Performance Comparison of Pretrained Convolutional Neural Networks on Crack Detection in Buildings

Ç.F. Özgenel[a] and A. Gönenç Sorguç[b]

[ab]Department of Architecture, Middle East Technical University, Turkey
E-mail: fozgenel@metu.edu.tr, arzug@metu.edu.tr

Abstract –

Crack detection has vital importance for structural health monitoring and inspection of buildings. The task is challenging for computer vision methods as cracks have only low-level features for detection which are easily confused with background texture, foreign objects and/or irregularities in construction. In addition, difficulties such as inhomogeneous illumination and irregularities in construction present an obstacle for fully autonomous crack detection in the course of building inspection and monitoring. Convolutional neural networks (CNN's) are promising frameworks for crack detection with high accuracy and precision. Furthermore, being able to adapt pretrained networks to custom tasks by means of transfer learning enables users to utilize CNN's without the requirement of deep understanding and knowledge of algorithms. Yet, acknowledging the limitations and points to consider in the course of employing CNN's have great importance especially in fields which the results have vital importance such as crack detection in buildings. Within the scope of this study, a multidimensional performance analysis of highly acknowledged pretrained networks with respect to the size of training dataset, depth of networks, number of epochs for training and expandability to other material types utilized in buildings is conducted. By this means, it is aimed to develop an insight for new researchers and highlight the points to consider while applying CNN's for crack detection task.

Keywords –

Crack Detection in Buildings, Convolutional Neural Networks, Transfer Learning

1 Introduction

Architectural artefacts and civil infrastructures are exposed to loss of structural performance due to both deterioration of materials in time and structural challenges such as natural disasters. Structural monitoring and assessment of buildings have utmost importance for both sustaining the lifespan of structures and predict possible failures.

Visual crack inspection and detection is a widely used method for gaining insight into the condition of the architectural artefacts and structures. While the majority of the inspection is conducted by means of manual observations, several disadvantages of manual observation process are documented in literature such as being time-consuming and subjectivity of the evaluation. [1,2]

Advancements in robotics and image capturing hardware make autonomous data capturing possible while machine learning methods and deep learning algorithms in image processing show promise in the fully autonomous inspection of structures. Utilization of deep learning in these tasks not only provides reduction of computational time but also enables precise measurement of features to be inspected without human error.

On the other hand, autonomous conduction of visual crack detection is a challenging task for all image processing methods due to three major practical reasons caused by nature of the subject matter as:

1. discriminative crack features are low-level which is easily confused with noise in the background texture or foreign objects (such as hair or vegetation)
2. inhomogeneous illumination of the surface endangering the conservation of crack continuity [3]
3. irregularities in the application such as exposure of jointing

These practical challenges mentioned above are illustrated in Figure 1.

Resolving the practical challenges of images based crack detection in the course of autonomous inspection is an active field of study.

Convolutional neural networks (CNN's) are adequate frameworks with their high accuracy predictions in image classification and recognition tasks. There are several studies on crack detection on buildings and civil infrastructures with the use of CNN's. Within the scope of this study, it is aimed to investigate the relationship between the performance of CNN's and the affecting parameters.

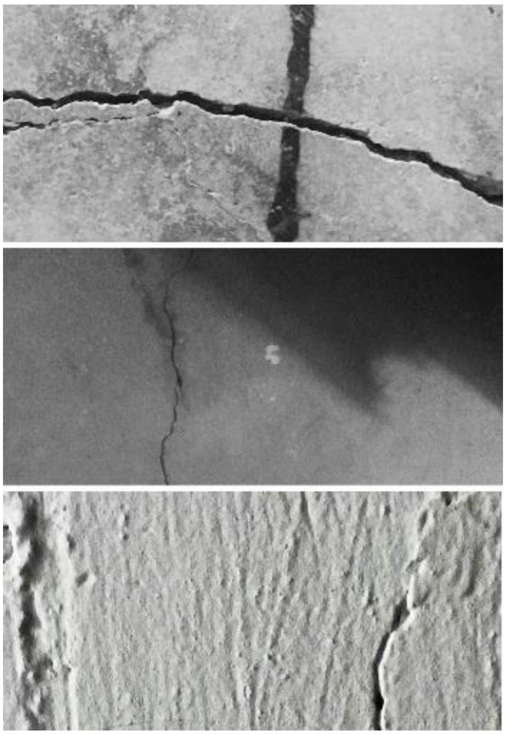

Figure 1. Practical challenges for crack detection, 1) the painting as the noise at the background (top), 2) shadow obfuscating crack and present noise (middle), 3) jointing at left presents noise (bottom)

