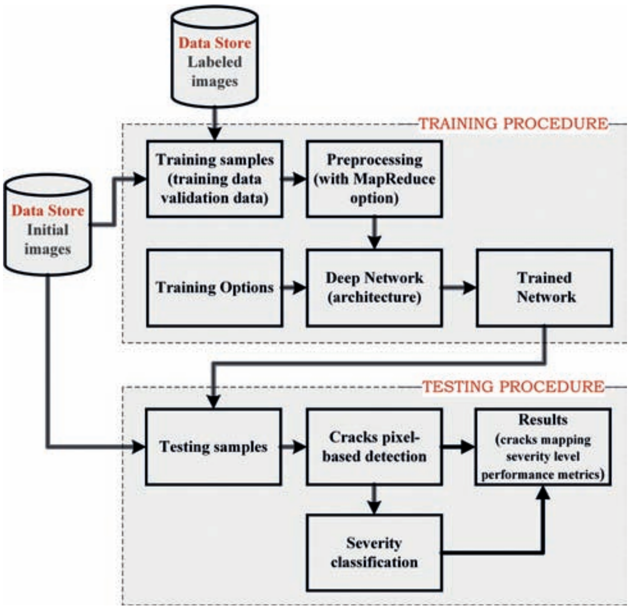
Data Store
Labeled images
TRAINING PROCEDURE
Data Store
Initial images
Training samples (training data validation data)
Preprocessing (with MapReduce option)
Training Options
Deep Network (architecture)
Trained Network
TESTING PROCEDURE
Testing samples
Cracks pixel-based detection
Results (cracks mapping severity level performance metrics)
Severity classification

Figure 4. Main modules of the crack detection and classification algorithm

We refer to [11] for more details on preprocessing and ROI identification. The edge detection methodology discussed in [11] defines the architecture of CNN used in the solution of (4) (see Fig. 5).

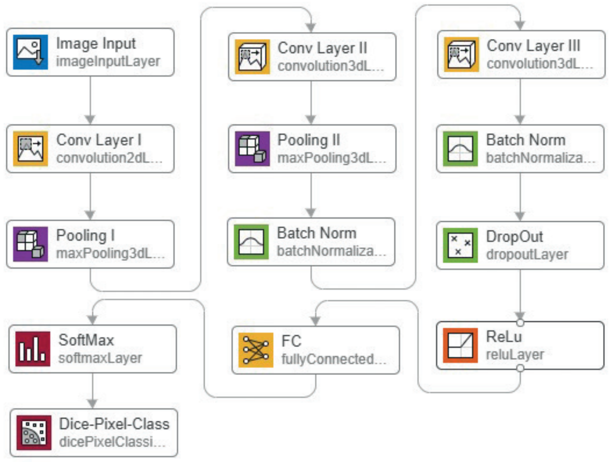

Figure 5. Sketch of overall architecture of "Deep Network" module.

To reduce the training time, the training procedure is implemented as a general MapReduce architecture. The main idea of this framework can be shortly described as follows. First training samples are divided on chunks. Each chunk is treated separately in parallel mode to perform a network training (data preprocessing, the weights initialization, forward propagation, backward propagation, the calculation of deviation, the calculation of offset, weights update such that to insure the convergence of the goal function (4)). As a result a "Key Value Store" is created. Next, these values are grouped by unique keys and reduced such that "Key Value Store" contains the parameters of the trained predictive model [19].

3.2. Evaluation metrics

The main idea of the crack detection algorithm is a binary classification of pixels as "cracked" or "non-cracked". Hence, to analyze the liaison between predicted and labeled ROIs, each target crack was marked by a minimum enclosing rectangle. The spatial coordinates of this last were used to indicate ROIs on the training sample. Therefore, accuracy of the classification can be measured by standard evaluation metrics such as

- precision

$$PPV = \frac{TP}{TP+FP}, \tag{6}$$

- recall sensitivity

$$TPR = \frac{TP}{TP+FN}, \tag{7}$$

- F1-score

$$F_1 = 2\frac{PPV \cdot TPR}{PPV+TPR}, \tag{8}$$

where TP and TN are the numbers of correctly classified "crack" and "non-crack" pixels as well as FP and FN are the number of misclassified pixels. These metrics can be used in a case of the severance-leveled classification. To study the global performance of this classification the mean-intersection-over-union is used. It can be estimated as

$$\overline{IoU} = \frac{1}{K} \sum_{k \in \{1,\ldots,K\}} \frac{TP_k}{TP_k+FP_k+FN_k}, \tag{10}$$

where K refers to the number of classes.

