Automation in Construction 116 (2020) 103199

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Deep convolution neural network-based transfer learning method for civil infrastructure crack detection

Qiaoning Yang*, Weimin Shi, Juan Chen, Weiguo Lin

College of Information Science and Technology, Beijing University of Chemical Technology, 100029 Beijing, China

ARTICLE INFO

Keywords:
Crack detection
Transfer learning
Deep convolution neural network

ABSTRACT

Crack detection is critical to guaranteeing safety of bridges, highway and other infrastructures. The deep convolution neural network (DCNN) makes it possible to efficiently and accurately implement image classification, and the accumulated knowledge of DCNN in other domains can be reused for crack detection. In this paper, we propose a transfer learning method based on DCNN to detect cracks. The proposed method models the knowledge learned by DCNN and transfers three kinds of knowledge from other research achievements: sample knowledge, model knowledge and parameter knowledge. New fully connected layers have emerged in the Visual Geometry Group (VGG) network as a new learning framework for crack detection. The performance and validity of the proposed method are verified. Compared with other detection methods, the proposed method can detect many kinds of cracks with a high detection accuracy. The detection accuracy for CCIC [24] is 99.83%, that for BCD [25] is 99.72%, and that for SDNET [45] is 97.07%. The accumulated knowledge in this method can also be transferred to other research work.

1. Introduction

Cracks are a common defect found in civil engineering structures. Cracks not only affect the structural health but also induce other issues [1]. Crack detection plays an important role in the structural health and reliability maintenance of infrastructures. Traditional manual crack detection methods are time-consuming, laborious, dangerous and subjective [2,3]. This leads to high maintenance costs, low maintenance efficiency and potential security risks. With the development of science and technology, researchers have carried out a variety of studies for technical research and exploration.

Image processing based-crack detection methods have been widely used in infrastructure health and maintenance. These methods use a variety of statistical features of crack imaging and machine learning algorithms to implement detection, classification, regression, etc. Eduardo Zalama [4] used the Gabor filter to determine spatial and frequency features and then used an assembled classifier to detect cracks. Zou et al. [5] proposed a geodesic shadow-removal algorithm and a crack probability map for automatic crack detection. The shadow-removal algorithm reduces the influence of background noises, and the crack probability map plays the role of decision maker. Zhao et al. [6] converted images into parametric surfaces and then detected crack surfaces by using geodesic distance maps. Prasanna et al. [7] used 5 intensity-based features and 5 scale-space features as the input of the spatially tuned robust multifeature (STRUM) classifier to detect cracks automatically with an accuracy of 95%. Lin et al. [8] used the connected domain area, shape extremum, morphology and support vector data description (SVDD) methods to detect cracks. These studies have devoted considerable effort to the representation of image features. However, due to the limited number of features, the differences among data sets, the presence of background noise and the variety of application scenarios, the traditional image processing methods may produce undesired results, and the adaptability of the model and the reusability of knowledge are limited [9].

In recent years, the deep convolutional neural network has demonstrated state-of-the-art and human-competitive performance in image classification, object detection, image segmentation, and so on [10–13]. Verified by the facts in the ImageNet large-scale visual recognition challenge [14,15], many excellent deep convolution neural network models, namely, AlexNet [11], GoogLeNet [13], VGG Net [16], and ResNet [17], have been developed. Their performance is equivalent to or even higher than that of human beings. Inspired by such achievements, many CNN-based algorithms have been proposed for crack segmentation [18–22] and crack classification [23–26]. Cao et al. [20] proposed a crack detection method based on a deep fully convolutional network (FCN) for semantic segmentation on concrete crack images. On a public concrete crack dataset [24], a whole encoder-decoder FCN network with the VGG16-based encoder was trained end to

* Corresponding author.
E-mail address: yangqn@mail.buct.edu.cn (Q. Yang).

https://doi.org/10.1016/j.autcon.2020.103199
Received 29 December 2019; Received in revised form 21 March 2020; Accepted 21 March 2020
0926-5805/

end. The FCN network achieved an approximately 90% average precision. Liu et al. [21] proposed a deep hierarchical convolutional neural network to predict pixelwise crack segmentation in an end-to-end method. The designed model learned and aggregated multiscale and multilevel features from the low-level convolutional layers to the high-level convolutional layers. Young-Jin Cha used a DCNN architecture with eight layers, including convolution, pooling, ReLU, and Softmax, to detect concrete cracks [23]. The trained classifier consequently recorded 98% accuracies in both training and validation. Xu et al. [25] proposed an end-to-end crack detection model based on the convolutional neural network, taking advantage of the atrous spatial pyramid pooling (ASPP) module and depthwise separable convolution. The ASPP module enables the network to extract multiscale context information, while depthwise separable convolution reduces computational complexity. The proposed model achieved a detection accuracy of 96.37% without pre-training using bridge crack data [27].

Compared with crack detection methods based on image processing, DCNN can automatically extract features and make classification or segmentation decisions. DCNN can also achieve better performance than other algorithms. However, how can the accumulated knowledge of DCNN be applied to other fields? Transfer learning [28–32] provides an approach to solve this problem, as this approach can transfer learning knowledge from one domain to other domains.

Transfer learning techniques attempt to transfer knowledge from previous tasks to target tasks when the latter has fewer high-quality training data [33]. As shown in Fig. 1, knowledge A and B accumulated in a source task can be transferred to a target task. Transfer knowledge A and B, combined with knowledge D in the new model, can effectively complete the target task. Knowledge achieved by CNN using the ImageNet dataset is widely reused by research to other domains [34–36]. In the crack detection domain, Zhang et al. [37] transferred generic ImageNet knowledge, including all convolution layers and corresponding weights, and the proposed detection approach achieved good performance (recall = 0.947, precision = 0.846). Kucuksubasi et al. [38] fine-tuned the InceptionV3 [39] network model with ImageNet weights to detect cracks by autonomous UAVs. The fine-tuned model accurately predicted 62,417 cracks from a total of 64,095. The accuracy reached 97.382% in cross validation. These results demonstrate the convenience of using an InceptionV3 model as the backbone for transfer learning for such a crack detection application. B. H. Kien et al. [40] transferred all convolution layers and weights of VGG16 to detect cracks in plastic gears, and the accuracy approached 99%. Mohamad Alipour et al. [41] used ResNet to study the robustness of material-specific deep learning models for crack detection across different materials. The combination of asphalt and concrete images were used to improve the performance of crack detection.

