| Title | Convolutional sparse coding-based deep random vector functional link network for distress classification of road structures |
|---|---|
| Author(s) | Maeda, Keisuke; Takahashi, Sho; Ogawa, Takahiro; Haseyama, Miki |
| Citation | Computer-aided civil and infrastructure engineering, 34(8), 654-676 https://doi.org/10.1111/mice.12451 |
| Issue Date | 2019-08 |
| Doc URL | http://hdl.handle.net/2115/79011 |
| Rights | This is the peer reviewed version of the following article: Computer-Aided Civil and Infrastructure Engineering, 34(8), 654-676 August 2019, which has been published in final form at https://onlinelibrary.wiley.com/doi/full/10.1111/mice.12451. This article may be used for non-commercial purposes in accordance with Wiley Terms and Conditions for Use of Self-Archived Versions.oses in accordance with Wiley Terms and Conditions for Self-Archiving. |
| Type | article (author version) |
| File Information | Word-ver-new.pdf |

# Convolutional Sparse Coding-based Deep Random Vector Functional Link Network for Distress Classification of Road Structures

Keisuke Maeda, Takahiro Ogawa, Miki Haseyama

*Faculty of Information Science and Technology, Hokkaido University, Kita-14, Nishi-9, Sapporo-shi, 060-0814 Japan*

&

Sho Takahashi

*Faculty of Engineering, Hokkaido University, Kita-13, Nishi-8, Sapporo-shi, 060-8628 Japan*

**Abstract:** *This paper presents a Convolutional Sparse Coding (CSC)-based Deep Random Vector Functional Link Network (CSDRN) for distress classification of road structures. The main contribution of this paper is the introduction of CSC into a feature extraction scheme in the distress classification. CSC can extract visual features representing characteristics of target images since it can successfully estimate optimal convolutional dictionary filters and sparse features as visual features by training from a small number of distress images. The optimal dictionaries trained from distress images have basic components of visual characteristics such as edge and line information of distress images. Furthermore, sparse feature maps estimated on the basis of the dictionaries represent both strength of the basic components and location information of regions having their components, and these maps can represent distress images. That is, sparse feature maps can extract key components from distress images that have diverse visual characteristics. Therefore, CSC-based feature extraction is effective for training from a limited number of distress images that have diverse visual characteristics. The construction of a novel neural network, CSDRN, by the use of a combination of CSC-based feature extraction and the DRN classifier, which can also be trained from a small dataset, is shown in this paper. Accurate distress classification is realized via the CSDRN.*

## 1 INTRODUCTION

Many road structures have been built worldwide, and degradation of such structures has been accelerating. Thus, the number of deteriorated road structures that must be inspected and repaired has been increasing. Since inspectors have been performing their operations based on their knowledge and experience, there is a possibility of human errors (Woo et al., 2016). Many studies on support of inspectors have been conducted with the aim of establishing methods for efficient and precise maintenance inspection (Bergquist and Söderholm, 2015; Xu et al., 2015). Recently, computer vision-based approaches have been studied (Koch et al., 2015; Tizani and Mawdesley, 2011), and it has been shown that there is a need for automatic detection and techniques for classification of distresses that have occurred in road structures (O 'Byrne et al., 2013; Zalama et al., 2014a). Since various kinds of distress occur in road structures (Li et al., 2018; Maeda et al., 2018a), the goal of our work is the realization of an accurate distress classification method based on machine learning techniques using images taken of distress parts of road structures.

In image recognition fields, convolutional neural networks (CNNs) (Krizhevsky et al., 2012), which require a large number of training images, have achieved outstanding results in various tasks such as image classification (Zhang et al., 2016; Zalama et al., 2014a; Koziarski and Cyganek, 2017). However, since the number of images of road structures is not sufficient for training CNNs, CNNs are not suitable for distress classification. Although fine-tuned CNNs have been proposed for a small dataset, it is difficult to use them since the visual characteristics of distress images are more diverse than those of images for generic object

recognition (Donahue et al., 2014; Zhou et al., 2014) as shown in Fig. 1.

To overcome this problem, a local receptive field-based neural network (LRF-NN) (Huang et al., 2015) has been proposed. The LRF-NN, which is a feedforward neural network, realizes convolutional feature extraction via LRFs and classification via a random-based neural network, which is an extended version of the random vector functional link (RVFL) network (Schmidt et al., 1992; Pao et al., 1994). Since almost all of the parameters in LRFs are determined randomly, that is, the number of parameters to be optimized is small, this network enables training from a small amount of training data. This is the advantage of the LRF-NN. Recent studies have shown that performance improvement for classification of images with diverse visual characteristics can be realized by using an LRF-based deep RVFL network (LRF-DRN) (Maeda et al., 2017a), which is our previous method, and the kernel version of the LRF-NN (KLRF-NN) (Shen et al., 2017). These methods are extended versions of the LRF-NN. However, the above feature extraction scheme, LRF, sets their convolutional filters to random values. This means that since the LRF is not trained from training samples, it cannot extract effective visual features. Therefore, realization of effective training from a small number of images with diverse visual characteristics is expected by adopting a new feature extraction scheme that can determine optimal convolutional filters.

It has been reported that a sparse coding approach, which is a traditional algorithm (Sarma and Adeli, 1995, 1996) that has been used for several problems, can represent characteristics of target images (Zhang et al., 2015). Specifically, sparse coding divides the target image into patches and reconstructs the patches by calculating the linear combination between dictionary filters obtained via dictionary learning and a sparse vector, which has many zero elements. Then the sparse coding exploits representation of the target patches. Thus, it has huge potential capabilities to solve many problems in a wide range of application fields, e.g., image denoising, inpainting and image classification. However, since general sparse coding represents these patches independently, it has difficulty in considering spatial information between the patches. Convolutional sparse coding (CSC), which can consider local interactions between patches by performing convolution of both dictionary filters and sparse feature maps, has been proposed to solve this problem (Wohlberg, 2016b; Zhang et al., 2017b). As shown in Fig. 2, traditional sparse coding approaches perform dictionary learning by using patches of images. However, since spatial phases of the patches are different, representation ability of the dictionaries calculated from these patches is limited. On the other hand, CSC calculates dictionaries with convolution between filters and images; in other words, it calculates dictionaries by sliding of images. Therefore, since CSC is robust for the difference of spatial phases, it is possible to calculate the features with high
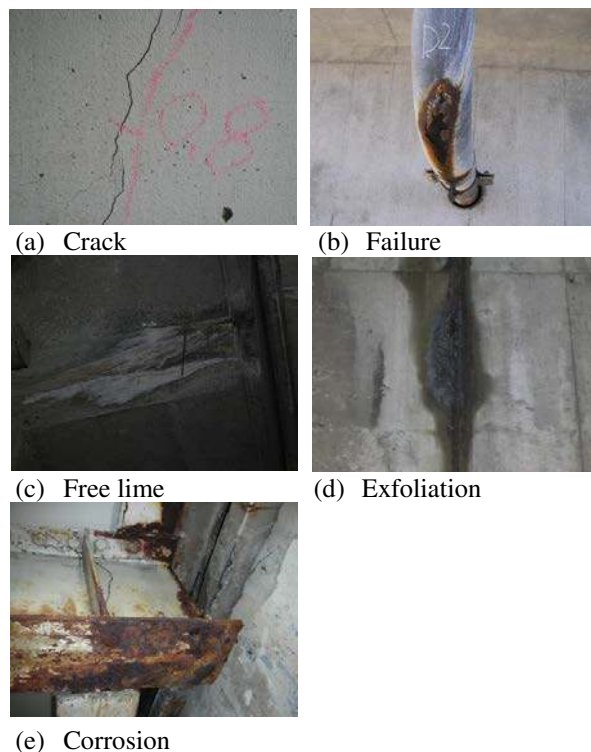


