World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(Research Article)

Advanced machine learning approach for automatic crack detection and classification in concrete surfaces

Suat Gokhan Ozkaya [1] * and Mehmet Baygin [2]

[1] *Department of Construction Technologies, Ardahan University, Ardahan, Turkey.*
[2] *Department of Computer Engineering, Erzurum Technical University, Erzurum, Turkey.*

World Journal of Advanced Research and Reviews, 2023, 18(03), 1367–1379

Publication history: Received on 15 May 2023; revised on 28 June 2023; accepted on 30 June 2023

Article DOI: https://doi.org/10.30574/wjarr.2023.18.3.1282

Abstract

Nowadays, one of the most commonly used construction materials is concrete. As a building material, concrete can be deformed under various conditions and cracks can form on this material. Depending on their condition and position, these cracks can pose serious hazards. Therefore, the automatic detection and classification of these cracks becomes a very important issue. The detection process, which is usually performed by manual observation, is labor intensive. In this research, a new machine learning method is proposed for automatic detection of cracks in concrete surface. The proposed method utilizes DenseNet201 based deep feature extraction approach. In addition, the model includes ReliefF-based feature selection and SVM-based classification phases. SDNET2018, an open access dataset, is used to test the proposed model. Both holdout cross validation and 10-fold cross validation techniques were applied for validation on this dataset. As a result of the test procedures, 93% classification success was achieved for 10-fold CV. The results obtained with the test procedures prove the success of the proposed method in automatic crack classification.

Keywords: DenseNet201; Relief; Machine Learning; Concrete Crack; Classification

1. Introduction

Concrete, a robust and resilient construction material, finds extensive application in the construction industry [1]. It is widely employed in diverse structures such as bridges, buildings, dams, and roads [2]. Nevertheless, over time, concrete structures may manifest cracks due to various factors [3]. The presence of such cracks can significantly impact the structural integrity and longevity of these constructions, potentially leading to consequential damage [4]. Consequently, the timely detection and accurate classification of cracks assume paramount significance in preemptively identifying structural impairment and undertaking suitable remedial actions [5]. The manifestation of cracks on the concrete surface arises from diverse origins and encompasses a range of typologies [6]. Stress-induced cracks, for instance, materialize due to the concrete's exposure to external forces exerted upon it. Temperature-related cracks, on the other hand, emerge as a consequence of the concrete's reaction to fluctuations in thermal conditions. Tensile cracks emerge from the application of tensile forces upon the concrete, while impact cracks result from the imposition of external impacts. Surface cracks, as the name suggests, manifest specifically on the exterior of the concrete structure [7, 8]. This multifariousness in crack characteristics presents challenges in their systematic classification and comprehensive comprehension.

The conventional methods employed for crack detection and classification necessitate extensive time and human resources [9]. These approaches commonly rely on the expertise of trained professionals and visual inspections [10]. However, these methods have some disadvantages. First of all, the risk of human error in such an inspection is quite high and the level of accurate detection may decrease in the absence of experienced experts. Furthermore, the process is time-consuming and analyzing large datasets presents several challenges. Therefore, detecting and classifying cracks

* Corresponding author: Suat Gokhan Ozkaya

manually is a challenging and limiting task [11]. At this point, automatic crack classification approaches have been of great interest. Especially artificial intelligence techniques offer important solutions for this problem [12]. Automatic classification methods mainly incorporate artificial intelligence and image processing techniques [13, 14]. Artificial intelligence-based solutions are among the popular approaches used to achieve successful results in detecting and classifying complex structural problems. These approaches usually include deep learning and machine learning approaches.

Deep learning models, a prominent subset of artificial intelligence techniques, leverage the inherent capabilities of artificial neural networks to learn from data through their multilayered architectures and adversarial frameworks [15]. Consequently, when trained on extensive datasets, these deep learning models exhibit remarkable performance, particularly in tasks involving image recognition and classification. Convolutional neural networks (CNNs) are widely employed for such purposes. Alternatively, classical machine learning models represent another approach within the realm of artificial intelligence. These models encompass distinct stages of feature extraction and classification, operating on extensive datasets. Unlike deep learning models, classical machine learning models lack layered structures and generally possess lower computational complexity. However, the classification outcomes achieved through these methods may be comparatively inferior to those attained by deep learning models. In light of these considerations, deep learning approaches demonstrate elevated classification prowess despite their higher computational demands. Conversely, machine learning approaches offer advantages in terms of computational efficiency but may exhibit relatively lower classification success rates.

In this study, we propose a novel classification model that combines the principles of classical machine learning with the powerful feature extraction capabilities of deep learning models. To achieve this, we employ the DenseNet201 [16] network, a deep neural network architecture, for feature extraction. The extracted features are referred to as deep feature vectors. In the second phase of our model, we employ the ReliefF [17] method to select the most significant features, thereby eliminating irrelevant features. This feature selection process helps enhance the effectiveness of the subsequent classification step. In the final step of our proposed method, we employ support vector machines (SVM), a classical classifier, for the classification task. SVM [18] is a well-established algorithm for its ability to handle complex classification problems. To evaluate the performance of our developed model, we employ the k-fold cross-validation technique, specifically choosing k as 10. This approach allows us to assess the model's robustness and generalization capabilities. Following the testing process, we construct a confusion matrix and compute performance metrics based on this matrix. The metrics employed include accuracy, precision, recall, and geometric mean. These values serve as indicators of the proposed model's performance and provide insights into its classification capabilities.

