Computer Assisted Methods in Engineering and Science, **26**: 105–112, 2019, doi: 10.24423/cames.267

IPM 2019

A comparison of deep convolutional neural networks for image-based detection of concrete surface cracks

Marek Słoński

Cracow University of Technology
Warszawska 24, 31-155 Kraków, Poland
e-mail: mslonski@L5.pk.edu.pl

The aim of this paper is to compare the performance of four deep convolutional neural networks in the problem of image-based automated detection of concrete surface cracks in the case of a small dataset. This crack detection problem is treated as a binary classification problem, and it is solved by training a deep convolutional neural network on the small dataset. In this context, overfitting during training was the main issue to cope with and various techniques were applied to overcome this issue. The results of the experiments suggest that the best approach for this problem is to use the pretrained convolutional base of a large pretrained convolutional neural network as an automatic feature extraction method and adding a new binary classifier on top of the convolutional base. Then, at the training the new classifier and fine-tuning the last few layers of the pretrained network take place at the same time. The classification accuracy of the best deep convolutional neural network on the testing set is about 94%.

Keywords: deep convolutional neural network, pretrained network, image based crack detection, binary classification, overfitting.

1. Introduction

Automatic image-based detection of surface defects such as concrete cracks is an important part of the vision-based structural health monitoring systems and condition assessment of civil infrastructure. Surface defects may indicate severe degradation processes being in progress in a structure. To this end, various image-based detection strategies of surface defects using computer vision and image processing methods were proposed [11, 16].

Currently, the most promising methods for fully automatic image-based defects detection are built with deep convolutional neural networks (DCNN or ConvNet) [18, 21]. For example, DCNNs were applied for automated computer vision-based pavement distress detection [7]. DCNNs were also successfully applied for crack damage detection obtaining accuracy close to 97% on testing set [1].

The main obstacle in the successful application of DCNNs for such problems is the lack of massive training datasets which are necessary to train large DCNNs from scratch. However, in the case of a small dataset it is still possible to build a reliable classifier and obtain the best results. It can be achieved by using a large ConvNet pretrained on a massive dataset from a similar domain.

In this paper, we show the comparison of convolutional neural network architectures and training methods for detecting cracks in concrete pavements in the case of a small dataset. Similar studies were presented in [5, 17, 19]. The comparison is based on the public benchmark dataset SDNET2018 [4]. The authors of the SDNET2018 image dataset applied a pretrained DCNN called AlexNet [12] for concrete crack detection in fully trained (FT) and transfer learning (TL) modes using the computational setup and procedure described in [3]. The obtained results confirmed that DCNNs could be accurate in predicting the presence of a crack in the specimen. The AlexNet was also used for detecting a crack in [9, 10].

The rest of the paper is structured as follows. In Sec. 2, the problem of image-based detection of concrete cracks is presented. Section 3 contains a short overview of the basic deep convolutional neural network architecture and the training methods. Section 4 describes the experimental study, and Sec. 5 presents the results of experiments. The final Sec. 6 concludes the paper with some findings from the analysis.

2. Image-based detection of concrete cracks

The problem of image-based automatic detection of concrete cracks can be formulated as a binary classification problem and solved by building a classifier based on logistic regression, neural network or support vector machine. The first step in building such a classifier is to collect a large number of images of materials or structural parts containing cracks and the corresponding set of images without cracks. The second step is to build the classification model using one part of the whole dataset called the training dataset and validate the model using another part called the validation dataset. Then we can apply the trained model to classify new images.

In this work, we use deep convolutional neural networks for building an image-based classifier of concrete cracks. A schematic diagram illustrating this approach is shown in Fig. 1.

