Evaluation of Transfer Learning for Visual Road Condition Assessment

Christoph Balada[1,2], Markus Eisenbach[2(✉)], and Horst-Michael Gross[2]

[1] German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
christoph.balada@dfki.de

[2] Neuroinformatics and Cognitive Robotics Lab, Ilmenau University of Technology, Ilmenau, Germany
markus.eisenbach@tu-ilmenau.de

Abstract. Through deep learning, major advances have been made in the field of visual road condition assessment in recent years. However, many approaches train from scratch and avoid transfer learning due to the different nature of road surface data and the ImageNet dataset, which is commonly used for pre-training neural networks for visual recognition. We show that, despite the huge differences in the data, transfer learning outperforms training from scratch in terms of generalization. In extensive experiments, we explore the underlying cause by examining various transfer learning effects. For our experiments, we are incorporating seven known architectures. Therefore, this is the first comprehensive study of transfer learning in the field of visual road condition assessment.

Keywords: Transfer learning · Road condition assessment · Surface distress detection

1 Introduction

Aging public roads need frequent inspections in order to guarantee their permanent availability. In many countries, this includes the standardized visual assessment of millions of images. Formerly, due to the lack of sophisticated approaches, the evaluation was typically done manually by human experts. Since large, annotated road surface image datasets, like the German asphalt pavement distress (GAPs) dataset [12], have been published recently, automated visual road surface analysis by machine learning approaches came into focus. In recent years, deep learning approaches dominated this field of application, e.g. for detecting cracks [2,3,6,10,16,22,23,27,32,37,39–41], potholes [7,25], or multiple types of surface distress simultaneously [1,9,11,13,24,29,30,36]. However, most of these

This work has received funding from the German Federal Ministry of Education and Research as part of the ASINVOS project under grant agreement no. 01IS15036 and as part of the FiN-2.0 project under grant agreement no. 03SF0568G, as well as from the Austrian Research Promotion Agency (FFG) as part of the ASFaLT project under grant agreement no. 86951

Int. Conf on Artificial Neural Networks (ICANN), Bratislava, pp. 540-551, Springer 2021

approaches design their own neural network, or train well known architectures from scratch, mainly justified by big differences of road images to the ImageNet dataset, which is typically used for pre-training on visual data. For a description of these methods, we refer to the surveys [5] and [15].

In this paper, we perform extensive experiments to analyze transfer learning for visual road surface data analysis. In contrast to related work on transfer learning in road surface analysis,

- we include more modern architectures in our analysis,
- we perform extensive hyper-parameter tuning for each experiment in order to ensure a fair comparison,
- we evaluate how a changed input encoding does effect transfer learning,
- we evaluate the impact of freezing different proportions of the layers, and
- we analyze the effects of transfer learning in comparison to trainings from scratch.

Consequently, in this paper we want to answer the following three questions: How much will visual road condition assessment be improved by applying transfer learning? What are the improvements achieved by? Which transfer learning effects known from literature (see Sect. 2.2) do occur in our setting?

2 Related Work

Due to frequent inspections, many road surface images are available. All images are analyzed by human experts. However, since results are needed in a timely manner, in most cases the labeling is very coarse. Therefore, detailed annotations are only available for a very small percentage of these data.

Since detailed labeling by experts is expensive, Seichter *et al.* [31] proposed an approach where unlabeled data are analyzed regarding uncertainty of a trained classifier. Thus, data worth annotating can be identified.

Even by increasing the percentage of annotations to a certain extent by applying this method, the amount of available annotated data is still very limited. Thus, the purpose of our work was to ensure a good generalization on the few training data by applying transfer learning.

2.1 Transfer Learning for Road Condition Assessment

In the following, we briefly analyze approaches, which utilize transfer learning on road surface data or related applications, i.e. all kind of damage detection of public infrastructure.