Within this context, the second section of the study offers a comprehensive literature review. The third section presents the details of the dataset used and outlines the proposed method. In the fourth section, the experimental results and discussions are provided. Finally, in the fifth and final section, the study concludes with a summary of findings.

2. Literature Review

Nowadays, artificial intelligence techniques are actively used in many different fields such as medical diagnosis and treatment, autonomous driving, finance, industrial robotics [19, 20]. Artificial intelligence approaches, which attract the attention of many different disciplines, offer significant advantages in solving many problems that require intensive workload and manpower [21, 22]. In the field of construction, crack detection stands as a crucial problem that demands effective solutions. Hence, artificial intelligence techniques hold immense potential in addressing this challenge. Furthermore, a review of existing literature reveals a focused interest on automatic crack classification. In this context, some of the studies in the literature for machine learning based automatic crack classification are summarized in Table 1.

Table 1 Machine learning based automatic crack classification

Author(s)	Year	Dataset	Method	Result(s)	Limitation(s)
Dorafshan et al. [23]	2018	SDNET2018	AlexNet	Acc.=91.92	High computational complexity
Yang et al. [24]	2020	SDNET2018	CNN	Acc.=97.07 Pre.=99.80	High computational complexity

Ali et al. [25]	2021	SDNET2018 and METU crack dataset	CNN	Acc.=98.50 Pre.=100 Rec.=97.30 F1.=98.60	High computational complexity
Yang and Ji [26]	2021	SDNET2018	VGG16 based CNN and knowledge database	Acc.=94.95 Auc.=98.21	High computational complexity and relatively low classification performance
Abdelkader [27]	2021	SDNET2018	VGG16 based deep feature extraction, k-nearest neighbour and differential evolution algorithm	Acc.=99.83 F1.=99.05	Complex method
Priyadharshini et al. [28]	2023	SDNET2018	Quaternionic wavelet transform (QWT) and CNN	Acc.=98.44	High computational complexity
Laxman et al. [29]	2023	METU crack dataset	CNN, Random Forest and XGBoost	Acc.=93.70	High computational complexity and low classification performance

Acc.=Accuracy, F1.=F1-Score, Pre.=Precision, Rec.=Recall, CNN=Convolutional neural network

As depicted in Table 1, the literature predominantly showcases studies that utilize deep learning-based methods for automatic crack classification. These approaches have proven to yield high success rates in classification tasks, owing to their multilayered structures. Nevertheless, it is important to note that these methods come with high computational complexity. In this study, we propose a novel approach that aims to address this computational complexity concern. Our proposed method incorporates deep feature extraction, feature selection, and classification steps while emphasizing lower computational complexity. By leveraging the strengths of deep learning for feature extraction and employing efficient feature selection techniques, we aim to achieve effective crack classification results while reducing computational demands.

3. Material and methods

3.1. Material

In this study, we employed the SDNET2018 [23] database, which is an open-access dataset, to validate the proposed method. This database consists of crack images classified into three distinct categories: bridge deck, wall, and pavement images. Each category comprises two classes, namely crack and non-crack. The database encompasses a collection of over 56,000 crack images, exhibiting crack sizes ranging from 0.06 mm to 25 mm. It is important to note that the dataset represents real-world scenarios, containing various challenging factors such as shadows, rough surfaces, and background obstructions. For this study, we specifically focused on the bridge deck category of the dataset. Upon analysis, we found a total of 2,025 crack images within this category. Additionally, there were 11,595 images labeled as non-crack. To ensure a balanced dataset for testing purposes, we randomly selected 2,510 images from the non-crack labeled data. This selection process helped create a balanced test set. To provide a visual representation of the test set creation process, a block diagram illustrating the steps is presented in Figure 1.

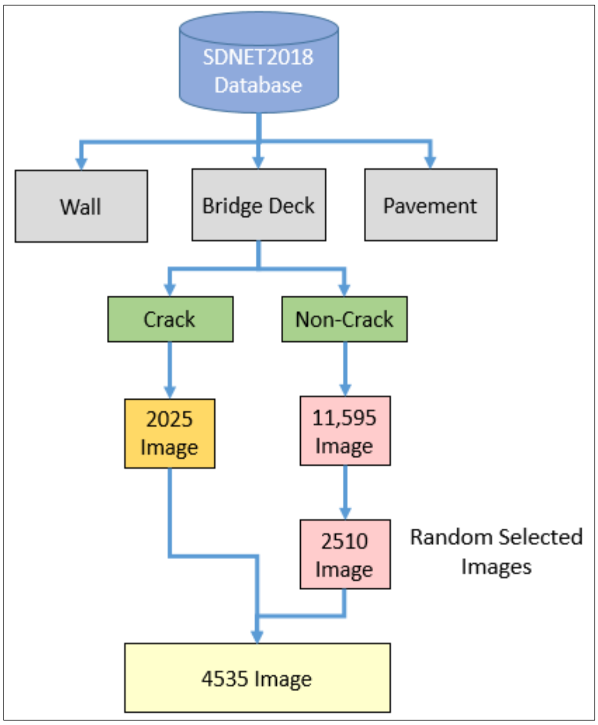

Figure 1 Dataset balancing

3.2. DenseNet201 based Automatic Crack Classification Method

This research presents a classification model that utilizes the pre-trained DenseNet201 [16] architecture. The proposed model incorporates feature extraction through this deep network architecture. The extracted features are then subjected to the ReliefF [17] algorithm, a feature selector, before being classified using the SVM [18] method. To provide a concise overview of the proposed model, a block diagram outlining its key components and their interconnections is presented in Figure 2.