(a)  Crack                                    (b)  Failure

(c)  Free lime                              (d)  Exfoliation

(e)  Corrosion

**Figure 1** Examples of distress images.

representation ability. This is the reason why we apply CSC to distress classification. Since the number of parameters to be optimized is less than that in CNN approaches, dictionary learning can be easily performed from a small number of training distress images. Optimal dictionaries trained from images have basic components of visual characteristics such as edge and line information. Since sparse feature maps estimated on the basis of dictionaries represent both strength of the basic components and location information of regions having their components, sparse feature maps can extract key components from images. Therefore, it is expected that CSC can realize effective training from a small number of distress images with diverse visual characteristics.

In this paper, we present a novel distress classification method using a Convolutional Sparse Coding-based Deep Random Vector Functional Link Network (CSDRN). CSC, which enables extraction of visual features with higher representation ability and determination of optimal dictionary filters as convolutional filters, is newly introduced in our method. Specifically, the proposed method constructs a novel neural network, CSDRN, consisting of four layers (a CSC layer, a pooling layer, a local response normalization (LRN) layer and a DRN layer) that is constructed on the basis of a deep RVFL network (Cecotti, 2016). CSC-based feature extraction enables consideration local interactions such as spatial information of images more successfully than LRF can. Furthermore, since optimal dictionary filters can be estimated from a small dataset, CSC is suitable for a distress

**Table 1** Abbreviations of our paper.

| Abbreviations | Official name |
|---|---|
| CSDRN | Convolutional Sparse Coding-based Deep Random Vector Functional Link Network |
| CSC | Convolutional Sparse Coding |
| MCDL | Multi-channel Convolutional Dictionary Learning |
| CNN | Convolutional Neural Network |
| PCA | Principal Component Analysis |
| LRN | Local Response Normalization |
| DFT | Discrete Fourier Transform |
| ADMM | Alternating Direction Method of Multipliers |
| RVFL | Random Vector Functional Link |
| LRN | Local Receptive Field |
| DRN | Deep Random Vector Functional Link Network |
| KLRF-NN | Kernel version of Local Receptive Field-based Neural Network |
| LRF-DRN | Local Receptive Field-based Deep Random Vector Functional Link Network |
| LRF-NN | Local Receptive Field-based Neural Network |
| SVM | Support Vector Machine |
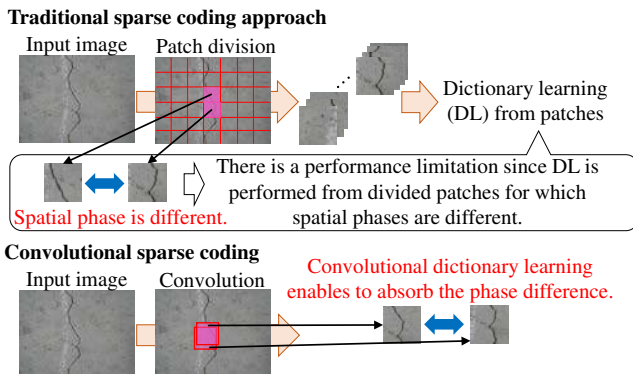


**Figure 2** Difference between traditional sparse coding and convolutional sparse coding.

classification task. Moreover, since parameters of the hidden layers of the DRN are determined by an auto-encoder-based method, that is, the number of DRN's parameters to be optimized is much less than that in general deep learning methods, the DRN classifier can also be trained from a small dataset. Thus, since CSC-based feature extraction has affinity for a DRN, the use of the combination of CSC and DRN is suitable for distress classification. In terms of stability, since CNNs also have problems such as overfitting and vanishing gradient, parameter tuning is difficult. On the other hand, although RVFL has a disadvantage due to random parameters, it is possible to efficiently search parameters with faster computation. Thus, since the introduction of RVFL is one of the reasonable solutions, the CSC and DRN are newly combined in the proposed method for improving the simple RVFL-based classifiers. Therefore, improvement of distress classification performance is realized by using CSDRN.

The difference between the proposed method and our previous methods (Maeda et al., 2017b, 2018b) is explained below. Maeda et al. (2017b) proposed an approach based on

the Bayesian network, which is one of probabilistic models. In addition, Maeda et al. (2018b) collaboratively used text features and visual features based on canonical correlation analysis (Hotelling, 1936), which maximizes the canonical correlation of multi-bivariates. The main contribution of our approach is the use of the combination of CSC and DRN. Our previous method (Maeda et al., 2017a) focused on LRF and DRN. LRF was used as the feature extractor. However, LRF is not suitable for feature extraction due to the random-based approach. Thus, LRF is replaced by CSC, which can extract more effective visual features, in the proposed method. Each element of our architecture has been previously proposed. However, by simple combination use of CSC and DRN, we have derived a new framework that can deal with a small amount of data. Thus, our approach solves the current problem in computing in the field of civil engineering. This is a strong contribution. For more details, although the original CSC has been proposed for image classification, only sparse feature maps were used; in other words, introduction of multiple pooling and normalization has not been proposed. Moreover, although general deep learning approaches prevent over-fitting by using dropout in fully connected layers, this problem can also be solved by sparse coding. The proposed method can also solve the over-fitting problem. Consequently, the proposed architecture is completely different from our previous methods.