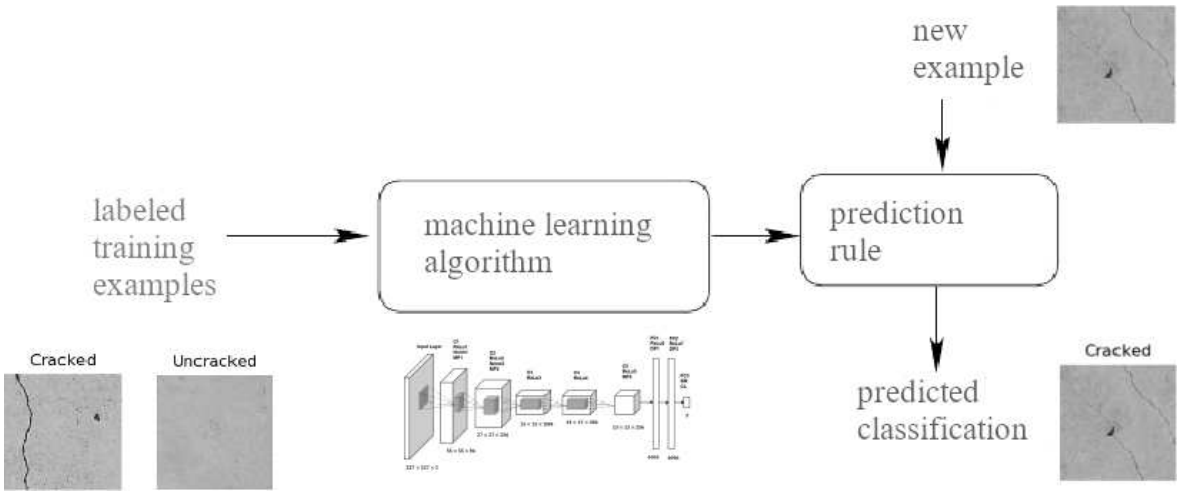

Fig. 1. Diagram for image-based detection of concrete cracks using DCNN-based classifier.

3. Deep convolutional neural network

A deep convolutional neural network (DCNN or ConvNet) is a special class of deep neural networks designed for processing inputs that have a known grid-like structure: audio signals (1D) images (2D) and videos (3D) [6, 13, 14]. A typical ConvNet consists of three types of layers: convolutional, pooling and fully-connected. These layers are stacked to form a ConvNet architecture. The aim of the first convolutional layer is to detect local features such as horizontal, vertical and diagonal edges in input images [2, 15].

ConvNets are usually trained using a combination of the backpropagation algorithm for computing the gradient of the loss function and the mini-batch stochastic gradient descent method (SGD) with some improvements such as momentum for iteratively updating the weights. The main problem during the training of large neural networks with many parameters on small datasets is overfitting. In the case of training DCNNs for image classification, data augmentation is a common technique to cope with overfitting. This method allows to artificially expand the training dataset via a number of random modifications such as rotation and flipping that yield images visually

similar [2]. Another technique is a dropout, which is a regularization technique to improve training process performance [22].

3.1. Pretrained network

A pretrained network is a network in which parameters were trained on a large training set from one domain. There are many commonly used pretrained DCNN models that can be used in other problems by applying transfer learning. Transfer learning is a method used to adapt the pretrained model to another dataset. It is often used with fine-tuning, which is a technique to improve the performance of the network by adapting a few layers of the convolutional base.

4. Experimental study

In this work, we compare four strategies for building a binary classifier for detecting cracks in images with a deep convolutional neural network (ConvNet):

1) a small ConvNet built from scratch,

2) a small ConvNet built from scratch with data augmentation,

3) a large pretrained ConvNet with data augmentation,

4) a large pretrained ConvNet with data augmentation and fine-tuning.

In the first strategy, we build a new classifier from scratch based on a small ConvNet without any regularization. Because for this model the main issue is overfitting, in the second strategy we try to overcome this problem by applying the data augmentation (DA) technique. In the third strategy, we use a convolutional base of a large pretrained ConvNet and a trained with DA binary classifier set on the top of the convolutional base. In the fourth strategy, we use the same model as in the third strategy, and we also apply fine-tuning (FT) of a few convolutional layers to improve the performance of the classifier. In the experiments, we apply a pretrained ConvNet called VGG16 described in short in the next section. Table 1 contains the basic parameters such as number of layers, total number of parameters and the size of the considered ConvNets.

Table 1. Basic parameters of the applied ConvNets.