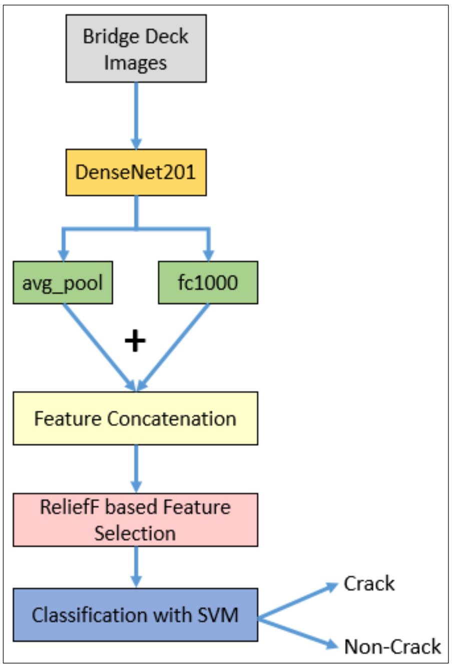

Figure 2 DenseNet201 based proposed method

As depicted in the figure, the developed model takes crack/non-crack images as input. The pre-trained DenseNet201 architecture is utilized for feature extraction. Specifically, the fully connected layer "fc1000" generates 1000 features, while the global average pooling layer "avg_pool" produces an additional 1280 features. These two sets of features are concatenated, resulting in a final feature vector with a total size of 2280 (1000 + 1280). To reduce the dimensionality of the feature vector and eliminate irrelevant features, the ReliefF algorithm is employed. This algorithm selects the top 1000 features based on their weights, effectively reducing the size of the feature vector. Following the feature selection stage, the classification process takes place. The selected feature vector is classified using the SVM algorithm, which assigns the input to the appropriate class. To provide a clearer understanding of the developed model, a pseudo code outlining its key steps is presented in Algorithm 1. This code summarizes the overall workflow and implementation of the model as described in the study.

Algorithm 1 Pseudocode of DenseNet201 based automatic crack classification

Input: SDNET2018 crack image dataset **Output:** Predicted vector	
00:	Load crack dataset.
01:	Get crack image.
02:	Deep feature extraction using the last global average pooling and fully connected layer of the DenseNet201 architecture.
03:	Combine features obtained in steps 2.
04:	Obtain a feature vector of size 2280.
05:	Repeat from first step until all images are complete.
06:	Select the 1000 most meaningful features using the ReliefF algorithm
08:	Apply the 10-fold cross validation and holdout validation method in the classification process.
09:	Classify the selected most significant features using the SVM algorithm.

The details of the steps given in Algorithm 1 are explained in subsections.

3.2.1. Feature extraction with DenseNet201

As stated earlier in this section, the developed model incorporates deep feature extraction utilizing the DenseNet201 [16] architecture. This architecture consists of 201 layers, and the pre-trained version is employed in this study. The DenseNet201 architecture has global average pooling and fully connected layers, which contribute to feature generation. Specifically, the global average pooling layer generates 1280 features, while the fully connected layer produces 1000 features. It is important to note that the DenseNet201 architecture was initially trained on the ImageNet database, enabling it to classify images into 1000 different categories. In this study, an approach based on transfer learning is adopted for feature extraction. This means that the pre-trained DenseNet201 architecture is utilized as a feature extractor rather than training it from scratch. By leveraging the knowledge and learned representations from the ImageNet training, the model can effectively extract meaningful features from the crack/non-crack images. To provide a visual representation of the feature extraction process using the DenseNet201 architecture, a block diagram is presented in Figure 3. This diagram summarizes the key steps involved in extracting features from the input images using the DenseNet201 architecture.