Section 2 shows related works. In Section 3, the construction of the CSDRN is explained. Section 4 shows experimental results. Finally, Section 5 shows conclusions. In order to simply explain our paper, Table 1 shows abbreviations.
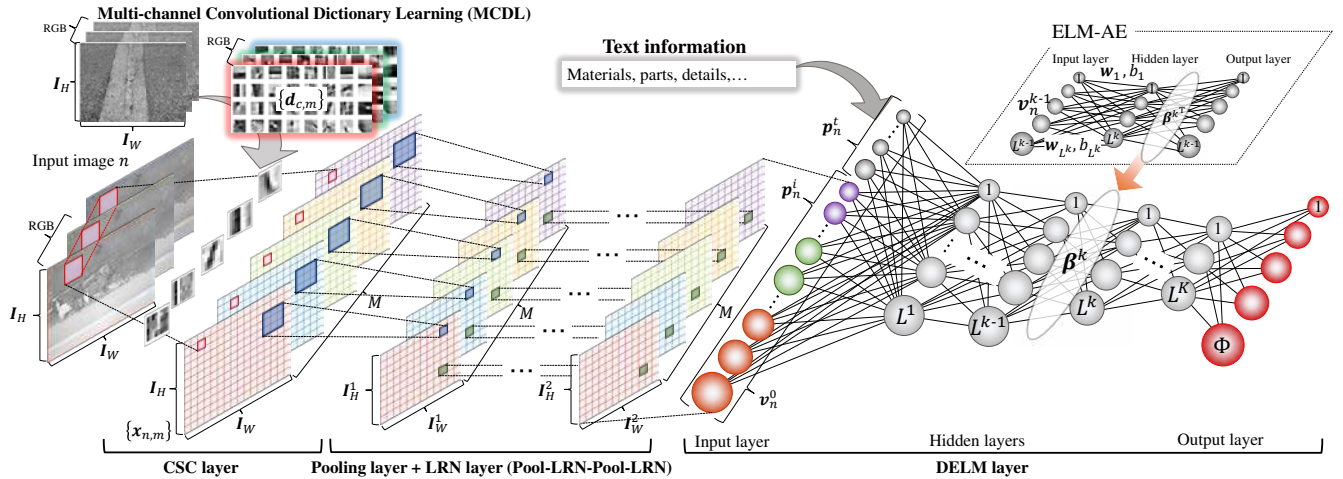
## 2 Related works

**Figure 3** An overview of the CSDRN. The CSDRN is constructed from four layers. The CSC layer estimates sparse feature maps via optimal dictionary filters obtained by Multi-channel Convolutional Dictionary Learning (MCDL) (3.1). The pooling layer performs the pooling operation, and the LRN layer performs local response normalization between channels for contrast adjustment (3.2). Finally, the DRN layer performs classification using the visual and text features (3.3).

In this section, we explain related works of computing in civil engineering. In this research area, the methods used in many studies can be roughly divided into deep learning-based methods and non-deep learning-based methods.

Recently, although deep learning-based approaches have been used to solve several problems in the field of civil engineering, some problems are complex, and more optimal approaches than deep learning-based approaches have sometimes been used (Wang et al., 2018; Poskus et al., 2018; Huangpeng et al., 2018; Liao, 2017; Amezquita-Sanchez and Adeli, 2015b; Qarib and Adeli, 2015). For example, Wang et al. (2018) developed an automatic method for measuring crack width using binary crack map images. They introduced a new crack width definition and formulated it using the Laplace equation so that crack width can be continuously and unambiguously measured, and they showed that their method was effective for characterizing propagation behavior of cracks with small widths. Although cracks are one of the most important types of distress, methods for detecting other types of distress have also been studied (Poskus et al., 2018; Huangpeng et al., 2018). Huangpeng et al. (2018) focused on the use of texture features to construct prior maps for distress detection. They formulated a detection process as a novel weighted low-rank reconstruction model with a texture prior map and improved the detection performance of state-of-the-art detection methods. Moreover, a solution to the vehicle routing problem has been proposed by using computational approaches (Liao, 2017). Liao (2017) developed on-line VRP because it is more appropriate for logistics operations in which customer service is requested in real time with a meta-heuristic algorithm. Thus, computational approaches have been used to solve various problems.

On the other hand, recent computational methods have used deep learning-based approaches to solve many problems such as crack detection and distress classification. Since deep learning-based methods generally require a large number of labeled training images for parameter tuning, high-resolution images were prepared and divided into patches in some studies (Zhang et al., 2016; Cha et al., 2017, 2018; Xue and Li, 2018; Chen and Jahanshahi, 2018; Gibert et al., 2017; Rafiei et al., 2017). By using the obtained patches, training of CNNs becomes feasible. Various crack detection methods have been proposed (Zhang et al., 2016; Chen and Jahanshahi, 2018; Cha et al., 2017). Zhang et al. (2016) used 3,264×2,448 original images and divided the images into 640,000 patches. Chen and Jahanshahi (2018) and Cha (2017) used about 300,000 and 40,000 patches, respectively. They constructed their own CNN architectures and outperformed other classifiers such as a support vector machine (SVM) (Cortes and Vapnik, 1995) and traditional crack detection methods such as a canny algorithm. High-resolution images were also used for region-based distress detection methods (Xue and Li, 2018; Cha et al., 2018) inspired by a fully convolutional network (Long et al., 2015) and fast R-CNN (Girshick, 2015). Although the number of distress images is often not sufficient for training CNNs, the use of high-resolution images provides a solution to introduce CNNs into the field of civil engineering.

However, there are cases in which it is difficult to prepare not only high-resolution images but also labeled training images. Other than research on hardware (Ortega-Zamorano et al., 2017), there are mainly four approaches based on software in such cases. Firstly, there are methods combining a conventional signal processing method and deep learning

**Table 2** A part of the inspection items of the inspection data. Inspection data include distress images and inspection items. Text data such as "Damaged parts", "Categories of structure" and "Details of structure" are recorded. IDs 1-5 correspond to Figs. 1 (a)-(e). For example, the variables "Abutment (front)", "Abutment", ... and "Concrete substructure" are recorded as ID 1 corresponding to Fig. 1 (a).

| ID | Distress | Inspection items | | | |
| | | Damaged parts | Categories of structure | ... | Details of structure |
|---|---|---|---|---|---|
| 1 | Crack | Abutment (front) | Abutment | ... | Concrete substructure |
| 2 | Failure | Road shoulder (right) | Bridge drainpipe | ... | Bridge drainage facility |
| 3 | Free lime | Stretch (right) | RC slab | ... | Concrete superstructure |
| 4 | Exfoliation | Abutment (front) | Abutment | ... | Concrete substructure |
| 5 | Corrosion | Main girder flange | Steel girder | ... | Steel structure |