2 Background

Visual crack detection task can be evaluated as a classification problem of crack presence in essence. Two types of methodological approaches are observed in the course of autonomous visual crack detection. The first type of studies is based on the sequential operation of feature extraction and classification by means of machine learning classifiers [4, 5]. In such studies, adaptive filters, transformations and/or morphological operations are utilized for extraction of features which are to be used for discriminating crack images from non-crack images. These features are then fed to machine learning classifiers to conduct classification. Studies of Adhikari *et al.* on pavement cracks [2] and Wu *et al.* and Sinha *et al.* on pipe defects [4,5] can be given as exemplary studies of this group. The second type of methodological approach is observed in studies utilizing deep learning (e.g. convolutional neural networks) which the feature extraction stage is conducted within the black box algorithm. In such a workflow, the input data is provided as raw images without the specification of features to search for, and algorithm finds patterns among the image data to conduct the desired task. As the features to be used for the classification of crack presence are determined by the system, human bias/error is avoided but replaced with the error of the system. In this sense, the workflow has a data-driven approach rather than knowledge-driven approach. Even though the adaptability and extensibility of the framework are more promising than sequential workflow, the errors and the factors affecting should be understood to further progress the potentials of deep learning based implementations.

Following studies are exemplary studies employing CNN's. It should be noted that it is not aimed to make a complete list but rather provide a baseline for the present study. The studies are compared in terms of number of convolutional layers utilized, number of images used for training and reported accuracy for crack detection. On the other hand, the classification accuracies of the inspected studies are highly dependent on the training and test datasets and not directly comparable.

Studies of Zhang *et al.* [6] and ASINVOS developed by Eisenbach *et al.* [7] can be given as example studies regarding the application of convolutional neural networks on crack detection on roads. Zhang, *et al.* uses a CNN with 6 convolution layers to conduct binary crack detection task on roads. Authors used 600K images for training and 200K for testing and got 0,8965 F_1 scores. The framework utilized in Eisenbach *et al.*'s study 11 convolution layers. Authors used 4,9 M image patches and tested the network with 1,2M images. ASINVOS is reported to score slightly better than the network developed by Zhang, *et al.* by scoring 0,7246 F_1 score compared to 0,6707 Zhang, *et al.*'s network on the dataset provided by Eisenbach *et al.* Similarly, Wang *et al.* [3] utilizes CNN with 5 convolutional layers for classification of asphalt pavement cracks but differently from the studies mentioned above, Wang *et al.* utilizes 3D data input including depth with 1mm resolution. Authors used 640K training image cells, 128K test image cells and scored 0,9429 accuracy. Pauly, *et al.* [8] also focus on crack detection on pavements and investigate the relation between the number of layers in CNN (deepness of network) by comparing performances of 6 layered and 7 layered networks. Authors also worked on two different subsets with the first subset contains 200K training images versus 40K test images, and the second subset contains 40K training images versus 60K testing images which are collected from different locations compared to training images. Study scored 0,913 accuracy with CNN containing 7 convolutional layers on the first subset. As the studies mentioned above are trained from scratch, they require a considerable amount of images for training which can be a limiting factor in terms of layers utilized in the network.

The study conducted by Cha, et al. [9] uses deep learning for crack damage detection for structural health monitoring. In that sense, the study is significant as it is implemented on building scale which the illumination conditions and forces which the material is subjected to show more variations compared to pavement and road inspections. Authors used a framework with 4 convolutional layers for concrete crack detection. The network is trained with several datasets with varying sizes from 2K to 40K images and testing is done with 54 full resolution images. Based on validation accuracies it is advised to use more than 10K images for training a network from scratch. For test results, mean accuracy of 0,9683 scored.