4. EXPERIMENT AND ANALYSIS

4.1. Dataset

The dataset consists of several concrete slab survey images with the same resolution. As it was mentioned before, the cracks areas are usually smaller than that of the total background. Hence, to avoid overloads of the training process we selected 4500 smaller blocks with a size 1024×1024 pixels. Next, on 300 patterns the cracks were manually labeled. Each labeled image was treated as the ground truth with two classes: "crack" and "no-crack" (see Fig. 6). The "crack" class was subdivided into four subclasses to indicate the severity level ("0" if $\overline{u} < 2mm$, "1" if $2mm \le \overline{u} < 4mm$, "2" if , "3" if $\overline{u} \ge 6mm$[1]). For

[1] This is just an example of classification. In practice, the norms have to be applied to define the severity classes [17, 18].

training-evaluation purposes the dataset was split on training, validation and test parts as 3:1:2.

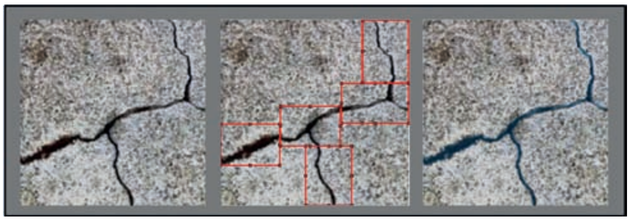

Figure 6. Example of image part selection and labeling

4.2. Implementation details

We used the personal computer with processor Intel (R) Core (TM) i7-470HQ CPU@2.50GHz RAM 16 Go and graphic card NVIDIA GeForce GTX 860M 1029MHz under 64-bit Windows 8.1 OS to explore the performance of the proposed methodology. The crack detection and classification algorithm was implemented in MatLab R2020b with DeepLearning, Parallel Computing and Computer Vision Toolboxes. MapReduce framework was executed on a parallel pool of four workers. The resolution parameter was automatically detected on the set of 100 images. This parameter was allied to find the average opening of detected cracks. The network was trained with the initial learning rate – “0.0001”, the maximum number of epochs – “50”, size of the mini-batch to use for each training iteration was “64”, the validation frequency – “50”. To insure the conversance of (4) we used the Adam optimization method setting $\varepsilon = 10^{-9}$, gradient decay factor as “0.9”, squared decay factor as “0.999”. Moreover, a random flip was applied to improve the robustness of the trained model.

4.3. Detection results

In the first stage of experiment we were interested only in binary classification task: the potential to find cracks. The performance of the proposed crack detection methodology (it is called as “*Method*” or “*MethodMR*” if MapReduce framework is applied) was compared with other semantic segmentation techniques based on the CNN architecture (U-Net [20] and CrackNet [15]). The experimental setup parameters as well as training, evaluation, and testing datasets were the same for each method. The methodology discussed in this paper was running twice: without MapReduce and with MapReduce framework. Table 3 contains the quantitative comparison of these methods using on the evaluation metrics (6) – (9) as well as training-evaluation time to finalize the predictive model. As it is possible to see, the proposed segmentation method supported by MapReduce framework achieves the highest performance. Results expose higher accuracy in binary classification and improvements in speedup and coincide with these were reported in [14, 18, 19].

Table 3. The performance comparison results on cracks detection using different architectures of CNN

CNN	PPV [%]	TPR [%]	F_1 [%]	$\overline{I_0U}$ [%]	$t_{tranining}$ [s]
U-Net	95.87	96.04	95.95	71.33	2784
CrackNet	96.18	97.09	96.63	72.54	2208
Method	97.99	97.43	97.71	74.15	2160
MethodMR	98.21	92.39	92.30	75.62	857

In the second stage of experiment, we have analyzed the severity levels detected by MethodMR. The severity classification results for one image are shown in Fig. 7. To evaluate the multiclass predictive model we use the receiver operating characteristic (ROC) curves (see Fig. 8) and estimated areas under the curves (AUC), namely:

$$AUC_{Class"0"} = 0.9159,\ AUC_{Class"1"} = 0.9488,$$
$$AUC_{Class"2"} = 0.9601,\ AUC_{Class"3"} = 0.9754.$$

As it is possible to see, *MethodMR* has better in-sample performance in classifying cracks with bigger openings.