ConvNet model	No. of conv. layers	No. of layers	No. of parameters
Small, from scratch	4	11	1 043 905
The convolutional base of VGG16	13	18	14 714 688
Large, pretrained VGG16	13	23	15 763 521

4.1. VGG16

VGG16 is a deep convolutional neural network proposed in 2014 by K. Simonyan and A. Zisserman from Visual Geometry Group at the University of Oxford [20]. It was trained on more than a million images from the ImageNet database, which is a database of over 14 million images belonging to 1000 classes, designed for use in visual object recognition software research. VGG16 was the winner of the ImageNet Challenge in 2014. It consists of 16 layers, including 13 convolutional layers with a filter size of 3×3. The VGG16 architecture is presented in Fig. 2.

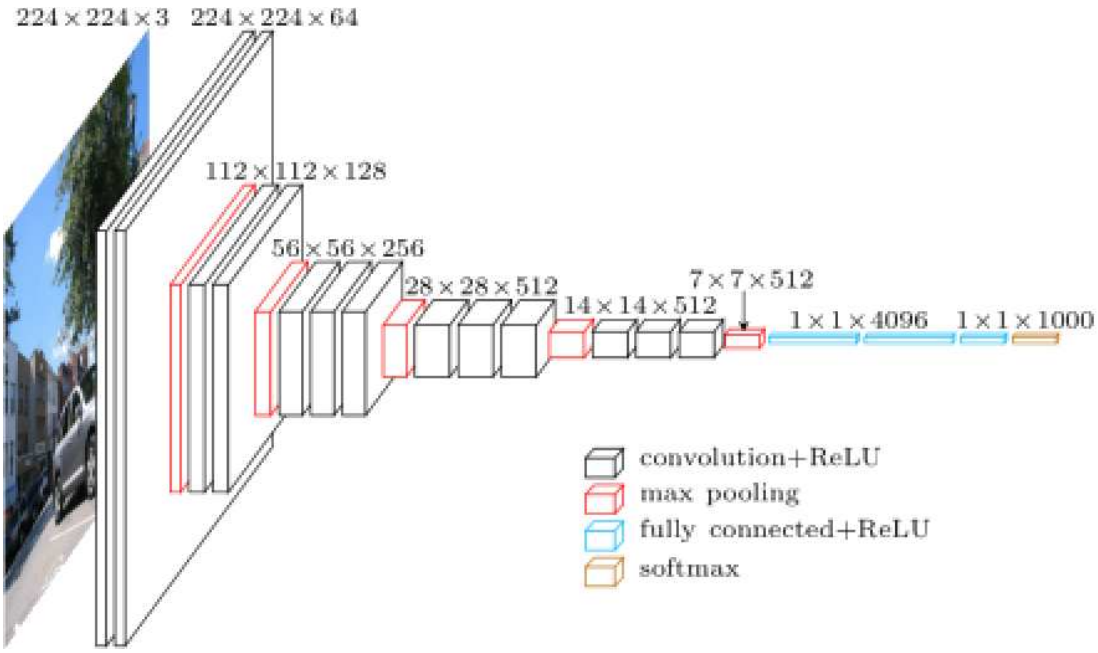

Fig. 2. The VGG16 architecture, taken from [8].

4.2. Dataset

The four ConvNets were built using images from the public dataset SDNET2018 [4]. The dataset contains more than 56 000 annotated small images of cracked and non-cracked concrete bridge decks, walls, and pavements. Images are 256 × 256-px RGB files stored in .jpg format. Few images of cracked and uncracked concrete pavements from the benchmark dataset are shown in Fig. 3. As can be seen, some cracks are hardly visible in the images so they might be misclassified by the classifier.