Availability of the pretrained networks eases the applicability of CNN's in new tasks without the requirement of high computational cost and deep knowledge on how CNN's operate. AlexNet developed by Krizhevsky, et al. [10], VGG networks developed by Oxford Visual Geometry Group [11], GoogleNets [12], and ResNet networks developed by Microsoft [13] can be given as examples for highly acknowledged pretrained networks which are used as the basis for application to new tasks. Study of Gopalakrishnan *et al.* [14] can be given as an example which uses transfer learning to utilize a pretrained network for pavement distress detection and employs the VGG16 network trained on ImageNet data. The study compares different classifiers in conjunction with VGG16 network. Authors used 760 images for training and 212 images for testing purposes and achieved the highest accuracy of 0,90 with the single-layered neural network classifier. When compared to studies which CNN's are trained from scratch, a similar accuracy is obtained with considerably fewer data with the use of transfer learning which is promising in terms of fast and easy implementation to new tasks.

All of these studies are concluded with accuracies above 90% for detecting cracks in images. On the other hand, performance of the mentioned studies based on several factors such as selection of data, number, and type of layers utilized other than convolutional layers, choice of filter sizes. Hence, these studies don't provide any indication of how deepness of networks and size of image datasets affect the performance of these frameworks.

Within the scope of this study, a comprehensive analysis on the applicability of CNN's on crack detection in building-oriented applications is conducted by means of transfer learning. In this regard, the influence of training dataset size, number of epochs used for training, number of convolution layers and learnable parameters on the performance of CNN's are inspected. In addition, transferability to new material types are investigated.

3 Data Preparation

In the present study, datasets utilized are explained in three categories as training, validation and test sets. The base dataset is obtained by extracting 40K image patches with the dimensions of 224 to 224 pixels, from 500 full resolution (4032 pixels to 3024 pixels) images taken from walls and floors of several concrete buildings in METU Campus. These images are taken approximately 1m away from the surfaces with the camera facing directly to the target. Even though the concrete surfaces have variation in terms of surface finishes (exposed, plastering and paint), the images are captured on the same day with similar illumination conditions. No data augmentation in terms of random rotation or flipping is applied. Image samples for training and test cases are shown in Figure 2. The base dataset is publicly shared [15].

The preparation of these datasets are explained as below:

Training dataset: As a convention, %70 of randomly selected images from the base dataset is used for training while %15 is used for validation throughout the training and %15 is used for testing. As a result, the biggest training dataset consists 28K images. The size of training dataset is then randomly reduced from 28K to 21K, 14K, 7K, 3,5K, 1,75K, 0,7K and 0,35K to imitate grid search for investigation of the relation between performance and size of the training dataset. All of the datasets are balanced in terms of classes, containing an equal number of positive and negative images.

Validation dataset: Validation dataset is used throughout the training to monitor the learning curve of the networks. The number of images used for validation is chosen with regard to the size of the respective training dataset size. The %70-%15 ratio between the training set and validation set is conserved for all training cases

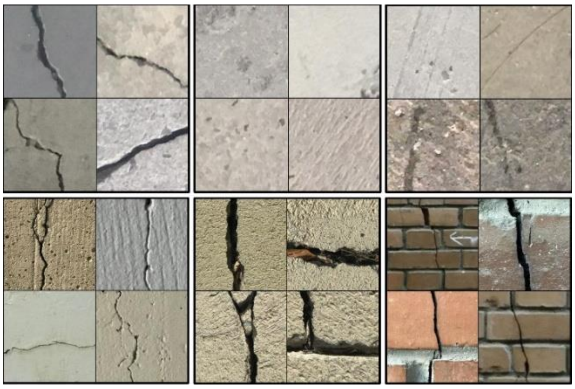

Figure 2. Training and test image samples. Positive training samples (top left), Negative training samples (top center), Possible false-positive training samples (top right), Concrete test samples (bottom left), Pavement test samples (bottom center), Brickwork test samples (bottom right)

Test datasets: Four distinct cases are chosen for testing purposes. The first case uses 6K images which are randomly chosen from the 40K base dataset. This partition corresponds to the %15 of the 40K dataset which is not utilized in either training or validation. All networks which are trained with varying sizes of training datasets are subjected to the same test dataset in order to observe the effect of training dataset size and performance in predicting visually similar images.