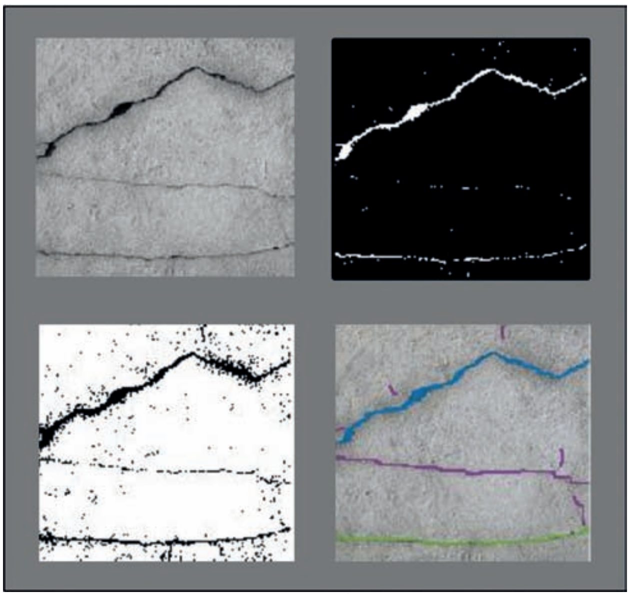

Figure 7. Severity classification marks on the test image ("purple" – class "1", "green" – class "2", "blue" – class "3")

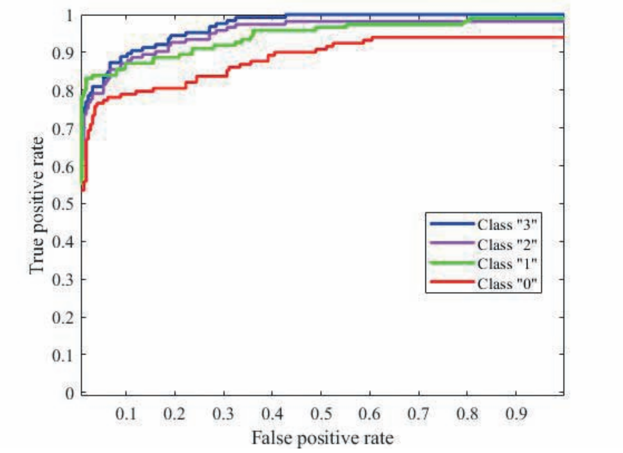

Figure 8. ROC curves for severity classification

5. CONCLUSIONS

The machine learning methods and especially convolutional neural networks have gained recognition in a wide array of applications of civil engineering permitting resolving the well-known problem of computer vision with better precision. Nonetheless, due to the enlargement of available information, the rise of computational complexity is observed. Big data severely impact training processes of CNN. The finalization of the CNN-based predictive model with desired accuracy suitable for the application has very high computational cost.

To overcome these obstacles, this study presented a novel semantic segmentation strategy for the automatic detection of cracks based on the distributed framework MapReduce applied to CNN training phase. The proposed method showed significant acceleration comparing to other segmentation networks. Moreover, results of cracks detection and classification reveal high accuracy. The proposed solution can be adapted to the specificity of the field of application to develop high-throughput damage detection systems implemented in portable devices using cloud technologies.

REFERENCES

1. **Papanicolaou C.G., Papantoniou I.C.** Optimum design of textile-reinforced concrete as integrated formwork in slabs // Editor(s): Thanasis Triantafillou, Textile Fibre Composites in Civil Engineering, Woodhead Publishing, 2016, pp. 245–274.
2. **Roetzel W., Luo X., Chen D.** Optimal design of heat exchanger networks // Editor(s): Wilfried Roetzel, Xing Luo, Dezhen Chen, Design and Operation of Heat Exchangers and their Networks, Academic Press, 2020, pp. 231–317.
3. **Behera B.K.** Modeling the structure of woven fabrics // Editor(s): K.L. Gandhi, [In] Woodhead Publishing Series in Textiles, Woven Textiles, Woodhead Publishing, 2012, pp. 229–263.
4. **Chen CJ.** An integrating genetic algorithm and modified Newton method for tracking control and vibration suppression // Artif Intell Rev, 2020, Vol. 53, pp. 3177–3199.
5. **Li L., Liu F.** Application of Particle Swarm Optimization Algorithm to Engineering Structures // [In] Group Search Optimization for Applications in Structural Design. Adaptation, Learning, and Optimization, vol 9. Springer, Berlin, Heidelberg, 2011.