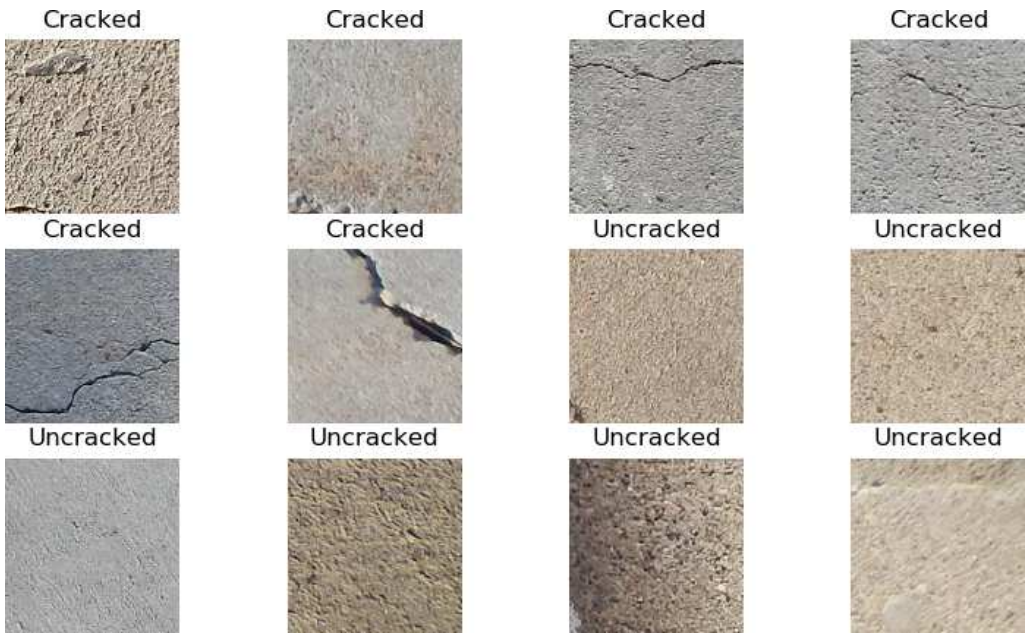

Fig. 3. Images of cracked and uncracked concrete pavements, sampled from SDNET2018 dataset.

4.3. Experimental setup

In this paper, we consider the case of a small dataset containing only a few thousands of images of pavements with crack and without crack and a balanced set of 5200 samples of images was prepared. In Table 2, details of the training, validation and testing datasets are given.

Table 2. Details of the training, validation and testing datasets.

	With crack	Without crack	Total
Number of training samples	1500	1500	3000
Number of validation samples	750	750	1500
Number of testing samples	350	350	700
Total number of samples	2600	2600	5200

The experiments were programmed in Python with Keras which is an open-source neural-network library running on top of TensorFlow. TensorFlow is an open-source machine learning library for research and production developed by Google [2].

All experiments were performed on a Dell Inspiron 15 laptop computer with 64-bit operating system Windows 10, 32 GB RAM memory, Quad-Core Intel Core i7 processor and NVIDIA GeForce GTX 1060 Ti (4 GB) graphics processing unit (GPU).

5. Results and discussion

In this section, the comparison of the performance of the four deep convolutional neural networks for image-based crack detection is presented.

5.1. Training and validation

In Fig. 4, the comparison of training and validation losses during a training within 100 epochs for four DCNNs considered in the paper is presented. It can be seen that the minimal training loss was achieved for the small ConvNet built from scratch without DA, but at the same time this ConvNet achieved the maximal validation loss, which is the result of overfitting. By applying the DA technique during the training, the problem of overfitting was resolved, but the overall performance was still low as can be seen in Table 3.

Table 3. The lowest training and validation losses, the corresponding epoch and training time for one epoch.

ConvNet model	Training loss	Epoch	Validation loss	Epoch	Time [s]
Small, from scratch	0.09	100	0.38	52	7
Small, from scratch with DA	0.26	100	0.38	63	17
Large, pretrained VGG16 with DA	0.24	100	0.30	72	21
Large, pretrained with DA and FT	0.12	100	0.20	82	22

The minimal validation loss was achieved for the large pretrained network with DA and FT. By applying FT we were able to lower the validation loss by more than 30% with respect to the same model without FT. In Table 3, the lowest training and validation losses and the corresponding epochs are given.