The second, third and fourth cases focus on the performance of the trained networks to investigate the transferability of learned features to new cases in terms of both physical conditions, such as illumination or camera angle, and material variations. The cases are respectively crack detection in pavements with concrete material, in buildings components with concrete material, and in buildings with brickwork material. 50 full resolution images are used for obtaining 500 test images per test case. All images are taken from different buildings and locations compared to the ones used training and validation at different times of the day.

The number of images used for training, validation, and tests is shown in Table 1 and Table 2.

Table 1. Number of images in datasets used for training, validation

	Training		Validation	
	Positive	Negative	Positive	Negative
28K	14000	14000	3000	3000
21K	10500	10500	2250	2250
14K	7000	7000	1500	1500
7K	3500	3500	750	750
3,5K	1750	1750	375	375
1,75K	875	875	188	188
0,7K	350	350	75	75
0,35K	175	175	38	38

Table 2. Number of images in datasets used for testing

Cases	Testing	
	Positive	Negative
15% randomly selected from base dataset (Test1)	3000	3000
Concrete Pavements (Test2)	250	250
Concrete Buildings (Test3)	250	250
Brickwork Buildings(Test4)	250	250

4 Performance Comparison

Within the scope of this study, the performance of seven highly acknowledged pretrained networks; namely, AlexNet [6], VGG16, VGG19 [7], GoogleNet [8] and ResNet50, ResNet101, and ResNet152 [9] on crack detection of concrete surfaces are inspected in relation with the size of training dataset size and number of epochs to obtain best results. In addition, the effect of complexity and depth of CNN's are investigated. The number of convolutional layers and number of learnable parameters which gives an insight into the number of filters used by networks is shared in Table 3.

Table 3. Pretrained networks, number of convolution layers and learnable parameters

	# of Convolution Layers	# of Learnable Parameters
AlexNet	8	60M
VGG16	16	138M
VGG19	19	144M
GoogleNet	22	7M
ResNet50	50	25.6M
ResNet101	101	44.5M
ResNet152	152	60.2M

The networks utilized in the scope of the study are pretrained on ImageNet data and obtained from MatConvNet website [16]. All tests are conducted using MatConvNet and Matlab on a desktop workstation with 2 Intel Xeon E5-2697 v2 @2,7 GHz CPU cores, 64GB RAM and NVIDIA Quadro K6000 GPU. While batch size is bounded with the GPU memory, other hyperparameters are determined as provided by MatConvNet website. On the other hand, the approach followed for the comparison is extendable to any pretrained network on any dataset with varying hyperparameter values.

Each of the pretrained network mentioned above is trained with training datasets shown in Table 1 for 10 epochs. It is observed that 10 epochs are sufficient for convergence of all network and after 10 epochs performance of networks fluctuate. Training on 7 pretrained networks on 8 different sizes of training datasets for 10 epochs yields 560 trained networks corresponding to all combinations of parameters taken into account in the comparison. After obtaining trained networks, the performance of these networks is evaluated with four cases resulting in 2240 scores which can be represented as 7x8x10x4 (network, dataset, epoch, test case respectively) matrix. For performance evaluation metrics, accuracy and F-scoring are used.

It is observed that the accuracy and F-scores of best-performing networks are compatible with each other. Even though the discussion regarding the performance of networks is advanced mentioning accuracy scores, the same remarks can be made for F-score results. For the sake of simplicity, the maximum accuracy and F-score results obtained per network are shown in Table 4 together with training dataset and epoch information. The results per row are discussed below.