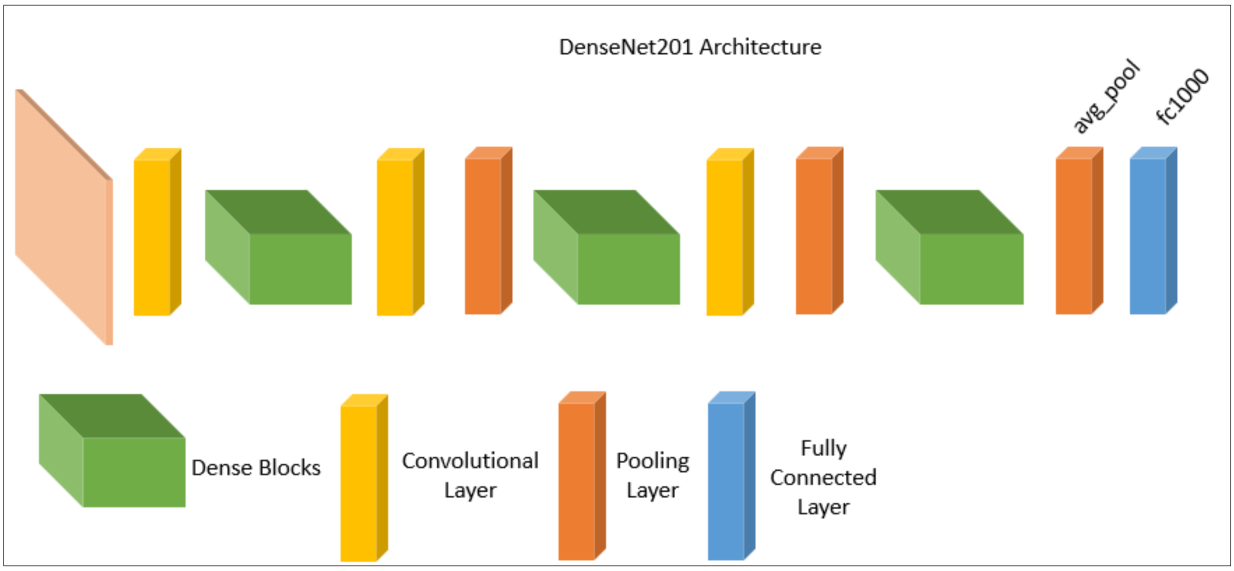

Figure 3 DenseNet201 architecture

As can be seen in Figure 3, the proposed approach uses the "avg_pool" and "fc1000" layers of the DenseNet201 network. Using these layers, 1280 and 1000 features are generated respectively.

Step 1: Generate feature vectors from DenseNet201 architecture using avg_pool and fc1000 layers.

$$D_1 = DenseNet201(im_j, avg_{pool}), j \in \{1,2, \dots, N\}$$
$$D_2 = DenseNet201(im_j, fc_{1000}), j \in \{1,2, \dots, N\} \quad (1)$$

where D_1 and D_2 are the features generated from the avg_{pool} and fc_{1000} layers respectively, im_j is the jth image in the dataset and N is the total number of images in the test set.

3.2.2. Feature concatenation

In this step, the features from the global average pooling and fully connected layers are combined to obtain the final feature vector. As a result of the feature concatenation process, a feature vector with a total size of 2280 (=1000+1280) is obtained. The mathematical equivalent of this process is given in Equation 2.

Step 2: Combine feature vectors.

$$cv_i = conc(D_i), i \in \{1,2\} \quad (2)$$

Herein, cv represents the concatenated feature vector and $conc$ represents the concatenation operation.

3.2.3. Feature selection

In order to improve the classification performance and reduce the computational complexity of the developed model, the RelieffF [17] algorithm, a well-known feature selector in the literature, is used. RelieffF algorithm considers each sample in different classes and selects the nearest neighbors. In this process, it uses the k-nearest neighbor algorithm and generates a weight value for each feature. The weight values produced by the RelieffF algorithm can be negative or positive. Features with negative weight values represent meaningless features. In this study, the first 1000 features with the highest weights were selected. This process step is given in Step 3.

Step 3: Select the top 1000 most meaningful features.

$$idx = ReliefF(cv)$$
$$sf(i,j) = cv(i, id(j)), i \in \{1,2, \dots N\}, j \in \{1,2, \dots, 1000\} \quad (3)$$

where idx is the weight values of the features, N is the number of images, j is the number of selected features and sf is the number of selected features. With this phase of the developed model, a selected feature vector of length 1000 is obtained.

3.2.4. Classification

The last phase of the developed model is classification. In this phase, SVM [18] algorithm is used to classify the test data into crack/non-crack classes. In the SVM algorithm, 10-fold CV strategy is used as a validation technique. This algorithm, which is frequently used in the literature, is a lightweight method. The parameters of the SVM method used in the model are given in Table 2.

Table 2 Hyper parameters of SVM classification algorithm

Parameters	**Value**
Kernel function	Quadratic
Box constraint level	1
Kernel scale mode	Auto
Multiclass method	One vs. One
Standardize data	Ok

Step 4: Classify selected features using SVM algorithm.

4. Experimental Results and Discussion

The developed model was tested using the SDNET2018 database, an open access database. This database contains three different categories of crack images. In this study, bridge deck images were used. For this purpose, all crack images and approximately 2500 non-crack images (for a balanced database) were used. The details of the testing process are given in the following sections.

4.1. Experimental Setup

In this paper, we use a deep feature extraction approach based on the proposed DenseNet201 architecture. In this perspective, end-to-end training is not applied in this study. The model is developed using a personal computer and the specifications of this computer are given in Table 3.

Table 3 Computer specifications

Parameters	**Value**
CPU	Intel Core i7 8th Gen
Ram	16 GB
Hard disk	256 GB
OS	Windows 11 Pro
Environment	MATLAB 2021b

As can be seen from Table 3, all testing was performed on a basic PC. The model was developed using the MATLAB platform. In addition, MATLAB Classification Learner Toolbox (MCLT) was used in the classification process. In the validation phase of the model, k-fold cross validation technique was preferred. Here, the value of k was chosen as 10. The main purpose of choosing this value is to easily compare with the literature. A confusion matrix was calculated to determine the performance of the model. Accuracy, precision, recall and geometric mean values were calculated using this matrix. These performance metric values are given in Equations (4)-(7) respectively.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$Pre = \frac{TP}{TP + FP} \quad (5)$$

$$Rec = \frac{TP}{TP + FN} \quad (6)$$

$$Geo = \sqrt{\frac{TP \times TN}{(TP + FN) * (TN + FP)}} \quad (7)$$

Where Acc refers to accuracy, Pre to precision, Rec to recall and Geo to geometric mean..