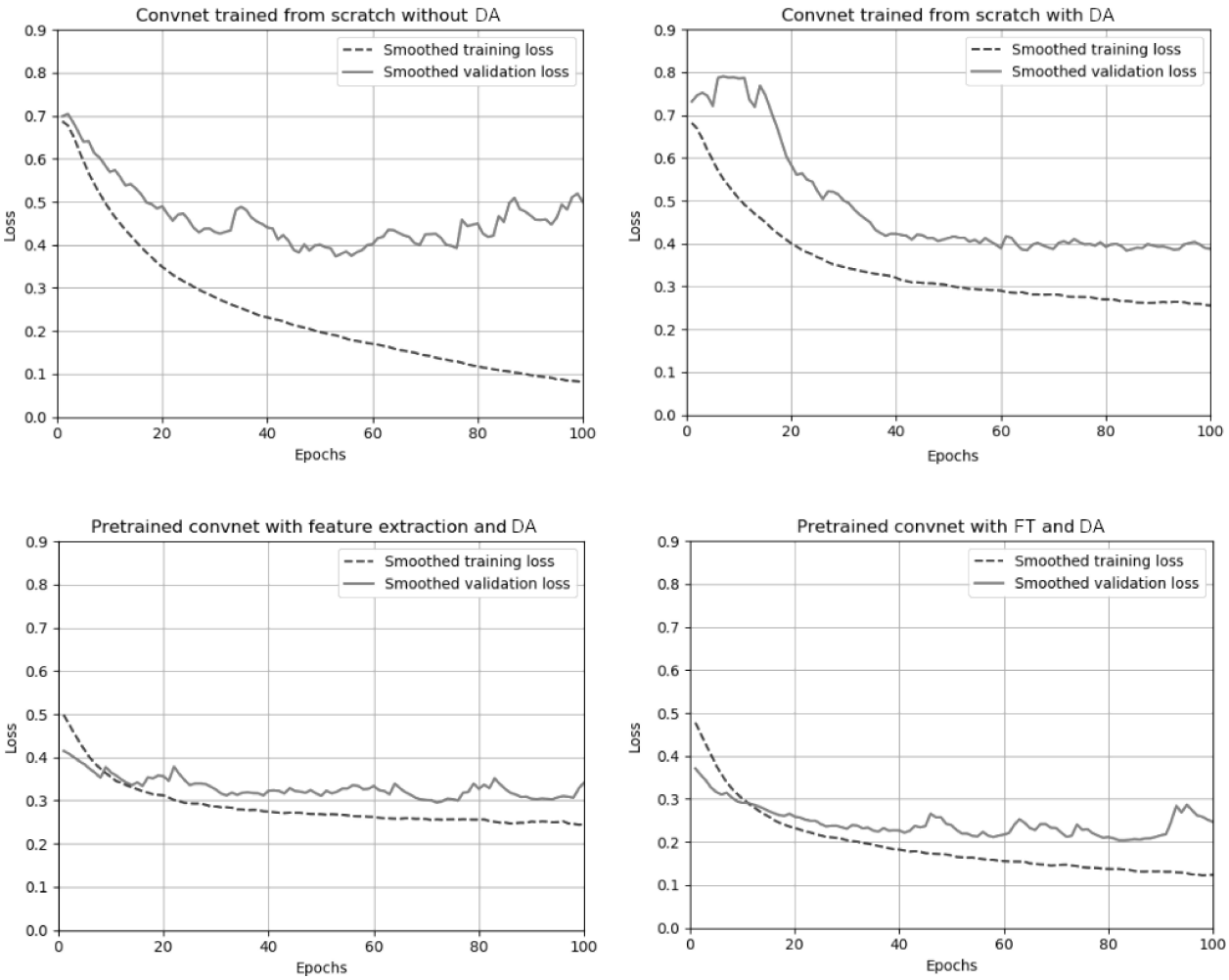

Fig. 4. Comparison of training and validation losses for four ConvNets: top left – a small ConvNet built from scratch without DA; top right – a small ConvNet built from scratch with DA; bottom left – a large pretrained ConvNet with DA and without FT; bottom right – a large pretrained ConvNet with DA and with FT.

In Fig. 5, the comparison of corresponding training and validation accuracies during the training within 100 epochs for four DCNNs considered in the paper is presented. It can be seen that the best results obtained in terms of accuracy are similar to the corresponding results in terms of loss. The training accuracy after 100 epochs changes slightly between 92% and 98%, while the final validation accuracy varies between 82% and 93%. In Table 4, the highest training and validation accuracies and the corresponding epochs are given. The table also contains the times needed to perform one epoch of training. The longest training time is for the case of the large pretrained ConvNet with DA and FT (22 seconds per epoch).