(Doulamis et al., 2018; Rafiei and Adeli, 2018, 2017) and methods devising preprocessing of inputs to the CNNs (Zhang et al., 2017a; Nabian and Meidani, 2018). Rafiei and Adeli proposed methods for health monitoring of large structures through integration of synchrosqueezed wavelet transform, fast Fourier transform and unsupervised deep Boltzmann machine. In addition, Amezquita-Sanchez, J. and Adeli, H. (2015a) presented a review journal in order to assess the health condition of a structure (Amezquita-Sanchez and Adeli, 2015a). Nabian and Meidani (2018) and Zhang (2017a) generated input features for deep learning by using their own feature extraction method. Secondly, Lin et al. (2017) created fake data based on a true data set and added the fake data to the training set (Lin et al., 2017). Thus, deep neural networks can be trained from a small amount of original data and a large amount of fake data. This approach means "data augmentation". Thirdly, active learning and weakly supervised approaches have been developed. Feng et al. (2017) developed a distress detection method based on active learning. In the initial phase of the method, only a small number of labeled training images are given, resulting in a distress classifier with poor performance. Although performing poorly, this classifier can filter out many non-distress images. Then the most difficult cases are sent to human experts for ground truth labels, and these images are added to train CNNs. This approach greatly improved classification performance. Xia (2018) pre-extracted objects from original images, and humans assigned each object to a certain class manually. Xia claimed that the approach is weakly-supervised labeling (Xia, 2018). Finally, many researchers have focused on a transfer learning approach (Gao and Mosalam, 2018; Gopalakrishnan et al., 2018, 2017). The transfer learning approach is used for two different strategies: feature extraction and fine-tuning. In the feature extraction, a target image is input into the pre-trained CNN, and outputs of the middle layer of the network are used as visual features. Thus, traditional classifiers such as SVM are constructed by using the obtained visual features. Furthermore, the fine-tuning approach retrains the networks with a target image

dataset by using pre-determined parameters as initial parameters. These approaches are effective for solving the problem of a shortage of labeled training data. However, there are cases in which transfer learning is difficult with data in the field of civil engineering since the visual characteristics of distress images are more diverse than those of images for generic object recognition such as ImageNet. To overcome this problem, a method that can fully train from original distress images without patch division and transfer learning is presented in this paper.

## 3 DISTRESS CLASSIFICATION VIA CSDRN

Distress classification based on the CSDRN is explained in this section. Figure 3 shows an overview of the CSDRN that consists of four layers: CSC layer, pooling layer, LRN layer and DRN layer. In order to construct the CSDRN, we first estimate optimal dictionary filters obtained via Multi-channel Convolutional Dictionary Learning (MCDL) (Wohlberg, 2016a) and sparse feature maps in the CSC layer. We construct MCDL and CSC inspired by (Wohlberg, 2016a,b). The optimal dictionary filter is constructed so as to minimize the least square error between the original image and the reconstructed image based on sparse representation. Details of construction of the CSC layer are shown in 3.1. Construction of the pooling and LRN layers is explained in 3.2, and construction of the DRN layer is explained in 3.3. Since it was revealed in our previous studies (Maeda et al., 2017b, 2018b) that the inspection items shown in Table 2 are effective for distress classification, we utilize not only visual features but also text features from the inspection data and input these features into the DRN classifier. Finally, procedures of the test phase of the CSDRN are shown in 3.4.

### 3.1 Construction of the CSC Layer

In order to estimate dictionary filters $\{\boldsymbol{d}_{c,m}\}$ ($c \in C$ and $m = 1, 2, \ldots, M$; $C = \{R, G, B\}$ and $M$ being the number of

dictionaries) as convolutional filters and enable estimation of sparse feature maps $\{x_{n,m}\}$ from an image $n$ with $I_H \times I_W$ pixels[1], we solve the following optimization problem:

$$\arg \min_{\{d_{c,m}\},\{x_{n,m}\}} \frac{1}{2}\sum_{n=1}^{N}\left\{\sum_{c\in C}\left\|\sum_{m=1}^{M}\{d_{c,m} * x_{n,m} - s_{n,c}\}\right\|_2^2\right\}$$
$$+ \lambda \sum_{n=1}^{N}\left\{\sum_{m=1}^{M}\|x_{n,m}\|_1\right\} \qquad (1)$$
$$\text{s.t.}\,\|d_{c,m}\|_2 = 1\,\,\forall c,m,$$

where $x_{n,m} \in \mathbb{R}^{I_H I_W}$ is a sparse vector and $d_{c,m}$ is the $m$-th dictionary vector for channel $c$. $N$ is the number of training images. Furthermore, $s_{n,c} \in \mathbb{R}^{I_H I_W}$ is an original signal of a target image $n$, and $*$ denotes the convolution operator. $\lambda$ is a regularization parameter. Equation (1) represents a summary of dictionary learning and sparse coding processes. The right term in this equation is for sparse representation. The optimization of Eq. (1) can be divided into an update process of the dictionary filters $\{d_{c,m}\}$ and an update process of the sparse feature maps $\{x_{n,m}\}$. Although a basic optimization process of Eq. (1) is the same as the process proposed by Wohlberg et al. (Wohlberg, 2016b), the update process of dictionary filters is slightly different from their process. Dictionary learning is generally performed by using only one target image. However, we need to extract basic components of visual characteristics from various kinds of distress in order to effectively classify the distress images. Thus, we optimize the problem by using $N$ distress images in MCDL. The details of these two update procedures are shown below. Since the rationale for CSC is fundamental, we moved the rationale to a chapter of the appendix. We have to make $d_{c,m} * x_{n,m}$ and $s_{n,c}$ all the same size, and it is necessary to be in the order of $d_{c,m} * x_{n,m}$.