4.2. Experimental Results

In order to determine the performance of the model, a confusion matrix was calculated. This matrix shows the predicted label values against the correct label values. In this context, the calculated confusion matrix is given in Figure 4. Moreover, the performance metric values calculated using this matrix are presented in Table 4.

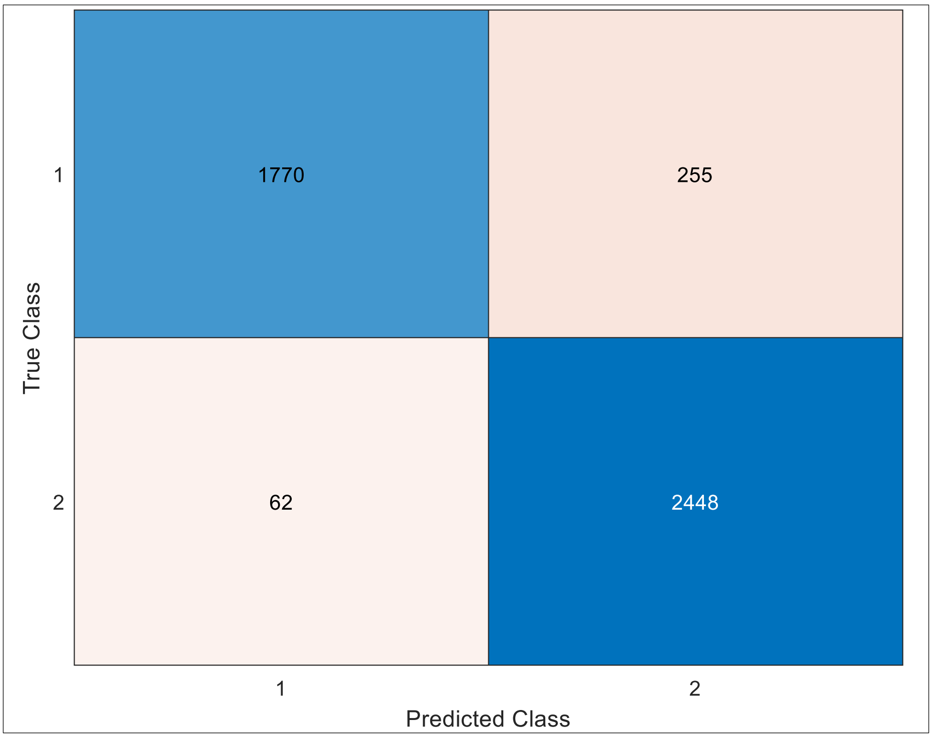

Figure 4 Calculated confusion matrix

Table 4 Performance metric values

Metric	**Value (%)**
Accuracy	93.01
Precision	96.62
Recall	87.40
Geometric Mean	92.22

As can be seen from Table 4, the DenseNet201 based feature extractor proposed in this study has a very high classification success. This model achieves over 93% classification success in separating crack/non-crack images. Moreover, the proposed method does not use an end-to-end training approach. Instead, it extracts features with a transfer learning approach, selects features with ReliefF and classifies the selected features with SVM algorithm. In this respect, the proposed model is similar to classical machine learning methods and its computational complexity is considerably lower than the literature.

4.3. Discussion

The model proposed in this research uses the DenseNet201 architecture. In the selection phase of this model, some pretrained networks were tested. The classification performance of these tested networks is given in Figure 5.