In [32] and [4], transfer learning was applied to a VGG16 and a ResNet-152 model, which were fine tuned on very few training samples without freezing layers. Both studies yield only mediocre results due to a limited amount of training data and too many parameters to be tuned. Also having only few samples, in [16,25], and [14], transfer learning was applied to a VGG16, an XceptionNet, and an InceptionNet V3 model, while the weights of all layers but the last one

Int. Conf on Artificial Neural Networks (ICANN), Bratislava, pp. 540-551, Springer 2021

was frozen. This led to better results. Zhang *et al.* [40], were able to fine tune an AlexNet with only the weights of the first layer being frozen since they had a larger dataset available in comparison to the studies above.

None of the aforementioned papers analyze the impact of transfer learning, nor do they analyze the influence of any hyperparameters, like the amount of frozen weights or the learning rate. The only paper analyzing transfer learning in more detail in a related application is [28], that detected cracks in buildings. The authors compared the generalization ability of seven architectures for a varying number of training samples. During fine tuning none of the weights were frozen. The authors did not compare their approach against training from scratch and did not tune any hyperparameters. In particular, the learning rate has a considerable influence on the generalization abilities. Therefore, the findings of that study should be treated with caution.

2.2 Transfer Learning Effects

When applying transfer learning, two major improvements are reported in literature: Faster training and better generalization ability (e.g. in the comprehensive survey of Zhuang *et al.* [42]).

Better Generalization. So far, generalization improvements by applying transfer learning for road pavement distress detection has not been analyzed. Therefore, we conducted experiments to analyze this issue. As reported in [38], initializing a model with pre-trained weights yields remarkable improvements in generalization. The improved generalization ability can be observed by a decreasing gap between validation and test performance, which we will use as criterion.

Faster Training. On road pavement data, training with transferred ImageNet weights as initialization is reported to converge within ten epochs [16]. But in that study, all weights except the ones of the classification layer were frozen. No experiments regarded convergence improvements by transfer learning, and the effect of freezing weights were not investigated. Therefore, we analyzed the convergence speed when all weights are adapted and compared to trainings where different proportions of weights were frozen. Additionally, we compared results from transfer learning with results achieved by trainings from scratch. We evaluated whether the required training time reduces to a fraction of epochs in comparison to training from-scratch, which is often the case in other fields of application.

Feature Adaption and Selection. Kim *et al.* [20] analyzed transfer learning effects in an application where material defects should be detected in microscope images. They reported that mainly due to the fact that early layers in neural networks tend to provide features which focus on simple structures, like edges or brightness changes, they require less adaption to new datasets, even if they are very different from ImageNet. This was also reported in [38]. Furthermore, Kim *et al.* [20] showed that during fine tuning of the pre-trained weights, a kind of feature selection takes place instead of learning completely new features. Unfortunately, they examined only one architecture, namely VGG. To address this effect, we analyzed how much the pre-trained weights are changed in different architectures during the fine tuning step of transfer learning. We also analyzed whether a feature selection is observable when transfer learning is applied to road pavement data and in case of more modern neural network architectures.

3 Setup

In the following, we describe our experimental setting.

3.1 Training

For transfer learning, we used the weights from pre-trainings on the ImageNet dataset, which come with the models. We fine tuned on road surface data for 25 epochs, which is sufficient to ensure convergence, using SGD with mini batches of size 32 and momentum of 0.9. During training, no further learning rate scheduling was applied. To provide well founded results, we performed extensive experiments with many different hyperparameter combinations for each model.

Hyperparameter Search. Per architecture, we examined at least nine different learning rates and six different amounts of frozen layers in approximately 20% steps (0%, 20%, 40%, 60%, 80%, 100% excluding the fully connected layers). Since requirements are different for each architecture, we adapted the search range for the learning rate individually. Overall 426 trainings were performed.

Reference: Training from Scratch. We trained each architecture from scratch with randomly initialized weights for 250 epochs, which is sufficient to ensure convergence. The best out of the three runs was used as reference for comparisons. To ensure a fair comparison, a hyperparameter tuning regarding the learning rate was applied.