Although these transfer learning methods based on DCNN have achieved good results in crack detection, the knowledge transferred is different. Some are data-based transfer, some are model-based transfer, and some are weight-based transfer. How to systemically and efficiently transfer the knowledge learned by DCNN is still worth investigating in the crack detection domain. In this paper, the knowledge modeling and knowledge transfer of DCNN for crack detection are investigated, and the main contributions are as follows:

- A transfer learning method based on multiple DCNN knowledge for crack detection is proposed.
- The knowledge transfer modeling of DCNN is studied, and the influence of transferring different kinds of knowledge on crack detection is evaluated.
- Compared with [25,42] and [43], the proposed transfer learning method has more accurate detection performance for various cracks in pavement, walls and decks in different databases.

This paper is organized as follows. Section 1 briefly reviews the research progress in crack detection. Section 2 analyses the knowledge learned from DCNN and proposes a transfer learning model for crack detection. In Section 3, an end-to-end DCNN-based transfer learning method for crack detection is proposed. Section 4 demonstrates the effectiveness of the proposed method by experiments. Finally, Section 5 concludes the paper.

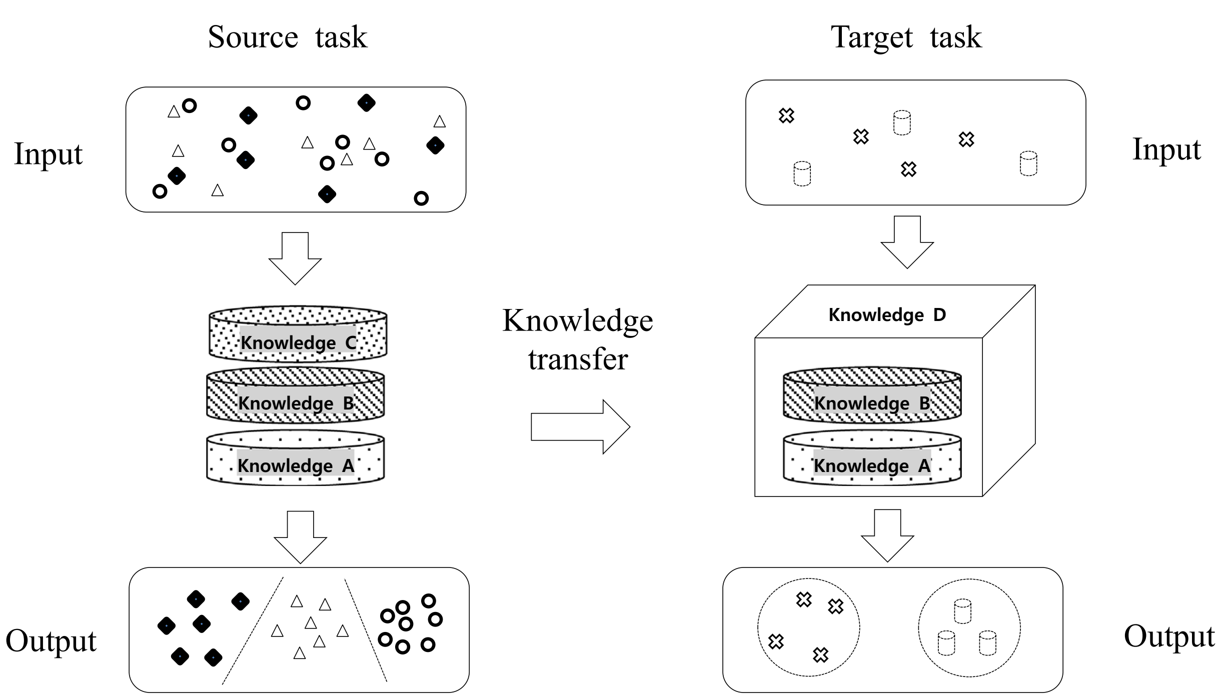


Fig. 1. Illustration of transfer learning.

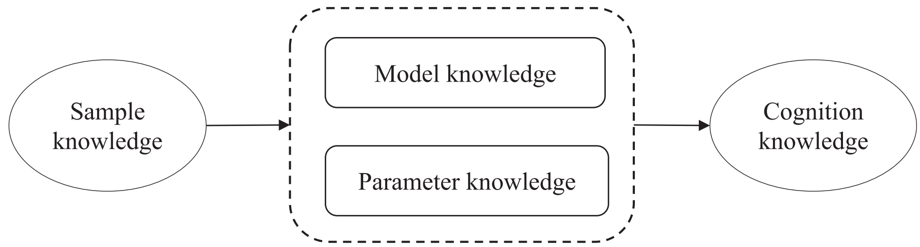


Fig. 2. Knowledge learned in DCNN.

2. Transfer learning model

2.1. Knowledge learned in DCNN

DCNN is a deep convolutional neural network model for machine learning. It includes four parts: sample data, model framework, parameters and output. The sample data provides a knowledge base for DCNN learning. The model framework is an approach to represent and process input knowledge. Combined with its parameters, its performance reflects mastery of the knowledge. The output results reflect the ability of DCNN to understand the input knowledge. Therefore, a pre-trained DCNN model in one domain has the following knowledge, as shown in Fig. 2.

- Sample knowledge: the content and scope of the input data.
- Model knowledge: the verified model framework.
- Parameter knowledge: the optional parameters of the model.
- Cognition knowledge: the output results of the model.

Based on the knowledge learned in one domain, the transfer of knowledge to crack detection is investigated in the following sections.

2.2. VGG16-based transfer learning model

VGG16 is a classic convolutional neural network that is trained on more than a million images from the ImageNet database [14] and widely used as a reference model. From the viewpoint of the knowledge model for DCNN shown in Fig. 2, classic models can be used for the study of transfer learning. Therefore, VGG16 is chosen as a base model for this transfer learning study. The model has 16 layers and 138 million parameters; as shown in Fig. 3, VGG16 can classify images into 1000 object categories with high confidence.