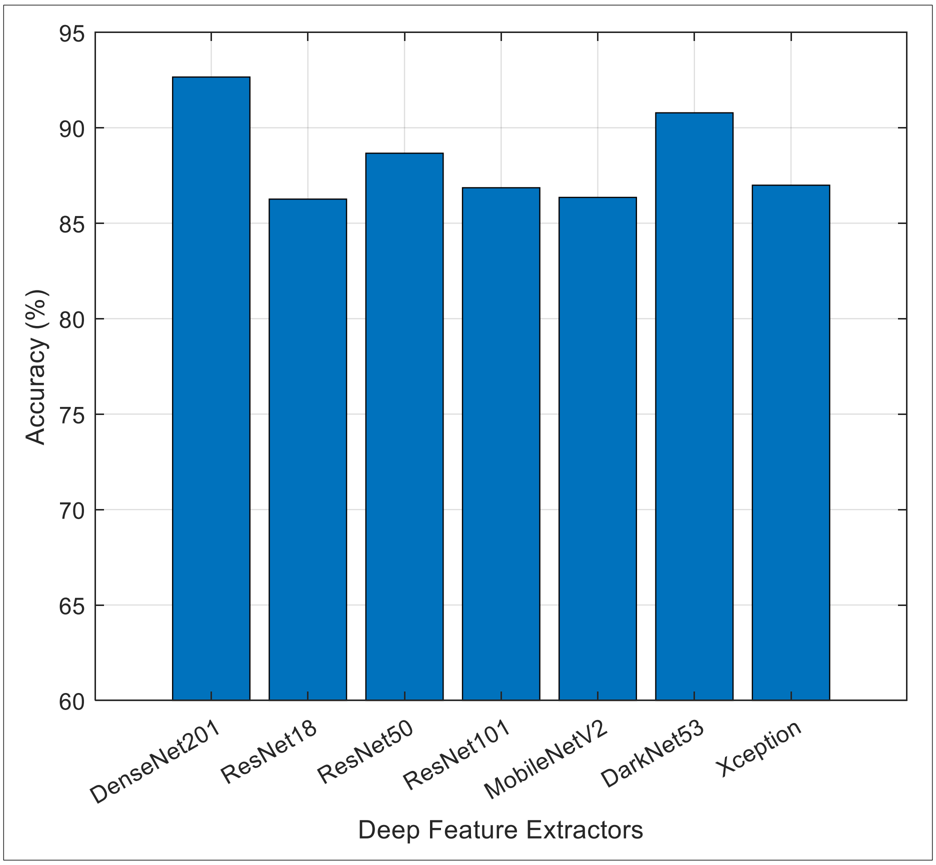

Figure 5 Tested pretrained deep learning architectures

As can be seen in Figure 5, the highest classification success was achieved with the DenseNet201 architecture. Therefore, this architecture was preferred in this study. In all of the tested networks, global average pooling and fully connected layers were used and the classification process was performed using ReliefF and SVM algorithms similar to the method used in this study. The number of features selected (1000) was determined by trial-error. Three different feature selectors were tested in the feature selection phase of the model. These are ReliefF, Chi2 and NCA methods respectively. The classification accuracies calculated using these selection methods are given in Figure 6.

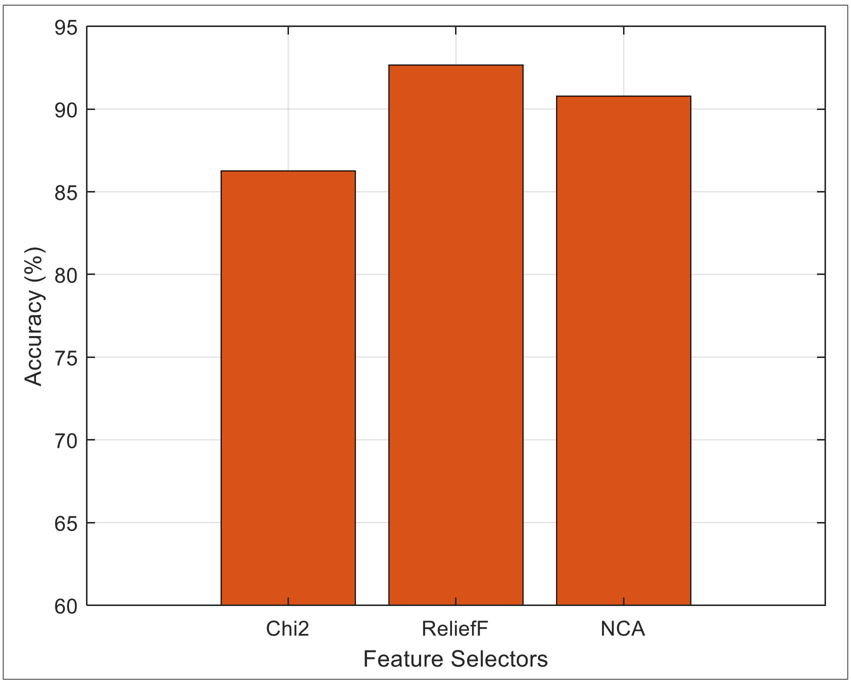

Figure 6 The performance of feature selectors

As shown in Figure 6, the highest classification accuracy was achieved with the ReliefF algorithm. Another phase tested in the model development process is classification. The tested classification algorithms and their calculated performances are given in Figure 7.

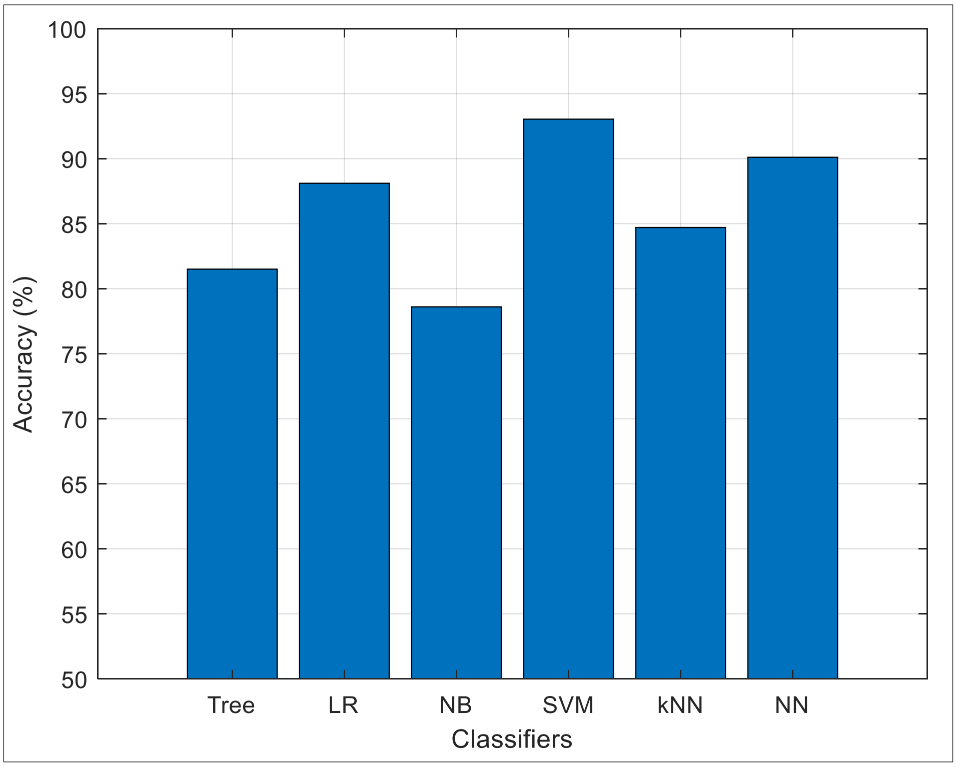

Figure 7 The performance of classifiers (LR: Linear regression, NB: Naïve bayes, SVM: Support vector machine, kNN: k-Nearest neighbour, NN: Neural network)