Table 4. The highest training and validation accuracies, the corresponding epoch and training time for one epoch.

ConvNet model	Training acc.	Epoch	Validation acc.	Epoch	Time [s]
Small, from scratch	98%	100	85%	78	7
Small, from scratch with DA	90%	100	86%	84	17
Large, pretrained VGG16 with DA	89%	100	88%	95	21
Large, pretrained with DA and FT	95%	100	93%	95	22

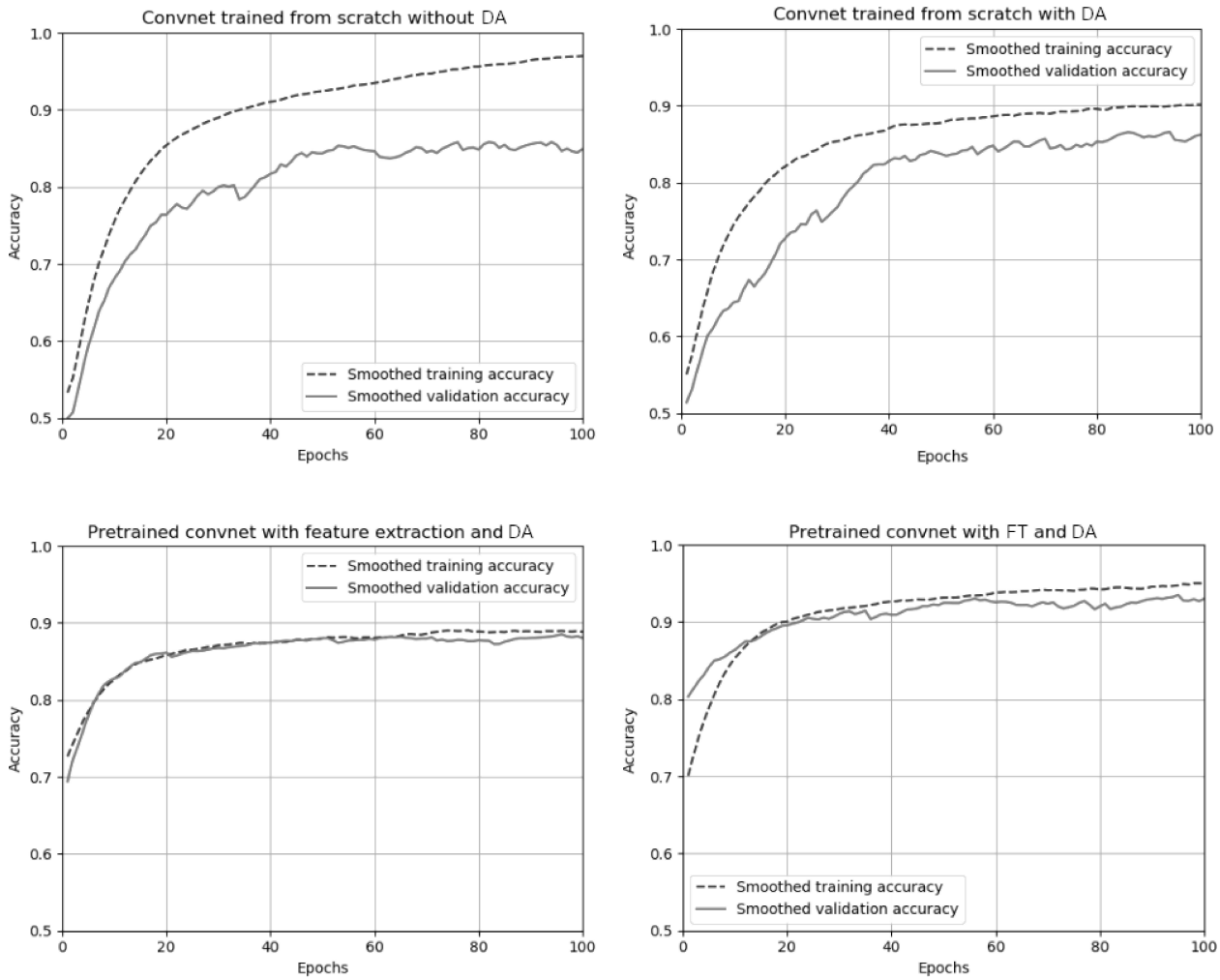

Fig. 5. Comparison of training and validation accuracies for each epoch for four ConvNets: top left – a small ConvNet built from scratch without DA; top right – a small ConvNet built from scratch with DA; bottom left – a large pretrained ConvNet with DA and without FT; bottom right – a large pretrained ConvNet with DA and with FT.