The pre-trained VGG16 model consists of a model framework and parameters. The model framework includes two parts: 13 convolutional layers and 3 fully connected layers. The former layers use a 3 × 3 convolution kernel to extract the features of input images, and the fully connected layers establish the representation relationship between the features and the classification results.

Joining the knowledge obtained in VGG16 with other knowledge, a transfer learning framework for crack detection is established in Fig. 4. There are three kinds of knowledge transferred from other domains. The first is sample knowledge, which consists of multiple crack datasets from different application scenarios. The second is model knowledge transferred from VGG16 convolution layers, and the third is the partial parameters of VGG16. New added fully connected layers and new parameters are used to classify crack and non-crack images.

3. End-to-end DCNN-based transfer learning method for crack detection

According to the transfer learning framework, an end-to-end DCNN-based transfer learning method is proposed for crack detection, as shown in Fig. 5. The method contains multiple input datasets, 13 convolution layers and 2 fully connected layers. Each input image is convoluted by a 3 × 3 convolution kernel to extract image features. The ReLu function is used as the activation function in the first 13 convolution layers and the first fully connected layer. Max-pooling layers after the convolutional layers are used to decrease the training time, reduce the amount of parameters and control overfitting. Two fully connected layers exist before the output of model is proposed: the first one contains 256 neurons, and the second contains 2 neurons. Softmax is utilized in the last fully connected layer, and this layer classifies the input image into cracks or non-cracks.

In this proposed method, sample transfer learning, model transfer learning and parameter transfer learning are three kinds of approaches used to fine-tune performance. We try to find the best combinational transfer learning approaches to detect cracks.

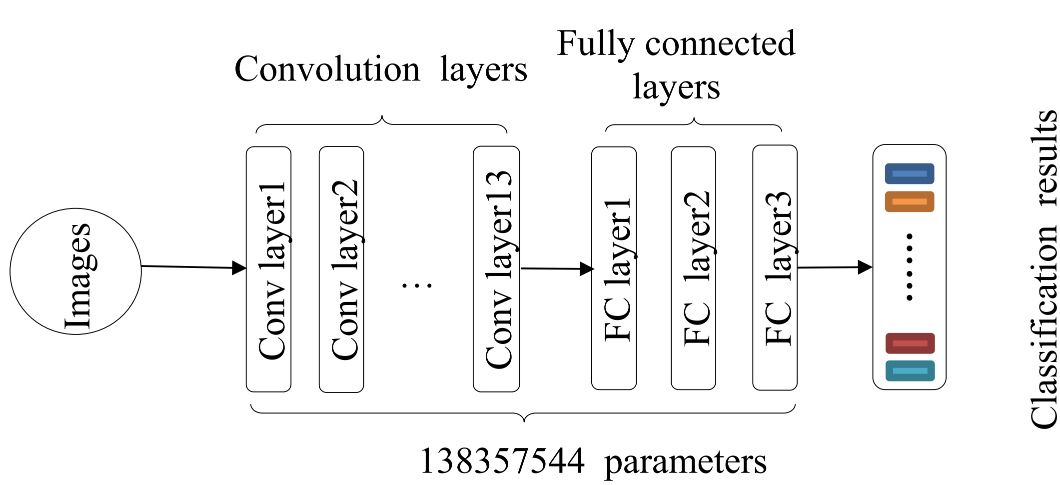


Fig. 3. VGG16 model structure.

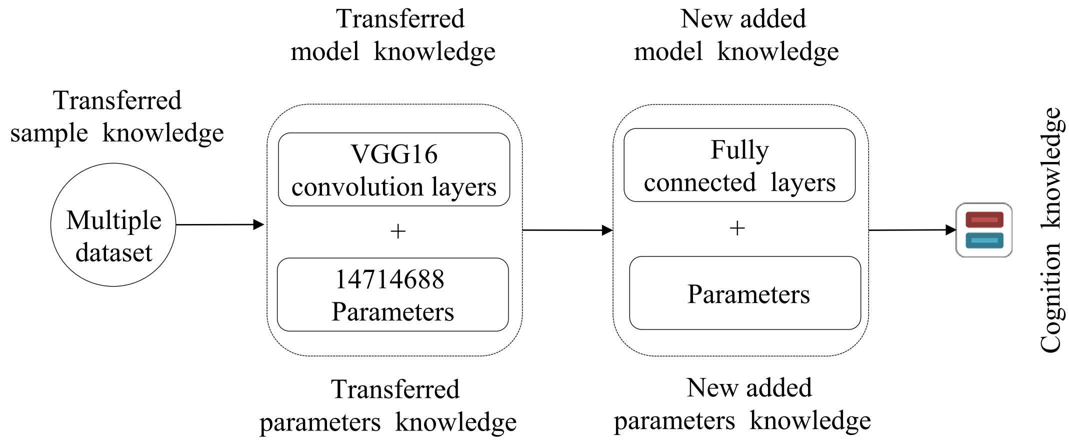


Fig. 4. Transfer learning framework for crack detection.

4. Experiments and analysis

All experiments are performed on an Intel(R) Core(TM) i9-9900K CPU@ 3.60 GHz CPU with 64 GB RAM and an NVIDIA 2080Ti*1 GPU. The proposed transfer learning method is implemented with the deep learning library Keras in the Python IDE PyCharm.

In this paper, some quantitative indicators are used to evaluate the performance of the detection method and detection results: accuracy, precision and the area under the receiver operating characteristic (AUC).

1. Accuracy

$$Accuracy = \frac{TN + TP}{TN + FN + FP + TP} \tag{1}$$

2. Precision

$$Precision = \frac{TP}{FP + TP} \tag{2}$$

where *TP* is the number of crack fragments correctly detected, *TN* is the number of non-crack fragments correctly detected, *FP* is the number of non-crack fragments treated as crack fragments, and *FN* is the number of crack fragments treated as non-crack fragments.

3. AUC

AUC, i.e., the area under the receiver operating characteristic (ROC) plot, is a measurement of the discriminatory capacity of classification models.