### 3.1.1 Process for update of dictionary filters

In MCDL, the dictionary filters $\{d_{c,m}\}$ are estimated by solving the following optimization problem:

$$\arg \min_{\{d_{c,m}\}} \frac{1}{2}\sum_{n=1}^{N}\sum_{c\in C}\left\|\sum_{m=1}^{M} x_{n,m} * d_{c,m} - s_{n,c}\right\|_2^2$$
$$+ \sum_{c\in C}\sum_{m=1}^{M} \ell_{C_{QN}}(g_{c,m}) \qquad (2)$$
$$\text{s.t.}\,\,d_{c,m} - g_{c,m} = 0\,\,\forall c,m,$$

where $C_{QN} = \{d_{c,m} \in \mathbb{R}^{I_H I_W}: (I - QQ^\top)d_{c,m} = 0, \|d_{c,m}\|_2 = 1\}$, $Q$ is a zero-padding operator, $\{g_{c,m}\}$ is an auxiliary variable, and $\ell_{C_{QN}}$ is an indicator function.

$$\ell_{C_{QN}}(g_{c,m}) = \begin{cases} 0 & \text{if } g_{c,m} \in C_{QN}, \\ \infty & \text{if } g_{c,m} \notin C_{QN}. \end{cases} \qquad (3)$$

$C_{QN}$ means "C"onstraint, zero-padding operator "Q" and "N"ormalization, respectively. Note that since "P" is used as feature vectors, we used "Q" for the zero-padding operator. Although there exists $x_{n,m} * d_{c,m}$ in Eq. (2), since $d_{c,m}$ in Eq. (2) is redefined by performing zero-padding, it is necessary to be in the order of $x_{n,m} * d_{c,m}$. MCDL is formulated on the basis of (Wohlberg, 2016a). MCDL is an extended version of single-channel convolutional dictionary learning dealing with gray scale images. Equation (2) shows that MCDL separately extracts dictionaries for each color channel and shares single sparse representation, that is, it can consider inter-channel statistical dependencies. The details of the solution of the above problem are shown in the appendix. Finally, we can obtain $\hat{d}$ by solving the linear system problem as follows:

$$\left(\sum_{n=1}^{N} \hat{\Omega}_n^H \hat{\Omega}_n + \sigma I\right)\hat{d} = \sum_{n=1}^{N} \hat{\Omega}_n^H \hat{s}_n + \sigma \hat{e}. \qquad (4)$$

The dictionary filter $d$ can be obtained by computing the inverse DFT of $\hat{d}$. Since dictionary filters can be estimated from many distress images, they have basic components of distress images.

### 3.1.2 Process for update of sparse feature maps

In order to estimate sparse feature maps, we solve the following optimization problem:

$$\arg \min_{\{x_{n,m}\}} \frac{1}{2}\sum_{c\in C}\left\|\sum_{m=1}^{M} d_{c,m} * x_{n,m} - s_{n,c}\right\|_2^2 + \lambda\sum_{m=1}^{M}\|x_{n,m}\|_1. \qquad (5)$$

The details of the solution of the above problem are shown in the appendix. Finally, we can obtain $\hat{x}$ by solving the linear system problem as follows:

$$\left(\sum_{c\in C} \hat{D}_c^H \hat{D}_c + \rho I\right)\hat{x}_n = \sum_{c\in C} \hat{D}_c^H \hat{s}_{n,c} + \rho \hat{z}_n. \qquad (6)$$

The sparse feature vector $x_n$ can be obtained by computing the inverse DFT of $\hat{x}_n$.

We can perform MCDL with a limited number of distress images and calculate the $M$ optimal dictionaries trained from all kinds of distress. These dictionaries have various basic components of visual characteristics of distress images, and the kinds of distress that each dictionary can represent are different. On the other hand, since the sparse feature maps estimated on the basis of dictionaries represent both strength

---

[1] For notational simplicity, each of the $\{x_{n,m}\}$ is considered to be an $I_H I_W$ dimensional vector.

of the basic components and location information of regions having their components, these maps can represent distress images that have diverse visual characteristics. Since each sparse feature map corresponds to each dictionary, the kinds of distress that each sparse feature map can represent are also different as is the case in the dictionary. Thus, the sparse features have discriminant ability to classify distresses. Therefore, the sparse feature maps are effective for representing distresses.

The solution of the CSC model used in the proposed method is based on (Wohlberg, 2016b). We describe the differences of other CSC models below. The method proposed in (Bristow et al., 2013) uses the ADMM algorithm (Gabay and Mercier, 1975) as a solver for the optimization problem and divides the original problem into subproblems. However, Bristow et al. did not use an efficient method for solving the subproblem, a linear system. Although our method inspired from (Wohlberg, 2016b) also uses the ADMM algorithm, we perform effective optimization for solving the subproblem based on the Sherman-Morrison approach. In addition, an extended version of CSC for improving its robustness has been proposed in (Heide et al., 2015). They argued that it was difficult to work with incomplete data by using a general CSC due to the boundary artifact. They therefore added a filter handling the boundaries into the objective function. Since this approach cannot use an effective frequency-based optimization approach, this is a big difference. Since general CSC-based methods are time-consuming tasks due to inversion of a linear operator related to convolution, Šorel and Šroubek proposed how these inversions can be computed non-iteratively in the Fourier domain using the matrix inversion lemma. This method led to efficient optimization. Our method uses an efficient algorithm based on not the approach proposed in (Šorel and Šroubek, 2016) but the Sherman-Morrison approach.

LRF calculates the vectors based on random values from the uniform distribution and then orthogonalizes these vectors. Finally, LRF generates filters via matricization of the obtained orthogonal vectors. In these three steps of LRF, there are two drawbacks: 1) since LRF uses random values, feature extraction suitable for the target data is difficult and 2) since LRF simply matricizes the orthogonal vectors, LRF cannot consider the spatial relationship between elements in the calculated filter. On the other hand, CSC can generate optimal filters via training from the target data. Furthermore, since CSC performs dictionary learning based on the convolution operation, it can consider the spatial relationship. From the above points, CSC solves the problems of LRF, and effective feature extraction is realized via CSC.

### 3.2 Construction of the Pooling Layer and LRN Layer

We use pooling and LRN layers to obtain transformation-invariant features and perform contrast adjustment between

Examples of inspection data.
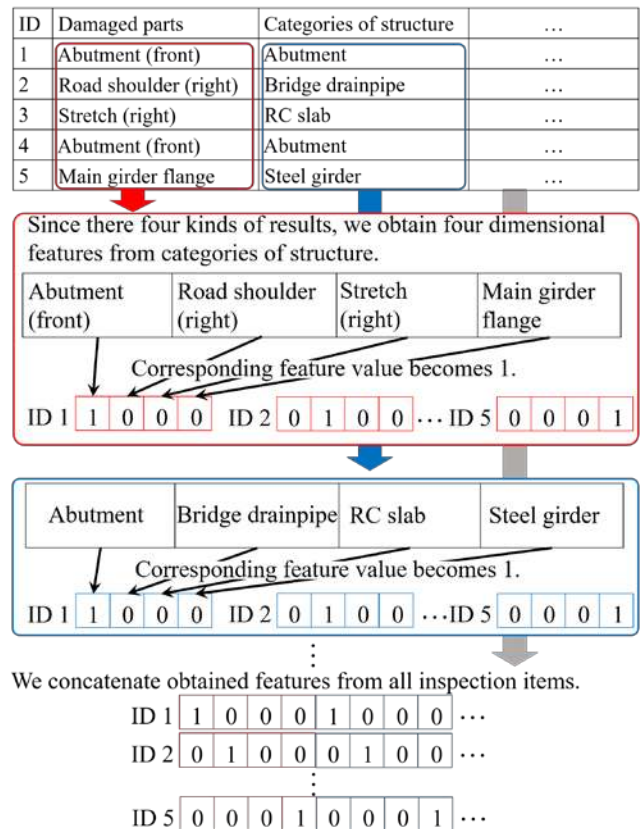


**Figure 4** Details of the encoding approach for the inspection data.

**Table 3** Dataset used in the experiment.

| Distress | Training data | | Test data |
| --- | --- | --- | --- |
| | Validation (train) | Validation (test) | |
| Crack | 400 | 170 | 180 |
| Failure | 400 | 170 | 160 |
| Free lime | 400 | 170 | 181 |
| Exfoliation | 400 | 170 | 152 |
| Corrosion | 400 | 170 | 169 |
| Sum | 2,850 | | 842 |