3.2 Dataset

As representative dataset for visual road condition assessment, we utilized the 50k binary classification set of the extended version of the German asphalt pavement distress dataset (GAPs v2) [34] that was suggested for experiments. While the complete GAPs v2 dataset yields ca. 6.7 M samples, the reduced set provides 50,000 training patches including 30,000 intact road patches and 20,000 damage samples composed of all types of surface distress. Additionally, 10,000 samples each are provided for validation, validation-test, and test. The four-way split was chosen as proposed by Ng [26]. Thus, validation data are used to find the best epoch, validation-test data for hyperparameter tuning, and test data only for the final evaluation. The validation set were taken from the same distribution as the training data. Validation-test and test data were recorded on roads that are geographically distinct from training and validation data. Different patch sizes are available, including 224×224 and 299×299, which are typical for inputs of ImageNet architectures. The label of the patch is based on the 64×64 image center, while the surroundings are needed as context. In [34], it has been found that the 50k subset represents the complete dataset with millions of patches very well. The results achieved on this subset are close to the results on the complete dataset. Therefore, as proposed in [34], we decided in favor of this subset in order to enable much more experiments.

3.3 Architectures

In our experiments, we analyzed the transfer-learning properties of seven widely used architectures, namely AlexNet [21], VGG19 [33], InceptionNet V3 [35], ResNet50 [17], XceptionNet [8], SE-ResNet50 [19], and MobileNet [18]. Table 1 summarizes their characteristics regarding input coding and model size.

All architectures are pre-trained on the ImageNet dataset. For VGG19, InceptptionNet V3, ResNet50, XceptionNet, and MobileNet, we used the models available in the Keras framework. For AlexNet and SE-ResNet50, we used publicly available implementations for Keras.

To address the differences in pre-processing shown in Table 1, we had to adapt the transferred weights regarding input size (224×224), channel count (1-channel grayscale) and input scaling ($[-1, 1]$).

3.4 Evaluation Metrics

Performance Measures. After each training epoch, we computed accuracy, F_1 score, and balanced error rate on the train, validation, and validation-test dataset. On the test dataset, we computed the metrics only once, based on the best epoch in terms of the validation-test performance.

Table 1. Characteristics of architectures included in our evaluation

Architecture	Zero-mean in terms of	Input scaling	Channel order	Input size	# Weights
MobileNet	Gray-world assum.	$[-1, 1]$	RGB	224^2	4,253,864
XceptionNet	Gray-world assum.	$[-1, 1]$	RGB	299^2	22,910,480
InceptionNet V3	Gray-world assum.	$[-1, 1]$	RGB	299^2	23,851,784
ResNet50	ImageNet dataset	$[0, 255]$	BGR	224^2	25,636,712
SE-ResNet50	Gray-world assum.	$[-1, 1]$	BGR	224^2	28,141,144
AlexNet	ImageNet dataset	$[0, 255]$	RGB	224^2	60,965,224
VGG19	ImageNet dataset	$[0, 255]$	BGR	224^2	143,667,240

Int. Conf on Artificial Neural Networks (ICANN), Bratislava, pp. 540-551, Springer 2021

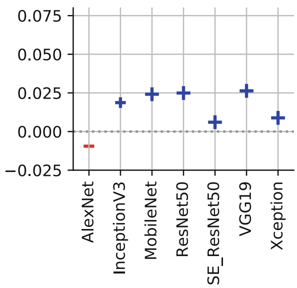
Additional Measures. We used the Euclidean distance of the weights before and after fine tuning and an activation-sparsity score to evaluate the appearance of a feature selection and the magnitude of weight changes during fine tuning as proposed in [20]. The activation-sparsity score counts zero or close-to-zero values with respect to a threshold for a given feature map.

4 Experimental Results

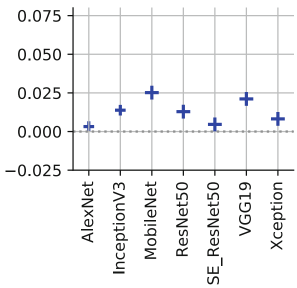
In the following, we compare the results achieved by transfer learning with trainings from scratch (Sect. 4.1). Based on these results, in Sect. 4.2, we analyze the transfer learning effects previously described in Sect. 2.2.