4.1. Sample transfer learning

Three crack datasets are used as input samples, namely, the CCIC dataset [24], the SDNET dataset [44] and the BCD dataset [25]. CCIC is collected from various METU campus buildings, as shown in Fig. 6. CCIC contains 40,000 RGB images with a resolution of 227 × 227 pixels, which are divided into negative and positive categories. The SDNET dataset contains more than 56,000 annotated images of cracked and non-cracked concrete bridge decks, walls, and pavements. The dataset includes cracks as narrow as 0.06 mm and as wide as 25 mm. The dataset also includes images with a variety of obstructions, including shadows, surface roughness, scaling, edges, holes, and

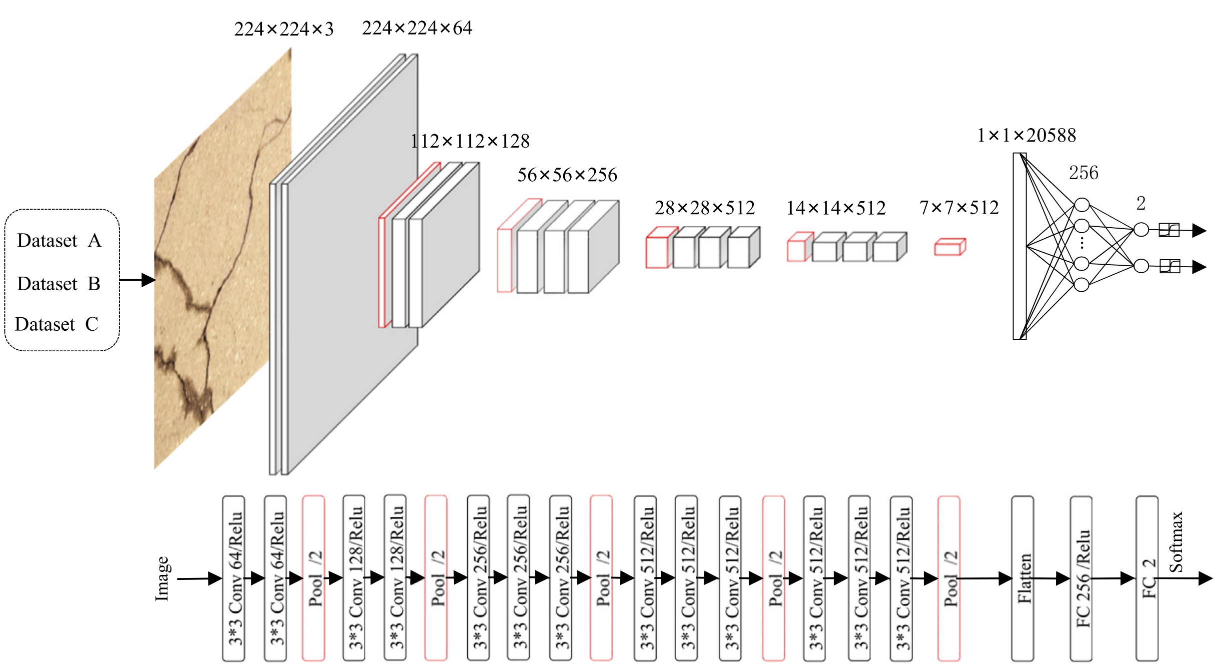


Fig. 5. End-to-End DCNN-based transfer learning method for crack detection.

(a) Crack images (b) Non-crack images

Fig. 6. Examples of images contained in the CCIC dataset.

(a) Crack images (b) Non-crack images

Fig. 7. Examples of images contained in the SDNET dataset.

(a) Crack images (b) Non-crack images

Fig. 8. Examples of images contained in the BCD dataset.

background debris, as shown in Fig. 7. A total of 5069 bridge crack and non-crack images are included in the BCD dataset with a resolution of 224 × 224 pixels. The dataset also contains images of bridge shading, water stains and strong light, as shown in Fig. 8.

From Figs. 6–8, it is easy to find that the content and the scope of these images are very different. Images in the SDNET dataset are from three different sources, including bridge decks, walls and pavements. The variety and complexity of SDNET are greater than those of the other two datasets. SDNET has a larger knowledge scope and richer knowledge than CCIC and BCD. As shown in Table 1, in this paper, 14,000 images from CCIC, 4916 images from SDNET and 5000 images from BCD are used to train, validate and test the proposed transfer learning method.

To investigate the influence of different input datasets on the proposed method, each dataset is used separately to train and validate the model under the same conditions. The trained models are then used to test the images from CCIC, SDNET and BCD separately. Therefore, the numbers of samples in all crack and non-crack images are balanced, and the accuracy is used to evaluate the performance of the proposed method. The results are shown in detail in Table 2.

As shown in Table 2, the proposed method trained by each dataset separately can achieve the best accuracy in testing its own images. The model trained by CCIC can detect the testing images in CCIC with 99.91% accuracy. Correspondingly, the model trained by SDNET can detect the testing images in SDNET with 90.95% accuracy, and the model trained by BCD can detect the testing images in BCD with 99.37% accuracy. However, using the model trained by one dataset to test the images from another dataset cannot achieve a good detection accuracy. The model trained by SDNET achieves 50.95% accuracy in testing the CCIC images and 64.96% accuracy in testing the BCD images. Such results reflect that the knowledge in one sample domain can be efficiently learned by DCNN to detect samples on its own, but the learned knowledge cannot accurately detect cracks in other datasets.

To evaluate the accumulation of knowledge transferred from different datasets, all image datasets from CCIC, SDNET and BCD are gathered into one, called the ALL dataset. The model trained by the ALL dataset can detect images with 99.9% accuracy for CCIC, 96.56% for SDNET and 99.61% for BCD, as listed in Table 2. These results show that the knowledge of different datasets is transferred and learned by the proposed method. The performance improvement for the small-sample datasets of SDNET and BCD can be achieved using sample transfer learning from larger datasets such as CCIC. Using the model trained by the ALL dataset to detect cracks in SDNET can achieve 96.56% accuracy, which is higher than that of the model trained by SDNET itself (5.61%). The same performance achievement with 0.24% is obtained in BCD image detection.