6. **Filatova D.** Multi-agent simulation for self-healing mechanisms of damaged geotechnical structures // International Journal for Computational Civil and Structural Engineering, 2019, Vol. 15(4), pp. 48–57.
7. **Braun C.E., Chiwiacowsky L.D., Gómez A.T.** Variations of Ant Colony Optimization for the Solution of the Structural Damage Identification Problem // Procedia Computer Science, 2015, Vol. 51, pp. 875–884.
8. **Sonmez M.** Artificial Bee Colony algorithm for optimization of truss structures // Applied Soft Computing, 2011, Vol. 11 (2), pp. 2406–2418.
9. **Wang GG., Gandomi A.H., Alavi A.H. et al.** A comprehensive review of krill herd algorithm: variants, hybrids and applications // Artif Intell Rev, 2019, Vol. 51, pp. 119–148.
10. **Hakim S.J.S., Razak H.A., Ravanfar S.A.** Fault diagnosis on beam-like structures from modal parameters using artificial neural networks // Measurement, 2015, Vol. 76, pp. 45–61.
11. **Filatova D., El-Nouty Ch., Punko U.** High-throughput deep learning algorithm for diagnosis and defects classification of waterproofing membranes // International Journal for Computational Civil and Structural Engineering, 2020, Vol. 16(2), pp. 26–38.
12. **Avci O., Abdeljaber O., Kiranyaz S., Hussein M., Gabbouj M., Inman D.J.** A review of vibration-based damage detection in civil structures: From traditional methods to Machine Learning and Deep Learning applications, Mechanical Systems and Signal Processing, 2021, Vol. 147, pp. 107077
13. **Song Y., Huang Z., Shen Ch., Humphrey Shi, Lange D.A.** Deep learning-based automated image segmentation for concrete petrographic analysis // Cement and Concrete Research, 2020, Vol. 135, 106118.
14. **Zhang A., Wang K.C.P., Li B.X., Yang E.H., Dai X.X. et al.** Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network // Computer-Aided Civil and Infrastructure Engineering, 2017, Vol. 32 (10), pp. 805–819.
15. **Zhang A., Wang K.C.P., Fei Y., Liu Y., Tao S.Y.** Deep learning-based fully automated pavement crack detection on 3d asphalt surfaces with an improved CrackNet // Journal of Computing in Civil Engineering, 2018, Vol. 32 (50), 04018041.
16. **Huang Y., Li J., Fu J.** Review on Application of Artificial Intelligence in Civil Engineering // Computer Modeling in Engineering & Sciences, 2019, Vol. 121 (3), pp. 845–875.
17. **Oliveira H., Lobato Correia P.** CrackIT – an image processing toolbox for crack detection and characterization // 2014 IEEE International Conference on Image Processing (ICIP), Paris, 2014, pp. 798–802.
18. **Song W., JiaG., Jia D., Zhu H.** Automatic Pavement Crack Detection and Classification Using Multiscale Feature Attention Network // IEEE Access, 2019, Vol. 7, pp. 171001-171012.
19. **Zhao C., Dong M., Ota K., Wu J., Li J. and Li G.** MapReduce Enabling Content Analysis Architecture for Information-Centric Networks Using CNN // 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, 2018, pp. 1–6.
20. **Ronneberger O., Fischer P., Brox T.** U-Net: Convolutional Networks for Biomedical Image Segmentation // [In]: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, 2015, pp. 234–241.

СПИСОК ЛИТЕРАТУРЫ

1. **Papanicolaou C.G., Papantoniou I.C.** Optimum design of textile-reinforced concrete as integrated formwork in slabs // Editor(s): Thanasis Triantafillou, Textile Fibre Composites in Civil Engineering, Woodhead Publishing, 2016, pp. 245–274.
2. **Roetzel W., Luo X., Chen D.** Optimal design of heat exchanger networks // Editor(s): Wilfried Roetzel, Xing Luo, Dezhen Chen,

Design and Operation of Heat Exchangers and their Networks, Academic Press, 2020, pp. 231–317.