4.1 Transfer Learning Vs. Training from Scratch

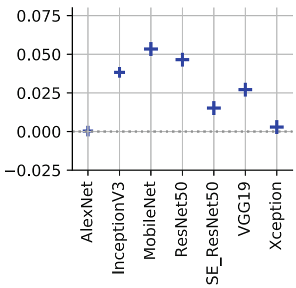
The performance gain of transfer learning for visual road condition assessment has not been analyzed in related work, yet. Therefore, in our extensive experiments, we analyzed multiple well known architectures with tuned hyperparameters (see Sect. 3.1). The best results for each architecture are shown in Fig. 1 and Table 2 for the validation, validation-test and test subset of the GAPs 50k dataset.



(a) Validation set



(b) Validation-test set



(c) Test set

Fig. 1. Absolute difference in F_1 score between a training from-scratch and fine tuning. + marks, where transfer learning performs better, – where it does not.

Figure 1 highlights the positive influence of transfer learning on the performance. Clearly, no architecture has any drawback from using transfer learning. Instead, we observe significant performance improvements for nearly every architecture and every subset of the GAPs 50k dataset. Most notably, the performance on the test dataset increased significantly, which shows the improvement in generalization.

Especially the InceptionNet V3 benefits from using transfer learning, as shown by the precision-recall curves in Fig. 2. Due to the improvement in generalization, it does even perform better on the test set than on the validation-test

Table 2. Comparison of the best results (F_1 score) for each architecture using transfer learning (TL) or training from scratch (FS), respectively.

	Validation		Validation-test		Test	
Architecture	TL	FS	TL	FS	TL	FS
InceptionNet V3	0.9441	0.9254	0.9024	0.8886	0.9143	0.8760
VGG19	0.9495	0.9231	0.9151	0.8940	0.9065	0.8794
ResNet50	0.9456	0.9206	0.9054	0.8925	0.8950	0.8485
MobileNet	0.9482	0.9241	0.9106	0.8854	0.8868	0.8334
XceptionNet	0.9473	0.9385	0.9155	0.9073	0.8842	0.8813
AlexNet	0.9089	0.9184	0.8895	0.8863	0.8604	0.8601
SE-ResNet50	0.9336	0.9276	0.8928	0.8881	0.8566	0.8414

set, which both contain road data geographically distinct from training data, but only the validation-test set was used for hyperparameter tuning. Additionally, on any subset of the GAPs 50k dataset, the transfer learning results are superior to the results achieved by training from scratch.

Convergence. For all architectures, the training converged within 15 epochs when transfer learning was applied. In comparison, training from scratch took considerably longer and converged within 150 epochs. In conclusion, transfer learning based on ImageNet pre-training speeds up the training significantly, even if the application is considerably different from ImageNet.