Table 1
Image samples used from the different datasets.

Number of images	CCIC		SDNET		BCD	
	Crack	Non-crack	Crack	Non-crack	Crack	Non-crack
Train	4000	4000	1500	1500	1000	1000
Validate	2000	2000	560	627	1000	1000
Test	1000	1000	229	500	500	500

Table 2
Sample transfer learning results with different training datasets.

Transfer learning approach	Sample knowledge	Model knowledge		Parameter knowledge		Cognition knowledge (accuracy %)			
	Data sets	VGG16 conv. layers	New FC layers	CLP trainable	CLP untrainable	Train	CCIC test	SDNET test	BCD test
Sample transfer learning	CCIC	√	2 dense	√		99.71	99.91	92.30	95.62
	SDNET	√	2 dense	√		95.67	50.95	90.95	64.96
	BCD	√	2 dense	√		96.90	96.65	84.22	99.37
	ALL	√	2 dense	√		98.53	99.90	96.56	99.61

Table 3
Model transfer learning results with different FC layers.

Transfer learning approach	Sample knowledge	Model knowledge		Parameter knowledge		Cognition knowledge (ACC %)			
	Data sets	VGG16 conv. layers	New FC layers	CLP trainable	CLP untrainable	Train	CCIC test	SDNET test	BCD test
Model transfer learning	ALL	√	1 dense	√		98.66	99.15	92.59	98.97
	ALL	√	2 dense	√		98.53	99.90	96.56	99.61
	ALL	√	3 dense	√		98.18	99.60	95.75	99.50

4.2. Model transfer learning

In the proposed method, 13 convolution layers are transferred from the VGG16 model, and the FC layers are newly added. With different numbers of FC layers, the performance of each model is evaluated. As shown in Table 3, the model with a two-neuron FC layer can detect cracks with 99.15% accuracy for CCIC, 92.59% accuracy for SDNET and 98.97% accuracy for BCD. Thus, the transferred convolution layer can extract the image features accurately. With an increasing number of FC layers, the nonlinear mapping capability of the model is improved. The model with 2 FC layers composed of 256 neurons and 2 neurons correspondingly achieves better performance than that with one FC layer. When 3 FC layers with two dense 256-neuron layers and a dense 2-neuron layer are used, compared with the model with 2 FC layers, the testing accuracy is decreased. Therefore, for a simple binary classification problem, 2 FC layers are enough.

4.3. Parameter transfer learning

In the proposed method, there are 14,714,688 parameters that are transferred from the pre-trained VGG16 model. These parameters contain filter weights and biases of the convolution layers, which are used to extract image features.

To verify the performance under different transferred parameters, two kinds of experiments are conducted separately. The first kind is that the parameters of the convolution layers (CLP) are pre-trained and untrainable in this task; only the parameters of FC (FCP) layers are trained and updated. The second kind is that the CLP and FCP are fine-tuned using ALL data. The performance results are listed in Table 4.

It is easy to find that the performance of the model with untrainable parameters is poor, particularly for SDNET, with an accuracy of only approximately 64%. The untrainable parameters transferred from VGG cannot effectively represent the crack images in SDNET. For the images in the CCIC and BCD datasets, the untrainable parameters can readily extract features and achieve good detection accuracies. There are two reasons for these results. One is that the pre-trained parameters of VGG can extract most kinds of image features, but the knowledge in SDNET images cannot be learned. The other is that the feature knowledge in CCIC and BCD is simple and therefore obtainable by the pre-trained parameters.

When the CLP and FCP are trainable, after 20 epochs, the model performance reaches a high level with 99.9% accuracy for CCIC, 96.65% accuracy for SDNET and 99.61% accuracy for BCD. With an increasing number of epochs, the performance achieves relatively little improvement. Therefore, in the proposed method, transferred parameters are set to be trainable, and the number of epochs is set to 20.

To further investigate the advantages of parameter transfer learning, full learning models with random initialized parameters are trained. The testing accuracy and training time are used to evaluate the performance difference between models. As shown in Table 5, after more than 200 epochs, the model with random initialized parameters obtains an increased accuracy. However, the performance is still lower than that of the model using parameter transfer learning because the knowledge of the three crack datasets is less than that of the ImageNet dataset. The knowledge learned by the parameters is less. With an increasing number of epochs, the model with random initialized parameters shows low learning efficiency on crack datasets, as shown in Fig. 9. The training time of the full learning model is much higher than that of the transfer learning model. The parameter learning model uses 20 epochs and 2524 s, which can achieve much better accuracy than the full learning model using 300 epochs and 38,198 s. Thus, parameter transfer learning has advantages not only in accuracy but also in efficiency.

4.4. All knowledge transfer learning

After the above analysis, sample transfer learning with multiple datasets, model transfer learning with the convolution layers of VGG16

Table 4
Parameter transfer learning results under different scenarios.

Transfer learning approach	Sample knowledge	Model knowledge		Parameter knowledge		Cognition knowledge (ACC %)			
	Data sets	VGG16 conv. layers	New FC layers	CLP trainable	CLP untrainable	Train	CCIC test	SDNET test	BCD test
Parameter transfer learning	ALL	√	2 dense	√ epoch = 20		98.53	99.90	96.56	99.61
	ALL	√	2 dense		√ epoch = 20	94.43	99.15	64.91	91.21
	ALL	√	2 dense	√ epoch = 50		98.87	99.95	96.02	99.40
	ALL	√	2 dense		√ epoch = 50	95.45	99.40	64.47	93.50

Table 5
Full learning results under different scenarios.