As can be seen from Figures 5, 6 and 7, the highest classification accuracy was obtained using DenseNet201, ReliefF and SVM algorithms. In the developed model, 10-fold CV method was used as a validation technique. However, the hold-out validation method was also tested within the scope of the research. In this method, the test data was separated and classified into two groups as training and test set. The ratios of this separation process are 60:40, 70:30, 80:20 and 90:10 respectively. Classification accuracies of the model according to these ratios are given in Figure 8.

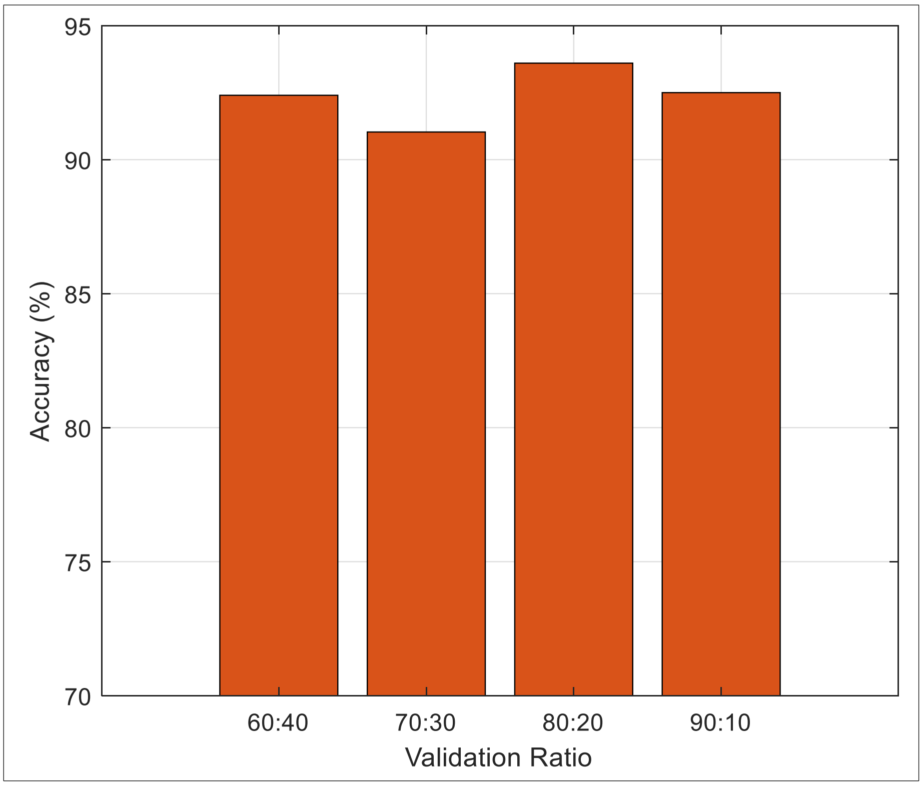

Figure 8 The results of holdout cross validation

The classification accuracies calculated using the hold-out validation method reveal the superiority of the proposed model. The obtained results show that the model has a very high potential to accurately classify crack images.

5. Conclusion

Concrete is the most intensively used material in the construction industry. This material can be deformed under various conditions and therefore cracks occur. The types and positions of these cracks are of great importance for the structure. Therefore, their detection and classification is very important. In this study, a new machine learning model for automatic crack detection is developed using an open access dataset.

The model consists of deep feature extraction, feature selection and classification phases. DenseNet201 architecture is used for deep feature extraction. In this architecture, 2280 features were extracted using the final pooling and fully connected layers. In the feature selection phase of the model, the most significant 1000 features were selected with the ReliefF algorithm and these selected features were classified using the SVM algorithm. The proposed method showed a very high success and achieved a classification accuracy of over 93%. The model developed in this study has a hybrid approach. In this context, the steps of classical machine learning are applied and the power of deep learning methods is utilized. In this respect, it has lower computational complexity compared to the literature.

Future Works

In future studies, it is planned to implement the proposed model in real time. For this purpose, it is aimed to collect concrete surface images through a camera to be placed on unmanned aerial vehicles and classify these images with the model proposed in the study. In this way, it is aimed to perform the labor-intensive concrete crack classification process automatically, quickly, safely and accurately.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest.

References

[1] F. Bos, R. Wolfs, Z. Ahmed, and T. Salet, "Additive manufacturing of concrete in construction: potentials and challenges of 3D concrete printing," *Virtual and physical prototyping,* vol. 11, no. 3, pp. 209-225, 2016.

[2] C. R. Gagg, "Cement and concrete as an engineering material: An historic appraisal and case study analysis," *Engineering Failure Analysis,* vol. 40, pp. 114-140, 2014.

[3] K. Bhargava, A. K. Ghosh, Y. Mori, and S. Ramanujam, "Model for cover cracking due to rebar corrosion in RC structures," *Engineering Structures,* vol. 28, no. 8, pp. 1093-1109, 2006.