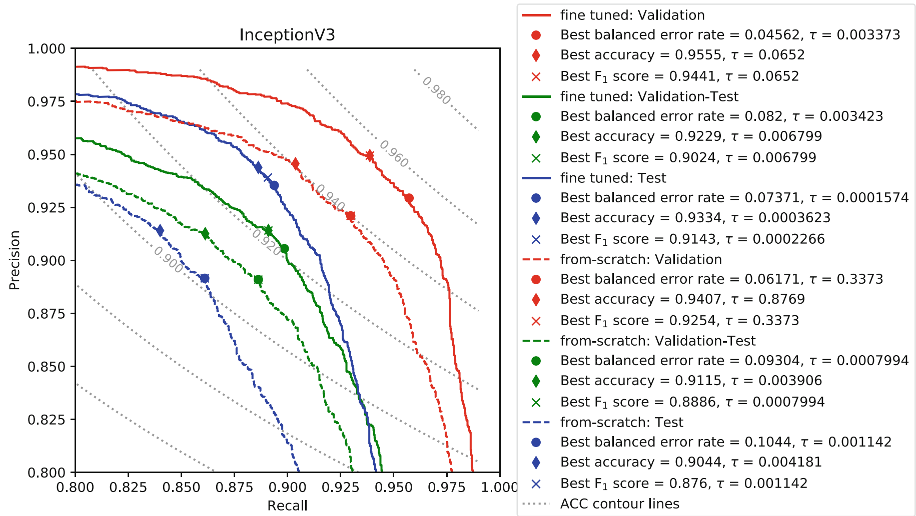


Fig. 2. Precision-recall curves for InceptionNet V3. Training from scratch is shown as dashed lines, transfer learning results as solid lines. The smaller gap between validation and test results for transfer learning (solid red and blue line) shows the better generalization. (Color figure online)

Hyperparameters. For each architecture, we performed a grid search to identify appropriate hyperparameters. The best learning rate for the individual architectures differed in a large range between 0.000355 and 0.1. Freezing layers turned out to not improve the performance of any architecture. Therefore, these experiments are omitted here.

4.2 Effects of Transfer Learning

According to literature, the following transfer learning effects should be observable: Feature selection in the final convolutional layer, only small weight changes in terms of Euclidean distance, and filters in early layers focus on simple features and can be re-used.

Feature Selection. Basically, a slight increase of the sparsity from input to output of the network can be observed for any architecture. Therefore, the final feature map of each architecture tends to be the one with the highest sparsity. Nonetheless, we found that the actual magnitude of sparsity highly depends on the specific architecture and the actual image sample which is passed through the network. Overall, we had no clear finding of an increase in terms of sparsity in the final feature map, regardless of the architecture, even though our application is significantly different from ImageNet as in [20].

We found that simple architectures like VGG19 or AlexNet tend to have a high sparsity (50% or more) in the final convolutional layers, and thus do only re-use some of the features. The activation-sparsity score counts zero or close-to-zero values with respect to a threshold for a given feature map. This high sparsity applies to the feature maps after transferring the weights as well as after fine tuning. In contrast, all modern architectures except InceptionNet V3 tend to have a sparsity of less than 10%. This means, more than 90% of the features in these modern architectures are activated regardless of the input, and therefore they re-use or re-combine most of the transferred features. Furthermore, while

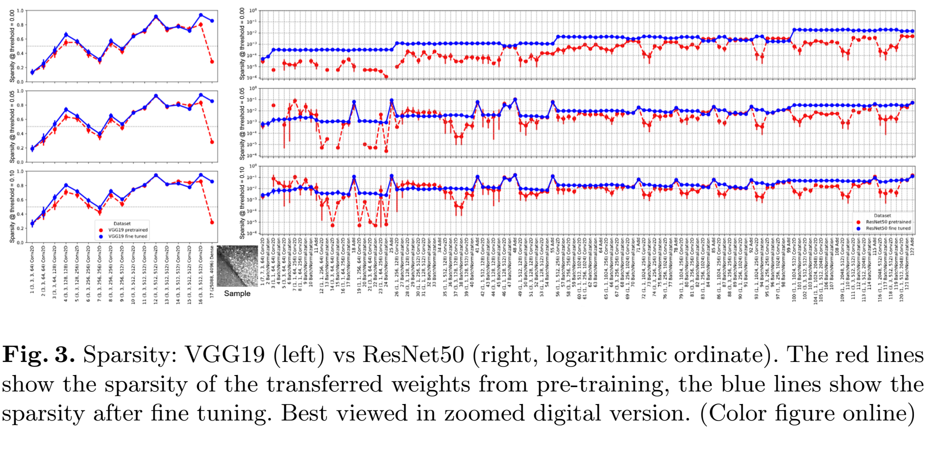


Fig. 3. Sparsity: VGG19 (left) vs ResNet50 (right, logarithmic ordinate). The red lines show the sparsity of the transferred weights from pre-training, the blue lines show the sparsity after fine tuning. Best viewed in zoomed digital version. (Color figure online)

the fine tuning leads to an increase of sparsity in the VGG19, the sparsity of ResNet50 seems to be not affected at all (see Fig. 3).