Transfer learning approach	Sample knowledge	Model knowledge		Parameter knowledge		Cognition knowledge (ACC %)				Training time(s)
	Data sets	VGG16 conv. layers	New FC layers	CLP trainable	Epochs	Train	CCIC test	SDNET test	BCD test	
Parameter transfer learning	ALL	√	2 dense	√	20	98.53	99.90	96.56	99.61	2524
Full learning	ALL	√	2 dense	Random initialized	100	95.04	98.00	94.65	87.88	13,034
	ALL	√	2 dense	Random initialized	200	97.17	98.85	93.55	91.59	25,703
	ALL	√	2 dense	Random initialized	300	97.69	99.75	94.23	94.55	38,198

and parameter transfer learning with the pre-trained parameters using the ImageNet dataset can all improve model performance. Applying these parameters to the transfer learning model can obtain very excellent performance. Therefore, the best choice for the proposed method is using CCIC, SDNET and BCD as sample datasets for training; the 13 convolution layers of VGG16 are used to extract crack features, 2 fully connected layers are employed for crack classification, and 14,714,688 pre-trained parameters are used for accelerating and fine-tuning the model. This transfer learning model for all knowledge with 20 epochs can achieve stable performance, as indicated by the multiple training and testing results shown in Table 6.

Compared with other CNN research works [25,42,43] based on the CCIC, SDNET and BCD datasets, the proposed method also has clear superiority in accuracy, precision and AUC scores, as shown in Table 7.

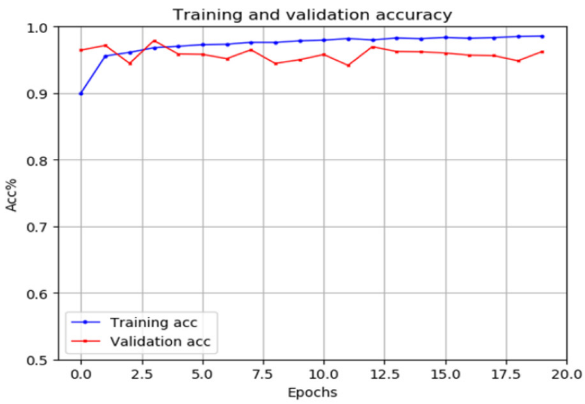
Zhang et al. [42] used 4 convolution layers and 2 fully connected layer to construct a detecting model trained by the CCIC dataset and achieved 86.96% precision and an AUC score of 0.9592 for the crack images in CCIC. On the same CICC test dataset, the proposed method has an accuracy of 99.50% and an AUC score of 1.

In 2018, Sattar et al. [43] used a crack detection algorithm based on the pre-trained AlexNet DCNN architecture to detect cracks. For different crack images, they obtained accuracies of 91.92% for a bridge deck (D), 89.31% for a wall (W) and 95.52% for pavement (P) in the SDNET dataset. Correspondingly, the proposed method in this paper can reach 97.07% accuracy for all crack and non-crack images in SDNET.

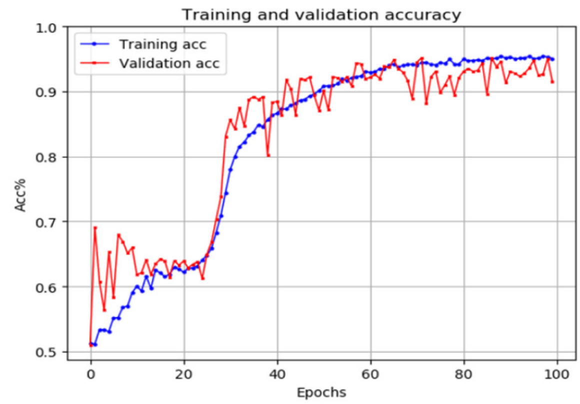
In 2019, Xu et al. [25] used the BCD dataset to train an ASPP-based convolution neural network with 16 convolution layers and 2 fully connected layers. The model is fully trained by samples in the BCD dataset, and the parameters are randomly initialized. An accuracy of 96.73% and a precision of 78.11% were obtained. Using the proposed method in this paper, the testing accuracy can reach 99.72%, and the precision can reach 96.46% for BCD images.

5. Conclusions

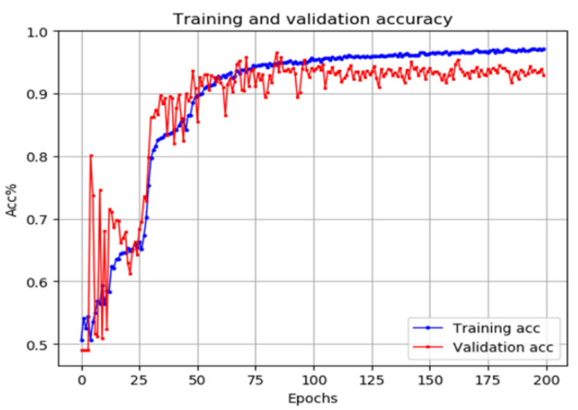
In this paper, a transfer learning model for crack detection based on a deep convolution neural network is proposed. Combined with sample knowledge, model knowledge and parameter knowledge, an end-to-end transfer learning crack detection method based on VGG16 is verified. Experimental results show that transfer learning can improve detection performance and efficiency as well as reduce training time. Transfer learning of multiple similar samples can improve the performance obtained by independent training of a single class of samples. For instance, bridge cracks can be used as training samples to improve the detection performance of wall cracks, and vice versa. Two fully connected layers are enough for VGG16 to be used for crack detection, and too many fully connected layers will not improve the overall performance of the model. Parameter transfer learning has obvious advantages over full learning model for the detection performance, especially for complex datasets such as SDNET and BCD, and the performance can be improved by 2.333% and 5.06%, respectively. At the same time, in terms of efficiency, parameter transfer learning can reduce the training time by more than 10 times relative to full learning to achieve the same performance. Compared with other DCNN-based models [25,42,43], the proposed method has obvious advantages in the accuracy and precision of crack detection. The knowledge accumulated by this method can also be used in other research and applications. However, there are still many issues to be further studied in the transfer learning of crack detection, such as the performance verification of the proposed transfer learning method based on other CNN models and the quantitative representation of transfer knowledge. These issues are exactly what we will continue to study in the future.