[4] A. F. Grandt Jr, *Fundamentals of structural integrity: damage tolerant design and nondestructive evaluation*. John Wiley & Sons, 2003.

[5] M. C. Brown, J. P. Gomez, M. L. Hammer, and J. M. Hooks, "Long-term bridge performance high priority bridge performance issues," United States. Federal Highway Administration. Office of Infrastructure …, 2014.

[6] C. Baggio *et al.*, "Field manual for post-earthquake damage and safety assessment and short term countermeasures (AeDES)," *European Commission—Joint Research Centre—Institute for the Protection and Security of the Citizen, EUR,* vol. 22868, 2007.

[7] K. Ohno and M. Ohtsu, "Crack classification in concrete based on acoustic emission," *Construction and Building Materials,* vol. 24, no. 12, pp. 2339-2346, 2010.

[8] C. J. Larosche, "Types and causes of cracking in concrete structures," in *Failure, distress and repair of concrete structures*: Elsevier, 2009, pp. 57-83.

[9] N. Kheradmandi and V. Mehranfar, "A critical review and comparative study on image segmentation-based techniques for pavement crack detection," *Construction and Building Materials,* vol. 321, p. 126162, 2022.

[10] Y. J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Computer-Aided Civil and Infrastructure Engineering,* vol. 33, no. 9, pp. 731-747, 2018.

[11] Ç. F. Özgenel and A. G. Sorguç, "Performance comparison of pretrained convolutional neural networks on crack detection in buildings," 2018, vol. 35: IAARC Publications, pp. 1-8.

[12] R. Khallaf and M. Khallaf, "Classification and analysis of deep learning applications in construction: A systematic literature review," *Automation in construction,* vol. 129, p. 103760, 2021.

[13] M. Baygin, O. Yaman, P. D. Barua, S. Dogan, T. Tuncer, and U. R. Acharya, "Exemplar Darknet19 feature generation technique for automated kidney stone detection with coronal CT images," *Artificial Intelligence in Medicine,* vol. 127, p. 102274, 2022.

[14] S. G. Kobat *et al.*, "Automated diabetic retinopathy detection using horizontal and vertical patch division-based pre-trained DenseNET with digital fundus images," *Diagnostics,* vol. 12, no. 8, p. 1975, 2022.

[15] P. D. Barua *et al.*, "Multilevel deep feature generation framework for automated detection of retinal abnormalities using OCT images," *Entropy,* vol. 23, no. 12, p. 1651, 2021.

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2017, pp. 4700-4708.

[17] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," *Machine learning,* vol. 53, pp. 23-69, 2003.

[18] W. S. Noble, "What is a support vector machine?," *Nature biotechnology,* vol. 24, no. 12, pp. 1565-1567, 2006.

[19] E. Kaplan, S. Dogan, T. Tuncer, M. Baygin, and E. Altunisik, "Feed-forward LPQNet based automatic alzheimer's disease detection model," *Computers in Biology and Medicine,* vol. 137, p. 104828, 2021.

[20] G. Tasci *et al.*, "Automated accurate detection of depression using twin Pascal's triangles lattice pattern with EEG Signals," *Knowledge-Based Systems,* vol. 260, p. 110190, 2023.

[21] M. Baygin *et al.*, "A hand-modeled feature extraction-based learning network to detect grasps using sEMG signal," *Sensors,* vol. 22, no. 5, p. 2007, 2022.

[22] P. D. Barua *et al.*, "Automated EEG sentence classification using novel dynamic-sized binary pattern and multilevel discrete wavelet transform techniques with TSEEG database," *Biomedical Signal Processing and Control,* vol. 79, p. 104055, 2023.

[23] S. Dorafshan, R. J. Thomas, and M. Maguire, "SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks," *Data in brief,* vol. 21, pp. 1664-1668, 2018.

[24] Q. Yang, W. Shi, J. Chen, and W. Lin, "Deep convolution neural network-based transfer learning method for civil infrastructure crack detection," *Automation in Construction,* vol. 116, p. 103199, 2020.

[25] L. Ali, F. Alnajjar, H. A. Jassmi, M. Gocho, W. Khan, and M. A. Serhani, "Performance evaluation of deep CNN-based crack detection and localization techniques for concrete structures," *Sensors,* vol. 21, no. 5, p. 1688, 2021.

[26] Q. Yang and X. Ji, "Automatic pixel-level crack detection for civil infrastructure using Unet++ and deep transfer learning," *IEEE Sensors Journal,* vol. 21, no. 17, pp. 19165-19175, 2021.

[27] E. Mohammed Abdelkader, "On the hybridization of pre-trained deep learning and differential evolution algorithms for semantic crack detection and recognition in ensemble of infrastructures," *Smart and Sustainable Built Environment,* vol. 11, no. 3, pp. 740-764, 2022.

[28] R. A. Priyadharshini, S. Arivazhagan, and M. Arun, "Crack recognition on concrete structures based on machine crafted and hand crafted features," *Expert Systems with Applications,* vol. 228, p. 120447, 2023.

[29] K. C. Laxman, N. Tabassum, L. Ai, C. Cole, and P. Ziehl, "Automated crack detection and crack depth prediction for reinforced concrete structures using deep learning," *Construction and Building Materials,* vol. 370, p. 130709, 2023.