Since we did not observe a clear feature selection for any of the examined architectures, we assume the feature selection observed in [20] is an effect caused by their dataset, the selected data samples, and the architecture (VGG), respectively.

Additionally, we observed some basic patterns regarding sparsity, which reveal insights on how the architectures work: In particular, ReLU activations, 1×1 convolutions, and Squeeze-and-Excitation blocks increase the sparsity drastically. While the increase at ReLU layers is caused by negative activations, the increase at the other layers is due to their feature selection effect. In contrast, purely spatial convolutions lead to a decrease of sparsity.

Small Weight Changes. In addition to an increase in generalization ability, transfer learning also offers a remarkable reduction in training time. Often, this observation is explained as follows: Weights that were learned during the pre-training require only small changes to fit the new dataset. In our experiments, we observed that fine tuning had the following effects:

- For AlexNet and VGG19, in early convolutional layers, the weight changes are lower than in trainings from scratch (Fig. 4), which suggests that these features could be re-used. For intermediate convolutional layers, the weights have to be adapted as much as in trainings from scratch. Features in late convolutional layers are fine tuned less.
- For all modern architectures, weight changes were rather small. Features could be re-used to a high degree, since weight differences in most layers are smaller for fine tuning than for training from scratch.
- Because of their random initialization, weights in fully connected layers at the very end of each architecture are changed drastically, even more than in trainings from scratch. In AlexNet and VGG19, these layers contain more than 90% of all weights. Therefore, in sum, weight differences are higher for fine tuning than for trainings from scratch in these architectures.
- Due to the re-training of transferred color features, which make up ca. 50% of the first-layer filters, we observed relatively strong changes in the first convolutional layers regardless of the architecture. However, these changes are lower than in trainings from-scratch, suggesting that some filters could be re-used.
- Weights in pointwise convolutional layers and squeeze-and-excitation blocks are changed significantly stronger than weights in other layers. We observed clear peaks in the weight change diagrams for these layers in all modern architectures. Both types of layers are responsible for combining or weighting features of the previous layer. This indicates that previously learned features are re-combined during fine tuning.

In summary, our experiments do not confirm that weights require less changes when pre-trained weights were used for initialization. For some layers, even

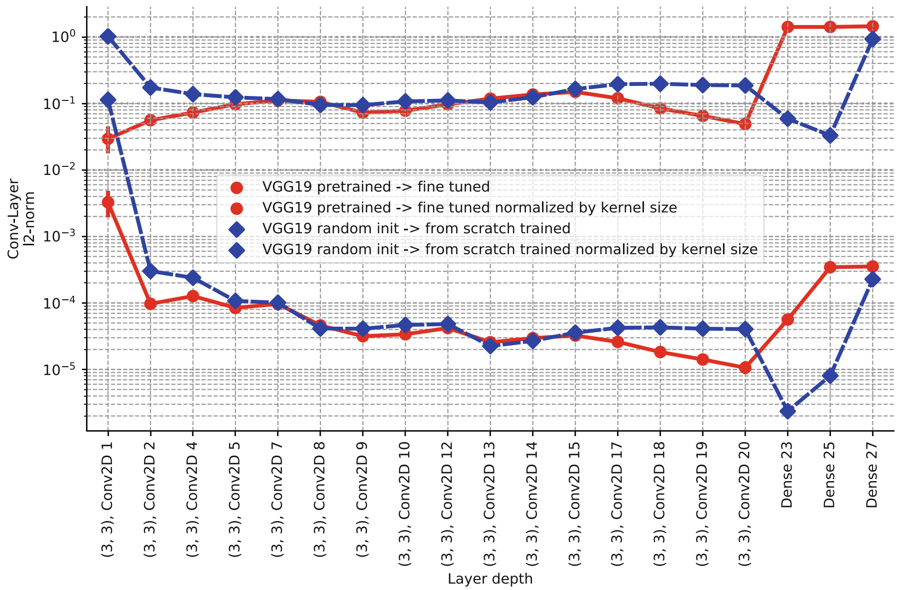


Fig. 4. Weight difference measured by euclidean distance of weights before and after training for layers of the VGG19 architecture (top curves) and the euclidean distance divided by the filter size (bottom curves)