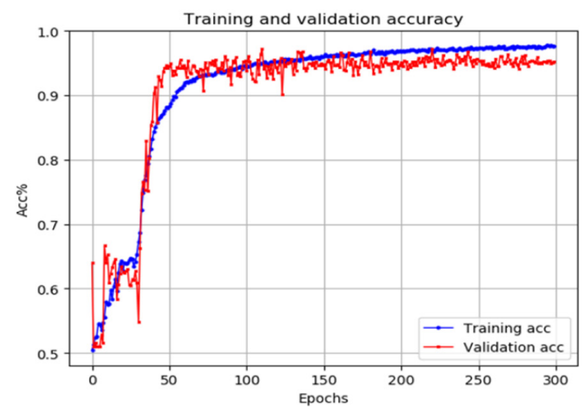
(a) Performance of transfer learning model, epoch =20



(b) Performance of full learning model, epoch =100



(c) Performance of full learning model, epoch =200



(d) Performance of full learning model, epoch =300

Fig. 9. Training and validation performance of parameter transfer learning and full learning model.

Table 6
Performance of the end-to-end DCNN-based transfer learning model.

Transfer learning approach	Sample knowledge	Model knowledge		Parameter knowledge		Cognition knowledge (ACC %)			
	Data sets	VGG16 conv. layers	New FC layers	CLP trainable	Epoch	Train	CCIC test	SDNET test	BCD test
Sample + model + parameter transfer learning	ALL	√	2 dense	√	20	98.53	99.90	96.56	99.61
	ALL	√	2 dense	√	20	98.54	99.90	97.26	99.90
	ALL	√	2 dense	√	20	98.54	99.70	97.39	99.65
Average						98.54	99.83	97.07	99.72

Table 7
Comparison with other CNN research works.

Models	Sample knowledge	Model knowledge		Parameter knowledge		Cognition knowledge		
	Data sets	Conv. layers	FC layers	Pre-trained parameter transfer	Epochs	Test accuracy (%)	Test precision (%)	AUC
Zhang 2016 [42]	CCIC	4	2	No	–	–	86.96	0.9592
Sattar 2018 [43]	SDNET	5 of AlexNet	3 of AlexNet	Yes	–	D: 91.92 W: 89.31 P: 95.52	–	–
Xu 2019 [25]	BCD	16	2	No	300	96.37	78.11	–
Proposed method	CCIC+ SDNET+ BCD	13 of VGG16	2	Yes	20	CCIC: 99.83 SDNET:97.07 BCD: 99.72	CCIC: 99.50 SDNET: 99.80 BCD: 96.46	CCIC: 1.0000 SDNET:0.9958 BCD: 0.9999

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. Gavilán, D. Balcones, O. Marcos, D.F. Llorca, M.A. Sotelo, I. Parra, M. Oca, P. Aliseda, P. Yarza, A. Amírola, Adaptive road crack detection system by pavement classification, Sensors 11 (10) (2011) 9628–9657, https://doi.org/10.3390/s111009628.

[2] H. Kim, S.H. Sim, S. Cho, Unmanned Aerial Vehicle (UAV)-powered concrete crack detection based on digital image processing, 6th International Conference on Advances in Experimental Structural Engineering, Urbana, United States, 2015, August.

[3] D.Y. Wang, P.H. Hu, Y.B. Dai, Asphalt pavement pothole detection and segmentation based on wavelet energy field, Mathematical Problems in Engineering: Theory, Methods and Applications 2017 (8) (2017) 1–13, https://doi.org/10.1155/2017/1604130.

[4] Eduardo Zalama, Jaime Gómez-García-Bermejo, Roberto Medina, José Llamas, Road crack detection using visual features extracted by Gabor filters, Journal Computer-Aided Civil and Infrastructure Engineering 29 (5) (2014) 342–358, https://doi.org/10.1111/mice.12042.

[5] Q. Zou, Y. Cao, Q. Li, Q.Z. Mao, S. Wang, CrackTree: automatic crack detection from pavement images, Pattern Recogn. Lett. 33 (3) (2012) 227–238, https://doi.org/10.1016/j.patrec.2011.11.004.

[6] G. Zhao, T. Wang, J. Ye, Surface shape recognition method for crack detection, J. of Electronic Imaging. 23 (3) (2014) 1267–1276, https://doi.org/10.1117/1.JEI.23.3.033013.

[7] P. Prasanna, K.J. Dana, N. Gucunski, B. Basily, Automated crack detection on concrete bridges, IEEE Trans. Autom. Sci. Eng. 13 (2) (2016) 591–599, https://doi.org/10.1109/TASE.2014.2354314.

[8] W. Lin, Y. Sun, Q. Yang, Y. Lin, Real-time comprehensive image processing system for detecting concrete bridges crack, Comput. Concr. 23 (6) (2019) 445–457, https://doi.org/10.12989/cac.2019.23.6.445.

[9] D. Sattar, J.T. Robert, M. Marc, Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete, Constr. Build. Mater. 186 (2018) 1031–1045, https://doi.org/10.1016/j.conbuildmat.2018.08.011.

[10] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[11] A. Krizhevsky, I. Sutskever, G. Hinton, ImageNet classification with deep convolutional neural networks, Neural Information Processing Systems 25 (2012) 1097–1105, https://doi.org/10.1145/3065386.

[12] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, Proceedings of the Computer Vision-European Conference on Computer Vision (ECCV), Zurich, Switzerland, 2014, pp. 818–833, , https://doi.org/10.1007/978-3-319-10590-1_53.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, IEEE Conference on Computer Vision and Pattern Recognition(CVPR), IEEE, 2015, , https://doi.org/10.1109/CVPR.2015.7298594.

[14] J. Deng, W. Dong, R. Socher, L.J. Li, F.F. Li, ImageNet: a large-scale hierarchical image database, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), June Miami, Florida, USA, IEEE, 2009, , https://doi.org/10.1109/CVPR.2009.5206848.

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z.H. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, F.F. Li, Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252, https://doi.org/10.1007/s11263-015-0816-y.

[16] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, The 3rd International Conference on Learning Representations (ICLR) (2015), http://arxiv.org/abs/1409.1556v6.

[17] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 770–778, , https://doi.org/10.1109/CVPR.2016.90.

[18] Song Qing, Wu Yingqi, Xin Xueshi, Yang Lu, Yang Min, Chen Hongming, Liu Chun, Hu Mengjie, Li Jianchao, Real-time tunnel crack analysis system via deep learning, IEEE Access 7 (2019) 64186–64197, https://doi.org/10.1109/ACCESS.2019.2916330.