Table 4. Maximum validation and test accuracies of pretrained networks

	AlexNet		VGG16		VGG19		GoogleNet		ResNet50		ResNet101		ResNet152	
	Mean Acc.	F-Score	Mean Acc.	F-Score	Mean Acc.	F-Score	Mean Acc.	F-Score	Mean Acc.	F-Score	Mean Acc.	F-Score	Mean Acc.	F-Score
Test1 Epoch1 0,35K	0.896	0.890	0.996	0.996	0.991	0.990	0.900	0.899	0.810	0.794	0.860	0.853	0.610	0.715
Test1 Epoch1	0.999	0.998	0.999	0.999	0.999	0.999	0.998	0.997	0.999	0.997	0.999	0.992	0.995	0.998
	28K	28K	21K	21K	28K	28K	21K	21K	14K	14K	21K	21K	14K	14K
Test1	0.999	0.999	0.999	1.000	0.999	1.000	0.999	0.999	0.999	0.999	0.999	0.999	0.998	0.998
	28K E6	28K E8	21K E1	21K E1	28K E3	28K E10	21K E9	21K E9	28K E2	14K E2	28K E6	28K E6	21K E10	21K E10
Test2 Concrete - Pavement	0.800	0.752	0.970	0.966	0.980	0.982	0.980	0.987	0.900	0.896	0.770	0.783	0.620	0.669
	28K E2	28K E2	21K E2	21K E2	3,5K E1	3,5K E1	0,35K E7	0.35K E7	0,35K E1	0,35K E1	7K E9	1,75K E8	7K E1	28K E9
Test3 Concrete - Building	0.860	0.856	0.980	0.977	0.960	0.963	0.920	0.916	0.640	0.695	0.740	0.746	0.540	0.681
	0,7K E1	0,7K E1	1,75K E6	1,75K E4	3,5K E1	3,5K E1	1,75K E7	1,75K E9	0,35K E1	0,7K E7	0,7K E1	0,7K E1	14K E6	0,7K E10
Test4 Brickwork -Building	0.870	0.855	0.960	0.955	0.880	0.890	0.900	0.912	0.690	0.739	0.720	0.741	0.620	0.696
	7K E1	7K E1	1,75K E4	1,75K E4	3,5K E5	3,5K E5	1,75K E9	1,75K E9	0,35K E1	0,35K E1	21K E10	3,5K E7	21K E10	14K E2

E: Epoch

4.1 Test Results

AlexNet, VGG16, VGG19 and GoogleNet networks converge quickly by scoring over 0,90 accuracy at the first epoch with 350 training samples while ResNet family achieved poorer scores at first iteration with the smallest training dataset. At second epoch all networks scored over 0,9. However, it is observed that higher test scores are obtained from higher epochs and larger datasets.

All networks benefitted from larger datasets achieving the best test score with more than 14K training samples. While ResNet50 and ResNet152 achieved best scores with 14K training set, VGG16, GoogleNet, and ResNet101 obtained best scores with 21K training dataset. Yet, the accuracy differences between results obtained with 14K, 21K, and 28K are barely noticeable and it is not possible to make an inference regarding the performance comparison of networks by solely inspecting maximum scores for test 1 which is based on the images similar to training dataset. Following test cases, which shows diversity in terms of illumination, camera orientation and distance with respect to the surface and material, are conducted to examine whether the networks overfit or prominent to learn generic crack features for further cases.

As can be seen from Figure 2, crack images on pavements are visually more discernable due to homogeneous illumination conditions and high contrast between cracks and background textures with respect to building application. As a result, all networks except AlexNet scored higher or similar scores with respect to third and fourth test cases. Low scores of ResNet101 and ResNet152 networks show that these networks are subjected to overfitting even with 350 training images. While both networks scored 0,99 accuracy in test 1 case, the maximum score obtained for test case 2 is below 0,8. The mismatch between the networks showing best performance for accuracy and F-score metrics also indicate that the networks are not stable.

Also, ResNet50 achieved the highest score with smallest training size and first epoch showing tendency to overfit. This tendency is evaluated as a mismatch between ResNet’s high capability of classifying objects with high-level features and cracks’ low-level features.

GoogleNet and VGG19 networks achieve over 0.95 accuracies with relatively limited training data. For VGG16, even though the obtained highest score is with the 21K dataset, it achieved 0.96 accuracy with 0,35K dataset at first and second epochs which are also comparable with successful counterparts.

Similar to the pavement case study, VGG networks, and GoogleNet were able to transfer learned features to building case scoring over 0,92 accuracy with at most 3,5K training dataset. While VGG networks are barely affected by the variations in illumination, background texture and camera orientation with respect to the surface, GoogleNet is subjected to 0,06 performance loss. On the other hand, ResNet networks show overfitting with decreasing scores regardless of the size of training data and number of epochs.