greater weight changes were applied by the fine tuning in comparison to the training from scratch. Instead, we assume that training is faster since early-layers features can be re-used in classical architectures and features can be re-combined in modern architectures. This re-usability of features trained on a large dataset also has a positive effect on the generalization ability.

5 Conclusion

So far, transfer learning has not been examined systematically in the state of the art for deep-learning-based visual road condition assessment. Therefore, we performed extensive experiments to analyze the impact of transfer learning on the performance in the application of road surface image analysis. The generalization ability of all architectures considered in our analysis was significantly improved by the use of transfer learning, although this application differs significantly from the ImageNet dataset, which was used for pre-training. The training time has been reduced to a fraction of the time required to train a network from scratch. Furthermore, we analyzed transfer learning effects in order to explore how different architectures benefit from transfer learning. Classical architectures (AlexNet and VGG19) benefit from re-using features of early layers. Therefore, weights in early layers are adapted less during fine tuning and the sparsity of activations in mid and late layers is high. In contrast, for modern architectures (InceptionNet V3, ResNet50, SE-ResNet50, XceptionNet), we observe a low sparsity of activations in all layers. During fine tuning, layers responsible for combining and weighting features of previous layers are adapted more than other layers. We conclude that modern architectures re-use a larger fraction of features by re-combining them. Therefore, for deep-learning-based visual road condition assessment, modern architectures, especially those that contain skip connections, should be preferred over classical architectures in a transfer learning setting.

References

1. Alfarrarjeh, A., Trivedi, D., Kim, S.H., Shahabi, C.: A deep learning approach for road damage detection from smartphone images. In: Big Data (2018)
2. Alipour, M., Harris, D.K., Miller, G.R.: Robust pixel-level crack detection using deep fully convolutional neural networks. JCCE **33**(6), 04019040 (2019)
3. Bang, S., Park, S., Kim, H., Kim, H.: Encoder-decoder network for pixel-level road crack detection in black-box images. CACIE **34**(8), 713–727 (2019)
4. Bang, S., Park, S., Kim, H., Kim, H., et al.: A deep residual network with transfer learning for pixel-level road crack detection. In: ISARC, vol. 35 (2018)
5. Cao, W., Liu, Q., He, Z.: Review of pavement defect detection methods. IEEE Access **8**, 14531–14544 (2020)
6. Cha, Y.J., Choi, W., Büyüköztürk, O.: Deep learning-based crack damage detection using convolutional neural networks. CACIE **32**(5), 361–378 (2017)
7. Chen, H., Yao, M., Gu, Q.: Pothole detection using location-aware convolutional neural networks. IJMLC **11**, 899–911 (2020)
8. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: CVPR, pp. 1251–1258 (2017)
9. Du, Y., Pan, N., Xu, Z., Deng, F., Shen, Y., Kang, H.: Pavement distress detection and classification based on YOLO network. Int. J. Pavement Eng., 1–14 (2020). Taylor & Francis
10. Dung, C.V., Anh, L.D.: Autonomous concrete crack detection using deep fully convolutional neural network. Autom. Constr. **99**, 52–58 (2019)
11. Eisenbach, M., Stricker, R., Debes, K., Gross, H.M.: Crack detection with an interactive and adaptive video inspection system. AGT-IM **94**, 94–103 (2017)
12. Eisenbach, M., Stricker, R., Seichter, D., et al.: How to get pavement distress detection ready for deep learning? a systematic approach. In: IJCNN (2017)
13. Eisenbach, M., Stricker, R., Sesselmann, M., Seichter, D., Gross, H.M.: Enhancing the quality of visual road condition assessment by deep learning. In: WRC (2019)
14. Feng, C., Zhang, H., Wang, S., et al.: Structural damage detection using deep convolutional neural network and transfer learning. JCE **23**(10), 4493–4502 (2019)
15. Gopalakrishnan, K.: Deep learning in data-driven pavement image analysis and automated distress detection: a review. Data **3**(3), 28 (2018)
16. Gopalakrishnan, K., Khaitan, S.K., et al.: Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. CBM **157**, 322–330 (2017)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
18. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., et al.: MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv (2017)

19. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR, pp. 7132–7141 (2018)
20. Kim, S., Kim, W., Noh, Y.K., Park, F.C.: Transfer learning for automated optical inspection. In: IJCNN, pp. 2517–2524 (2017)
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NIPS, pp. 1097–1105 (2012)
22. Li, B., Wang, K.C., Zhang, A., Yang, E., Wang, G.: Automatic classification of pavement crack using deep convolutional neural network. In: IJPE (2018)
23. Liu, Y., Yao, J., Lu, X., Xie, R., Li, L.: DeepCrack: a deep hierarchical feature learning architecture for crack segmentation. Neurocomputing **338**, 139–153 (2019)
24. Maeda, H., Sekimoto, Y., Seto, T., et al.: Road damage detection and classification using deep neural networks with smartphone images. CACIE **33**(12), 1127–1141 (2018)
25. Milhomem, S., da Silva Almeida, T., da Silva, W.G., et al.: Weightless neural network with transfer learning to detect distress in asphalt. arXiv (2019)
26. Ng, A.: Nuts and bolts of building AI applications using deep learning. In: Tutorial NIPS (2016)
27. Nhat-Duc, H., et al.: Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network. AIC **94**, 203–213 (2018)
28. Özgenel, Ç.F., Sorguç, A.G.: Performance comparison of pretrained convolutional neural networks on crack detection in buildings. In: ISARC, vol. 35 (2018)
29. Riid, A., Lõuk, R., et al.: Pavement distress detection with deep learning using the orthoframes acquired by a mobile mapping system. Appl. Sci. **9**(22), 4829 (2019)
30. Roberts, R., Giancontieri, G., et al.: Towards low-cost pavement condition health monitoring and analysis using deep learning. Appl. Sci. **10**(1), 319 (2020)
31. Seichter, D., Eisenbach, M., Stricker, R., et al.: How to improve deep learning based pavement distress detection while minimizing human effort. In: CASE (2018)
32. Silva, W.R.L.d., Lucena, D.S.d.: Concrete cracks detection based on deep learning image classification. In: MDPI Proceedings (2018)
33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv (2015)
34. Stricker, R., Eisenbach, M., et al.: Improving visual road condition assessment by extensive experiments on the extended gaps dataset. In: IJCNN (2019)
35. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR, pp. 2818–2826 (2016)
36. Wang, Y.J., Ding, M., Kan, S., Zhang, S., Lu, C.: Deep proposal and detection networks for road damage detection and classification. In: Big Data (2018)
37. Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X., Ling, H.: Feature pyramid and hierarchical boosting network for pavement crack detection. TITS (2019)
38. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: NIPS, pp. 3320–3328 (2014)
39. Zhang, A., Wang, K.C., Fei, Y., et al.: Automated pixel-level pavement crack detection on 3D asphalt surfaces with a recurrent neural network. CACIE **34**(3), 213–229 (2019)
40. Zhang, K., Cheng, H., Zhang, B.: Unified approach to pavement crack and sealed crack detection using preclassification based on transfer learning. JCCE (2018)
41. Zhang, L., Yang, F., Zhang, Y.D., Zhu, Y.J.: Road crack detection using deep convolutional neural network. In: ICIP, pp. 3708–3712 (2016)
42. Zhuang, F., et al.: A comprehensive survey on transfer learning. arXiv (2019)