[19] D. Lee, J. Kim, D. Lee, Robust concrete crack detection using deep learning-based semantic segmentation, International Journal of Aeronautical and Space Sciences 20 (2019) 287–299, https://doi.org/10.1007/s42405-018-0120-5.

[20] D. Cao, L.D. Anh, Autonomous concrete crack detection using deep fully convolutional neural network, Autom. Constr. 99 (2018) 52–58, https://doi.org/10.1016/j.autcon.2018.11.028.

[21] Y.H. Liu, J. Yao, X.H. Lu, R.P. Xie, L. Li, DeepCrack: a deep hierarchical feature learning architecture for crack segmentation, Neurocomputing 338 (2019) 139–153, https://doi.org/10.1016/j.neucom.2019.01.036.

[22] Yue Fei, Kelvin C.P. Wang, Allen Zhang, Cheng Chen, Joshua Q. Li, Yang Liu, Guangwei Yang, Baoxian Li, Pixel-level cracking detection on 3D asphalt pavement images through deep-learning-based CrackNet-V, IEEE Trans. Intell. Transp. Syst. 21 (1) (2020) 273–284, https://doi.org/10.1109/TITS.2019.2891167.

[23] Y.J. Cha, W. Choi, Vision-based concrete crack detection using a convolutional neural network, Dynamics of Civil Structures 2 (2017) 71–73, https://doi.org/10.1007/978-3-319-54777-0_9.

[24] Ç.F. Özgenel, Concrete Crack Images for Classification, 1 Mendeley Data, 2018, https://doi.org/10.17632/5y9wdsg2zt.1.

[25] H.Y. Xu, X. Su, H.Y. Cai, Y. Wang, Automatic bridge crack detection using a convolutional neural network, Appl. Sci. 9 (14) (2019) 2867, https://doi.org/10.3390/app9142867.

[26] Mahtab Mohtasham Khani, Sahand Vahidnia, Leila Ghasemzadeh, Y. Eren Ozturk, Mustafa Yuvalaklioglu, Selim Akin, Nazim Kemal Ure, Deep-learning-based crack detection with applications for the structural health monitoring of gas turbines, Struct. Health Monit. (2019), https://doi.org/10.1177/1475921719883202.

[27] L.F. Li, W.F. Ma, L. Li, C. Lu, Research on detection algorithm for bridge cracks based on deep learning, Acta Automat. Sin. 45 (9) (2019) 1727–1742, https://doi.org/10.16383/j.aas.2018.c170052.

[28] K. Weiss, T.M. Khoshgoftaar, D.D. Wang, A survey of transfer learning, J. Big Data 3 (2016), https://doi.org/10.1186/s40537-016-0043-6.

[29] X. Glorot, A. Bordes, Y. Bengio, Domain adaptation for large-scale sentiment classification: a deep learning approach, Proceedings of the 28th International Conference on Machine Learning (ICML), Bellevue, Washington, USA, Omnipress, 2011, https://dl.acm.org/doi/10.5555/3104482.3104547.

[30] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks? Adv. Neural Inf. Proces. Syst. 27 (2014) 3320–3332 https://arxiv.org/abs/1411.1792.

[31] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7533) (2015) 436–444, https://doi.org/10.1038/nature14539.

[32] L. Zhang, Transfer adaptation learning: a decade survey, https://arxiv.org/abs/1903.04687, (2019).

[33] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359, https://doi.org/10.1109/TKDE.2009.191.

[34] Z. Chen-Mccaig, R. Hoseinnezhad, A. Bab-Hadiashar, Convolutional neural networks for texture recognition using transfer learning, International Conference on Control (ICCAIS), IEEE, Chiang Mai, 2017, pp. 187–192, , https://doi.org/10.1109/ICCAIS.2017.8217573.

[35] M. Mehdipour, B. Yanikoglu, E. Aptoula, Plant identification using deep neural networks via optimization of transfer learning parameters, Neurocomputing 235 (2017) 228–235, https://doi.org/10.1016/j.neucom.2017.01.018.

[36] X. Tan, Y. Liu, Y. Li, P. Wang, X. Zeng, F. Yan, X. Li, Localized instance fusion of MIR data of alzheimer's disease for classification based on instance transfer ensemble learning, Biomed. Eng. Online 17 (1) (2018) 49, https://doi.org/10.1186/s12938-018-0489-1.

[37] K. Zhang, H. Cheng, A novel pavement crack detection approach using pre-selection based on transfer learning, International Conference on Image and Graphics, 10666 Springer, Cham, 2017, pp. 273–283, , https://doi.org/10.1007/978-3-319-71607-7_24.

[38] F. Kucuksubasi, A. Sorguc, Transfer learning-based crack detection by autonomous UAVs, International Symposium on Automation and Robotics in Construction (ISARC), Berlin, German, 2018, https://doi.org/10.22260/ISARC2018/0081.

[39] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 2818–2826, , https://doi.org/10.1109/CVPR.2016.308.

[40] B.H. Kien, D. Iba, Y. Ishii, et al., Crack detection of plastic gears using a convolutional neural network pre-learned from images of meshing vibration data with transfer learning, Forsch. Ingenieurwes. 83 (2019) 645–653, https://doi.org/10.1007/s10010-019-00354-5.

[41] Mohamad Alipour, Devin K. Harris, Increasing the robustness of material-specific deep learning models for crack detection across different materials, Eng. Struct. 206 (2020) 110157, , https://doi.org/10.1016/j.engstruct.2019.110157.

[42] L. Zhang, F. Yang, Y.D. Zhang, Y.J. Zhu, Road crack detection using deep convolutional neural network, IEEE International Conference on Image Processing (ICIP), IEEE, Phoenix, AZ, 2016, pp. 3708–3712, , https://doi.org/10.1109/ICIP.2016.7533052.

[43] D. Sattar, 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 (2018) 1664–1668, https://doi.org/10.1016/j.dib.2018.11.015.

[44] M. Maguire, S. Dorafshan, R.J. Thomas, SDNET2018: A Concrete Crack Image Dataset for Machine Learning Applications. Browse all Datasets, (2018), p. 48, https://doi.org/10.15142/T3TD19.