5.2. Testing results

Finally, the best ConvNet, with respect to validation accuracy, was checked using the testing set. The generalization accuracy for this ConvNet was close to 94%, which confirms that the best strategy for this problem was to apply the pretrained large network (VGG16) with FT.

6. Final remarks

In this paper, we compared the performance of four different ConvNets for the problem of image-based detection of concrete surface cracks. We compared a small ConvNet trained from scratch without and with DA and a large pretrained ConvNet with DA and without and with FT. In the computational experiments, we used the public benchmark dataset SDNET2018. Based on the results of the experiments, it can be stated that the best validation accuracy has the large pretrained ConvNet with DA and with FT, but more experiments are needed to fully exploit the capabilities of DCNNs for this task.

References

[1] Y.-J. Cha, W. Choi, O. Büyüköztürk. Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, **32**(5): 361–378, 2017.

[2] F. Chollet. *Deep Learning with Python*. Manning Publications Co., 2018.

[3] S. Dorafshan, R.J. Thomas, M. Maguire. Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Construction and Building Materials*, **186**: 1031–1045, 2018.

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

[5] C.V. Dung, L.D. Anh. Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction*, **99**: 52–58, 2019.

[6] I. Goodfellow, Y. Bengio, A. Courville. *Deep Learning*. MIT Press, 2016.

[7] K. Gopalakrishnan. Deep learning in data-driven pavement image analysis and automated distress detection: A review. *Data*, **3**(3): 28, 2018.

[8] L. Blier. A brief report of the Heuritech Deep Learing Meetup #5. *hueritech Le Blog*, 29.02.2016, https://blog.heuritech.com/2016/02/29/a-brief-report-of-the-heuritech-deep-learning-meetup-5/, accessed on 15.02.2019.

[9] B. Kim, S. Cho. Automated vision-based detection of cracks on concrete surfaces using a deep learning technique. *Sensors*, **18**(10), 2018.

[10] H. Kim, E. Ahn, M. Shin, S.-H. Sim. Crack and noncrack classification from concrete surface images using machine learning. *Structural Health Monitoring*, **18**(3): 725–738, 2019.

[11] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, P. Fieguth. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Advanced Engineering Informatics*, **29**(2): 196–210, 2015.

[12] A. Krizhevsky, I. Sutskever, G.E. Hinton. ImageNet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[13] Y. LeCun. Generalization and network design strategies. In: R. Pfeifer, Z. Schreter, F. Fogelman, L. Steels [Eds], *Connectionism in perspective*. Elsevier, 1989.

[14] Y. LeCun, Y. Bengio, G. Hinton. Deep learning. *Nature*, **521**(7553): 436, 2015.

[15] S. Mallat. Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society of London Series A*, **374**, 16 pages, 2016.

[16] A. Mohan, S. Poobal. Crack detection using image processing: A critical review and analysis. *Alexandria Engineering Journal*, **57**(2): 787–798, 2018.

[17] Ç.F. Özgenel, A.G. Sorguç. Performance comparison of pretrained convolutional neural networks on crack detection in buildings. In: P*roceedings of the International Symposium on Automation and Robotics in Construction*, Vol. 35, pp. 1–8. IAARC Publications, 2018.

[18] W. Rawat, Z. Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, **29**(9): 2352–2449, 2017.

[19] W.R.L. Silva, D.S. Lucena. Concrete cracks detection based on deep learning image classification. In: *Multidisciplinary Digital Publishing Institute Proceedings*, Vol. 2, p. 489, 2018.

[20] K. Simonyan, A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[21] B.F. Spencer, V. Hoskere, Y. Narazaki. Advances in computer vision-based civil infrastructure inspection and monitoring. *Engineering*, **5**(2): 199–222, 2019.

[22] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, **15**(1): 1929–1958, 2014.