Brickwork images are relatively the most challenging case among the four test cases as brickwork jointing and background textures are challenging noises. Among the tested cases, VGG16 and GoogleNet achieved more than 0,90 accuracy. Especially 0,96 accuracy performance of VGG16 is promising in terms of achieving a generic crack detection framework regardless of material with limited dataset size.

4.2 Discussions

Regardless of the test case and utilized network, the performance of training datasets with fewer samples are comparable to counterparts with a high number of samples. While obtaining test 1 accuracy with the highest number of training samples, training datasets with 3,5K were sufficient for obtaining the best scores for other test cases. One exception can be given as the ResNet family performance for Test 2, Test 3, and Test 4 where the performance scores significantly drop indicating overfitting. In the case of the training data and test data being similar, size of training dataset positively influences accuracy. On the other hand, when the networks are used for varying cases in terms of illumination or spatial relations between camera and surface, then the increasing the size of dataset pose a risk of overfitting. This analysis is also valid for training epochs. As the number of epoch for training increase, accuracy for test data with similar conditions increases while for diverse test cases, the networks have a tendency to overfit or have a bias towards the training dataset with the increased number of epochs. For future studies, it is advised to start with few hundreds of images per class for training and gradually increase the number of training samples until overfitting is observed. It is also noted that the level of variance in the dataset is more important than the number of samples. Yet, variance among the training dataset is highly case specific and should be evaluated with respect to representation level of real-life cases.

Regarding the influence of network complexity in terms of the number of convolution networks and learnable parameters, it is observed that the influence of the number of convolution layers is more dominant than the number of learnable parameters. Best performing pretrained networks have 16-22 convolution layers (VGG16, VGG19, and GoogleNet). While AlexNet with 8 convolutional layers has difficulties in transferring learned features to new cases, ResNet networks with more than 50 convolutional layers have a tendency to overfit the training data. It should be noted that the layer configuration (GoogleNet having inception module and

ResNet being based on residual units) is disregarded within the scope of this study. In order to examine the influence of different layer configuration ResNet family networks are required to be truncated to obtain the same number of convolution layers. On the other hand, similar layer configuration of AlexNet, VGG16, and VGG19 shows that increased number of convolution layers contributes to the performance of networks in crack detection task for VGG16 and VGG19 compared to AlexNet. Another deduction can be made considering the number of fully connected layers. Simple CNN's having hierarchical layer connections (AlexNet and VGG networks) have shown resilience in varying test cases while DAG networks (GoogleNet and ResNet networks) have difficulty in transferring learned features to new cases. Even though the number of fully connected layers is not the only reason for the performance drop, hierarchical networks have proven themselves to adapt to new cases.

The number of learnable parameters, on the other hand, does not have a direct relationship with the performance but plays a significant role in computation time. GoogleNet, having almost five percent of VGG parameters scored a similar score. This can be linked to the low number of features defining cracks.

The computational time required to train 28K dataset per epoch for all networks are shared in Table 4.

Table 4. Computational time for training 28K dataset per epoch

28K dataset | per Epoch	Training Time (s)
AlexNet	133
VGG16	2827
VGG19	2943
GoogleNet	1227
ResNet50	1666
ResNet101	2447
ResNet152	3789

It should be noted that the majority of the learnable parameters are used by the fully connected layers of VGG networks. Hence, the trade-off between computational time and performance emerges as a trade-off between the number of fully connected layers and training time. On the other hand, it is possible to limit the number of filters to reduce the number of learnable parameters, thus the computational time while conserving the number of convolution layers for developers constructing the network from scratch.

5 Conclusion

Within the scope of the study, the performance of highly acknowledged pretrained networks on crack detection task is evaluated for buildings. The relations between training dataset size, number of epochs for training, number of CNN layers and learnable parameters are thoroughly investigated. It is shown that the pretrained networks can be fine-tuned for crack classification task with a limited number of training samples when the variance among data is provided or the test case constitutes images similar to the training samples. In the absence of variance among training images, increasing number of image samples not only contributes to the computational time without enhancing performance but also increases the risk of overfitting as the number of images with similar features analyzed per epoch increases. In the case of test case being visually incompatible with respect to training samples, it is advised to train the network with a limited number of samples and observe the tendency to overfit with the increasing number of training dataset size.