channels of input images. In the proposed method, we use two sets of these layers, that is, we construct pool-LRN-pool-LRN layers. In the first pooling layer, we apply the $e_p^1 \times e_p^1$ pooling operation with the sliding interval $e_s^1$ to the input map. Since it has been reported that max pooling is effective for sparse features (Yang et al., 2009; Wang et al., 2010), it is also used in the proposed method. Thus, we obtain the first pooling map $\widetilde{x}_{n,m}^1 \in \mathbb{R}^{I_H^1 I_W^1}$. Furthermore, for performing contrast adjustment between channels of input images, we calculate the first LRN map $\overline{x}_{n,m}^1 \in \mathbb{R}^{I_H^1 I_W^1}$ by considering the local response normalization, which has been used in

**Table 4** Relation between predicted class and true class. "TP" is the number of images correctly classified as the target class. "TN" is the number of images correctly classified as not the target class. "FP" is the number of images misclassified as the target class. "FN" is the number of images misclassified as not the target class.

| | | Predicted class | |
|---|---|---|---|
| | | Target class | Not target class |
| *True class* | Target class | True positive (TP) | False negative (FN) |
| | Not target class | False positive (FP) | True negative (TN) |

(Krizhevsky et al., 2012). In the second pooling layer, we apply max pooling to the obtained $\bar{x}_{n,m}^1$ and normalize it. Then we obtain $\bar{x}_{n,m}^2 \in \mathbb{R}^{I_H^2 I_W^2}$ from the second LRN layer.

The max pooling operation realizes not only calculation of transformation-invariant features but also emphasis of activated pixels. Furthermore, the LRN operation adjusts the contrast of pooling maps. Consequently, transformation-invariant features with higher representation ability can be obtained via the pooling and normalization.

### 3.3 Construction of the DRN Layer

The visual feature vector $\bar{x}_n^2 \in \mathbb{R}^{I_H^2 I_W^2 M}$ is obtained by aligning the LRN maps $\bar{x}_{n,m}^2$. We also obtain the visual feature $p_n^i \in \mathbb{R}^{d_i}$ by performing PCA (Wold et al., 1987) to reduce the dimension since $I_H^2 I_W^2 M$ is a high dimension. Furthermore, we calculate text features $p_n^t \in \mathbb{R}^{d_t}$ of a training image *n* from inspection data shown in Table 2. Note that $d_t$ is the dimension of $p_n^t$. All elements of text feature vectors are binary values. Specifically, although the original text data contain many inspection items such as "details of structures", "inspection date" and "a remark column", many of these items have defects, i.e., blank space due to omission. Thus, the proposed method used inspection items that have no defects. We selected six inspection items from inspection data. By using these selected inspection items, we could obtain text features. The details of the calculation of the text features are shown below. Specifically, our encoding approach for the text features is shown in Fig. 4. As shown in the column of "categories of structure", since there are four kinds of inspection results, we obtain four-dimensional features for this item. ID1 has an abutment as "categories of structure", and the corresponding feature value becomes 1. Otherwise, it becomes 0. In this way, we search all inspection items and assign feature values. Finally, we obtain the text feature by concatenating the calculated features. Thus, we obtain feature vectors $p_n^i$ and $p_n^t$. Furthermore, defining a feature vector $v_n^0 = \left[ p_n^{i\top}, p_n^{t\top} \right]^\top \in \mathbb{R}^{d_i + d_t}$, we train the DRN classifier. As shown in Fig. 3, DRN has *K+2* layers including *K* hidden layers. In the training phase of DRN, we calculate an output weight matrix between the output layer and the *K*th hidden layer. DRN consists of feature transformation and classification. When *k* is smaller than *K+1*, auto-encoder-based feature transformation (Cambria et al., 2013) is performed. When *k* is equal to *K+1*, the classification is

performed via the RVFL-based neural network. The above procedures are explained in detail in the appendix.

Parameters of the hidden layers of DRN are determined via the auto-encoder; that is, the number of DRN's parameters to be optimized is less than that in general deep learning methods, and the DRN classifier can also be trained from a small dataset. Furthermore, since the least-squares approach can solve an objective function of RVFL, the optimal weight can be estimated even if the number of training images is small. Therefore, DRN is suitable for distress classification. Consequently, due to the introduction of the regularization term in the CSC and DRN, we can determine a unique solution. Therefore, over-fitting does not occur in the proposed method.

Furthermore, the CSC layer does not use the nonlinear activation function. In contrast, in the DRN layer, nonlinear feature transformation is carried out by using the activation function for each layer. In other words, non-linear transformation of the DRN layer enables compensation for the linear transformation in CSC. Thus, since there is high affinity for the CSC and RVFL series, our framework, CSDRN, is an effective approach.

### 3.4 Test Phase of CSDRN

Given a new test image, the sparse feature maps $\{x_m\}$ are estimated in the CSC layer including the estimated optimal dictionary filters in the same manner as that in Eq. (5). Next, by performing pooling and normalization, we can obtain an LRN map. By applying PCA-based projection obtained in the training phase to the LRN map, we can obtain the feature vector $p^i$. Furthermore, we calculate the text feature $p^t$ corresponding to the target image. Finally, by using $v^0 = \left[ p^{i\top}, p^{t\top} \right]^\top$ and the obtained $\beta^k (k = 2,3,...,K+1)$, the output value $f = [f_1, f_2, ..., f_\Phi]^\top$ is obtained as $f = \beta^{K+1} v^K$. Furthermore, the final classification result corresponds to the index of the output node that has the highest value. Consequently, distress classification is realized by the DRN classifier using sparse features having strong representation ability via CSC.