3. **Behera B.K.** Modeling the structure of woven fabrics // Editor(s): K.L. Gandhi, [In] Woodhead Publishing Series in Textiles, Woven Textiles, Woodhead Publishing, 2012, pp. 229–263.
4. **Chen CJ.** An integrating genetic algorithm and modified Newton method for tracking control and vibration suppression // Artif Intell Rev, 2020, Vol. 53, pp. 3177–3199.
5. **Li L., Liu F.** Application of Particle Swarm Optimization Algorithm to Engineering Structures // [In] Group Search Optimization for Applications in Structural Design. Adaptation, Learning, and Optimization, vol 9. Springer, Berlin, Heidelberg, 2011.
6. **Filatova D.** Multi-agent simulation for self-healing mechanisms of damaged geotechnical structures // International Journal for Computational Civil and Structural Engineering, 2019, Vol. 15(4), pp. 48–57.
7. **Braun C.E., Chiwiacowsky L.D., Gómez A.T.** Variations of Ant Colony Optimization for the Solution of the Structural Damage Identification Problem // Procedia Computer Science, 2015, Vol. 51, pp. 875–884.
8. **Sonmez M.** Artificial Bee Colony algorithm for optimization of truss structures // Applied Soft Computing, 2011, Vol. 11 (2), pp. 2406–2418.
9. **Wang GG., Gandomi A.H., Alavi A.H. et al.** A comprehensive review of krill herd algorithm: variants, hybrids and applications // Artif Intell Rev, 2019, Vol. 51, pp. 119–148.
10. **Hakim S.J.S., Razak H.A., Ravanfar S.A.** Fault diagnosis on beam-like structures from modal parameters using artificial neural networks // Measurement, 2015, Vol. 76, pp. 45–61.
11. **Filatova D., El-Nouty Ch., Punko U.** High-throughput deep learning algorithm for diagnosis and defects classification of waterproofing membranes // International Journal for Computational Civil and Structural Engineering, 2020, Vol. 16(2), pp. 26–38.
12. **Avci O., Abdeljaber O., Kiranyaz S., Hussein M., Gabbouj M., Inman D.J.** A review of vibration-based damage detection in civil structures: From traditional methods to Machine Learning and Deep Learning applications, Mechanical Systems and Signal Processing, 2021, Vol. 147, pp. 107077
13. **Song Y., Huang Z., Shen Ch., Humphrey Shi, Lange D.A.** Deep learning-based automated image segmentation for concrete petrographic analysis // Cement and Concrete Research, 2020, Vol. 135, 106118.
14. **Zhang A., Wang K.C.P., Li B.X., Yang E.H., Dai X.X. et al.** Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network // Computer-Aided Civil and Infrastructure Engineering, 2017, Vol. 32 (10), pp. 805–819.
15. **Zhang A., Wang K.C.P., Fei Y., Liu Y., Tao S.Y.** Deep learning-based fully automated pavement crack detection on 3d asphalt surfaces with an improved CrackNet // Journal of Computing in Civil Engineering, 2018, Vol. 32 (50), 04018041.
16. **Huang Y., Li J., Fu J.** Review on Application of Artificial Intelligence in Civil Engineering // Computer Modeling in Engineering & Sciences, 2019, Vol. 121 (3), pp. 845–875.
17. **Oliveira H., Lobato Correia P.** CrackIT – an image processing toolbox for crack detection and characterization // 2014 IEEE International Conference on Image Processing (ICIP), Paris, 2014, pp. 798–802.
18. **Song W., JiaG., Jia D., Zhu H.** Automatic Pavement Crack Detection and Classification Using Multiscale Feature Attention Network // IEEE Access, 2019, Vol. 7, pp. 171001–171012.
19. **Zhao C., Dong M., Ota K., Wu J., Li J. and Li G.** MapReduce Enabling Content Analysis Architecture for Information-Centric Networks Using CNN // 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, 2018, pp. 1–6.
20. **Ronneberger O., Fischer P., Brox T.** U-Net: Convolutional Networks for Biomedical

Image Segmentation // [In]: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, 2015, pp. 234–241.

Darya Filatova, Professor, Dr. hab., Scientific laboratory CHART EPHE, 4-14 Rue Ferrus, 75014 Paris, France; E-mail: dfilatova@interia.pl.

Charles El-Nouty, Professor, Dr. hab., LAGA, Université Sorbonne Paris Nord; 99 avenue J-B Clément; 93430 Villetaneuse; e-mail: elnouty@math.univ-paris13.fr.

Филатова Дарья, профессор, доктор физико-математических наук, Лаборатория человеческого и искусственного интеллекта, рактическая Школа Высших Исследований; 4-14, ул. Ферруса, 75014, Париж, Франция; e-mail: dfilatova@interia.pl.

Шарль Эль-Нути, профессор, доктор физико-математических наук, Лаборатория ЛАГА, Университет Сорбонна Париж Север; авеню Ж.-Б. Кленанта, д. 99, 94340 Вильтанез, Франция; E-mail: elnouty@math.univ-paris13.fr.