Regarding the effect of the number of convolutional layers to accuracy, even though the study does not involve a grid search for an optimum number of layers for the task, networks with 16 to 22 convolutional layers scored highest compared to both AlexNet with 8 convolutional layers and ResNet networks with more than 50 layers. In addition, networks with hierarchical layer connections and multi fully connected layers are observed to perform better in varying conditions and show promise in the course of achieving a generic crack detection framework regardless of material.

In conclusion, the pretrained networks have high applicability on crack detection even if they are trained on completely different datasets due to the low-level features shared with cracks and any objects with more abstract features. It is observed that the features learned in the course of training are transferable to other materials with high accuracy. In addition, the required number of less training samples and fast convergence networks make pretrained networks a favorable option for implementing CNN's for crack detection task.

References

[1] Bianchini, A., Bandini, P. & Smith, D.W. Interrater reliability of manual pavement distress evaluations. *Journal of Transportation Engineering*, 136 (2), 165-172, 2010

[2] Adhikari, R.S., Moselhi, O., Bagchi, A. Image-based retrieval of Concrete Crack Properties, *Automation in Construction*, 39(1), 180-194, 2014.

[3] Wang, K. C. P., Zhang, A., Li, J. Q., Fei, Y., Chen, C. and Li, B. Deep Learning for Asphalt Pavement Cracking Recognition Using Convolutional Neural Network In *International Conference on Highway Pavements and Airfield Technology 2017*, pages 166–177, Chicago, USA, 2017.

[4] Wu, W., Liu, Z., and He, Y. Classification of

defects with ensemble methods in the automated visual inspection of sewer pipes. *Pattern Analysis and Applications*, 18(2), 263–276, 2015.

[5] Sinha, S. K. and Fieguth, P. W. Neuro-fuzzy network for the classification of buried pipe defects. *Automation in Construction*, 15(1), 73–83, 2006

[6] Zhang, L., Yang, F., Zhang Y. D. and Zhu, Y. J. Road Crack Detection Using Deep Convolutional Neural Network In *2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, USA, 2016.

[7] Eisenbach, M., Stricker, R., Debes K. and Gross, H.M. Crack Detection with an Interactive and Adaptive Video Inspection System In *Arbeitsgruppentagung Infrastrukturmanagement*, pages 94–103, 2017.

[8] Pauly, L., Peel H., Luo, S., Hogg, D. and Fuentes, R. Deeper Networks for Pavement Crack Detection In *Proceedings of the 34th ISARC. 34th International Symposium in Automation and Robotics in Construction*, pages 479–485 Taipei, Taiwan, 2017.

[9] Cha, Y.J., Choi, W. and Büyüköztürk O., Deep Learning Based Crack Damage Detection Using Convolutional Neural Networks, *Computer-Aided Civil and Infrastructure Engineering*, 32(5): 361–378, 2017.

[10] Krizhevsky, A., Sutskever, I. and Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks In *Advances In Neural Information Processing Systems25 (NIPS 2012)*, pages 1097-1105, Nevada, USA, 2012.

[11] Simonyan K. and Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition In *International Conference on Learning Representations (ICRL)*, pages 1–14, Vancouver, Canada, 2015.

[12] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. Going deeper with convolutions In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1-9, Boston, USA, 2015.

[13] He, K., Zhang, X., Ren, S. and Sun, J. Deep Residual Learning for Image Recognition In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Seattle, USA 2016.

[14] Gopalakrishnan, K., Khaitan, S. K., Choudhary, A. and Agrawal, A. Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection *Construction and Building Materials*, 157(September):322–330, 2017.

[15] Özgenel, Ç. F. Concrete Crack Images for Classification, Mendeley Data, v1, 2017.

[16] MatConvNet Team, Pretrained CNN's - MatConvNet. Online: http://www.vlfeat.org/matconvnet/pretrained/ , Accessed: 13.01.2018