The differences between the other CSC-based methods (Chen et al., 2016; Yu and Sun, 2017; Luo et al., 2017) and the proposed method are described below. In the method proposed by Chen et al., filters are constructed for each class using training data. Next, when a test image is given,

**Table 5** Methods used in the experiment. "C", "P" and "N" mean CSC, pooling and LRN layers, respectively.

| Methods | Details | Feature | Classifier |
|---|---|---|---|
| Proposed method | CSDRN | CSC (C-P-N-P-N) | DRN |
| Comparative method 1 | CSDRN | CSC (C-P-N) | DRN |
| Comparative method 2 | CSRN | CSC (C-P-N) | RVFL |
| Comparative method 3 | LRF-DRN (Maeda et al., 2017a) | LRF | DRN |
| Comparative method 4 | Kernel LRF-NN (Shen et al., 2017) | LRF | Kernel RVFL |
| Comparative method 5 | LRF-NN (Huang et al., 2015) | LRF | RVFL |
| Comparative method 6 | Inception v3-based RVFL | Inception-v3 | RVFL |
| Comparative method 7 | Fine-tuned VGG-based CNN | VGG16 (Simonyan and Zisserman, 2014) | |
| Comparative method 8 | Text-based RVFL | Text features only | RVFL |
| Comparative method 9 | DenseNet-based RVFL | DenseNet-201 | RVFL |
| Comparative method 10 | Inception-ResNet-v2-based RVFL | Inception-ResNet-v2 | RVFL |
| Comparative method 11 | ResNet50-based RVFL | ResNet50 | RVFL |
| Comparative method 12 | VGG16-based RVFL | VGG16 | RVFL |
| Comparative method 13 | Xception-based RVFL | Xception | RVFL |
| Comparative method 14 | DenseNet-based DRN | DenseNet-201 | DRN |
| Comparative method 15 | Inception-ResNet-v2-based DRN | Inception-ResNet-v2 | DRN |
| Comparative method 16 | Inception-v3-based DRN | Inception-v3 | DRN |
| Comparative method 17 | ResNet50-based DRN | ResNet50 | DRN |
| Comparative method 18 | VGG16-based DRN | VGG16 | DRN |
| Comparative method 19 | Xception-based DRN | Xception | DRN |
| Comparative method 20 | Fine-tuned DenseNet-based CNN | DenseNet-201 (Huang et al., 2017) | |
| Comparative method 21 | Fine-tuned Inception-ResNet-v2-based CNN | Inception-ResNet-v2 (Szegedy et al., 2017) | |
| Comparative method 22 | Fine-tuned Inception-v3-based CNN | Inception-v3 (Szegedy et al., 2016) | |
| Comparative method 23 | Fine-tuned ResNet50-based CNN | ResNet50 (He et al., 2016) | |
| Comparative method 24 | Fine-tuned Xception-based CNN | Xception (Chollet, 2017) | |

dictionary learning and sparse coding are performed by using only the test image. By using sparse features obtained from the test image and dictionary filters trained for each class, reconstructed images for each class are obtained. Finally, the class to minimize the error between the original image and the reconstructed image is adopted as a final result. The number of unnecessary components such as a background in the dataset used in (Chen et al., 2016) is small. Therefore, it is a possible to perform sufficient training by using linear transformation only via CSC. On the other hand, since images used in the proposed method are more complex, non-linear transformation is required. Thus, we can solve the problem by inputting the sparse features obtained from CSC to DRN, which is a fully connected layer including a nonlinear transformation.

Yu and Sun focused on the theory that neurons in each layer are sparse from a physiological point of view. They therefore proposed a novel RVFL classifier including a sparse hidden layer. Specifically, since it is difficult to perform sparse coding of the hidden layer by using random-based weights, their method combines minimization of both the least squares error and L1 regularization as the objective function. The goal of the proposed method is to perform sparse coding of input features. Thus, there is a notable difference between the approach of their technique and the proposed method. Moreover, in their approach, it is necessary to set some kinds

of features as input features; in other words, classification results depend on the characteristics of the input features. However, since the proposed method enables input features to be adaptively determined from input images, the obtained results are robust.

Luo et al. achieved a reduction in computational costs by using a convolutional sparse auto-encoder, which is proposed in their paper. Moreover, dictionaries calculated via CSC were set to the initial filters of a two or three-layered CNN, and they examined the use of a combination of CNN and CSC. However, although they used a large amount of general data, the performance of their approach was close to that of AlexNet (Krizhevsky et al., 2012), which is a traditional CNN. On the other hand, the proposed method constructs a new framework considering characteristics of the data used in the field of civil engineering by combining CSC and RVFL, which can be constructed by using a small amount of data.

## 4 EXPERIMENTAL RESULTS

This section verifies that the CSDRN is effective. In 4.1, experimental settings are described. Evaluation of the performance of the proposed method is explained in 4.2. Finally, the effectiveness of our feature extraction approach via CSC is discussed in 4.3.

## 4.1 Experimental Settings

We evaluated the performance of the distress classification by using inspection data provided by East Nippon Expressway Company Limited. This company is called NEXCO. In this experiment, we examined five kinds of distress, crack, failure, free lime, exfoliation and corrosion, as shown in Fig. 1. We used 3,692 samples. The details of the dataset are shown in Table 3. We divided the dataset into training data and test data for performance evaluation. The proposed method randomly decided training data and test data from the entire dataset provided by NEXCO in order to fairly consider the spatial characteristics. We also divided the training data into training and test validation data for parameter settings. The previous work (Liang, 2018) trained deep networks (not scratch but fine-tuning) from hundreds of images. The method classifies whether major failure of structures exists or not in the image classification step. Since this classification task is a simple binary classification problem, the authors of the above reference claim that "hundreds of images" means "a small dataset". However, since our classification task is a more complex multi-class classification problem, we can regard "thousands of images" as "a small amount of training data". Note that since general multi-class classification tasks need hundreds of thousands of images (Krizhevsky et al., 2012), we can regard our dataset as "a small amount of training data". The images used in this experiment were taken manually by inspectors in the daytime at bridges and tunnels along an expressway. Inspectors take distress images at difference angles. The images are taken at close and distant positions, but images taken from a distant position are difficult to analyze. Therefore, we used only images taken from a close position in this experiment. In addition, the resolution of all images is $640 \times 480$ pixels. We resized the original images to $160 \times 120$ pixels for calculating MCDL and CSC. High-resolution images can also be applied to the CSC. However, since MCDL requires a large amount of memory when there are many training images, the number of training images should be reduced if high-resolution images are used. On the other hand, if distress regions are small such as regions with cracks, distresses cannot be visually recognized in low-resolution images, and more than a quarter size of the original image is required.

Recall, Precision and F-measure were used in this experiment. They are defined as follows:

$$\text{Recall} = \frac{TP}{TP+FN}, \tag{7}$$

$$\text{Precision} = \frac{TP}{TP+FP}, \tag{8}$$

$$\text{F-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}. \tag{9}$$

where the relationships between "TP", "TN", "FP" and "FN" are shown in Table 4. In this experiment, we used eight comparative methods shown in Table 5. Comp. 1 is a CSDRN with one set of pooling and LRN layers. Comp. 2 is a method replacing the DRN in comp. 1 with a general RVFL. Comps. 3 and 4 are state-of-the-art methods for a small number of training images with diverse visual characteristics. Comps. 6 and 7 are general deep learning methods. Comp. 6 uses the Inception-v3 model (Szegedy et al., 2016) for visual feature extraction, and comp. 7 uses a fine-tuned CNN only when outputs of the text-based RVFL are lower than a pre-determined threshold value. All of the methods except for comps. 7 and 8 use visual and text features by concatenating those features. Comp. 8 uses text features $\boldsymbol{p}^t$ only. Furthermore, we extracted other advanced deep learning-based features and constructed the RVFL-based neural network and DRN (comps. 9-19). In addition, we performed fine-tuning of the above advanced deep learning methods (comps. 20-24).

The details of parameters used in the proposed method are shown in Table 6. We selected six inspection items from inspection data. The number of results obtained from the six inspection items is 129. The details of calculation of the text features are shown in subsection 3.3. In (Šorel and Šroubek, 2016), $M$ is set to 32. In addition, since the convergence of the MCDL becomes severe by increasing $M$, we experimentally determine $M$. By changing the value $M$, further improvement in accuracy is expected. Note that the hidden nodes of RVFL and DRN were set to values in such a way that each classification performance becomes the best by applying the validation dataset to each method as shown in Table 3. Since the computational cost of performing MCDL increases as the number of images increases, we used 50 training images to perform MCDL in this experiment. Specifically, ten training images for each class were used. On the other hand, in all procedures without MCDL, all of the training images were used. We determined the parameters $C_1, C_2$, $K$ and $L$ by applying a grid search to the validation data. In addition, we determined the dictionary size of CSC in the same manner as (Wohlberg, 2016b). Since other parameters were set experimentally, performance improvement is expected by searching their optimal values. Since the size of input images is different for each framework, we optimally resized the input images for each deep learning method in order to perform fair experiments. On the other hand, since it is necessary to reduce the size by considering parameter convergence, the image size was experimentally set in the proposed method. In other words, since we set optimal parameters for each deep learning technique, we can fairly show comparable experimental results. With respect to other parameters, we performed a grid search in all methods by using validation data. Therefore, the performance comparison is fair for all methods.

Since the combination use of CSC and DRN is a novel framework, comp. 1 can be regarded as a part of the proposed method. We explain the reason for performing the pooling-

norm twice. As can be seen from CNNs, in a general classification problem, classification performance is improved by constructing robust activation maps. Since CSC performs image reconstruction so that the square mean error is minimized, that is, the error is calculated based on each pixel, calculated feature maps are sensitive to a shift and distortion. Thus, sparse feature maps need to be updated by a pooling approach for improving robustness of the shift and distortion. Furthermore, since the calculated sparse feature maps via CSC have very high sparsity, we cannot obtain sufficient robustness by using only single pooling. Therefore, pooling-normalization is used twice in the proposed method. In fact, since an inception-based network, which is one of the strong deep neural networks, also continuously uses two poolings, this approach is reasonable.

## 4.2 Performance Evaluation

Table 7 shows Recall, Precision and F-measure. In the averages of all measurements, the proposed method outperforms all comparative methods including the state-of-the-art methods comps. 3 and 4. Specifically, since the performance of CSDRN and comp. 2 is higher than that of comps. 3 and 5, respectively, the sparse features estimated from the optimal dictionary filters can effectively represent the characteristics of distresses. Furthermore, although it has been reported that visual features obtained from the Inception-v3 model have higher representation ability for generic object recognition, the performance of comp. 6 is lower than that of CSC-based methods (our method and comps. 1 and 2) for distress classification. Thus, feature extraction via CSC is effective for images with diverse visual characteristics since CSC can adaptively train visual features with consideration of characteristics of the target dataset. RVFL is a three-layered neural network that consists of input, hidden and output layers. DRN is an extended version of RVFL. Specifically, DRN is constructed by inserting multiple hidden layers into RVFL. The proposed method outperforms comp. 2, and DRN is more effective than RVFL. Furthermore, since the proposed method outperforms comp. 3, the advantage of the CSC-based feature extraction can be confirmed. Moreover, since the proposed method outperforms the fine-tuned CNN (Simonyan and Zisserman, 2014) for a small number of training images, CSC-based feature extraction is also effective for a small dataset. As shown in Table 7, the performance for failure and corrosion is very high. This is because text information has a great influence on performance improvement. These distresses occur in specific parts of structures. For example, corrosion often occurs in metal parts, not in concrete. A comparison of the performance of the proposed method and that of comp. 1 shows that the use of multiple sets of pooling and LRN layers is effective. By performing transfer learning based on various deep learning architectures, it is verified that the performance of the proposed method is higher than that of advanced deep

**Table 6** Performance settings of our method.

| Details | Parameter | Value |
|---|---|---|
| Size of input image | $I_H$ | 120 |
| | $I_W$ | 160 |
| Num. of dictionaries | $M$ | 32 |
| Dictionary filter size | No listed | 8×8 |
| Dictionary filter's sliding interval | No listed | 1 |
| Pooling size | $e_p^1$ | 5 |
| | $e_p^2$ | 3 |
| Pooling sliding interval | $e_s^1$ | 5 |
| | $e_s^2$ | 2 |
| Num. of dimensions | $d_i$ | 367 |
| | $d_t$ | 129 |
| Regularization parameters | $\lambda$ | 0.02 |
| | $C_1$ | $2^5$ |
| | $C_2$ | $2^5$ |
| ADMM parameters | $\sigma$ | 50 |
| | $\rho$ | 1.5 |
| Sparsity parameter (DRN) | $\eta$ | 0.05 |
| Num. of hidden layers | $K$ | 2 |
| Num. of hidden nodes | $L^0$ | 496 |
| | $L^1$ | 1488 |
| | $L^2$ | 744 |
| Num. of classes | $\Phi$ | 5 |

learning methods. Moreover, in (Maeda et al., 2017a), it was shown that LRF-DRN improves SVM by comparing comp. 3 with SVM. Furthermore, since the proposed method outperforms comp. 3, it can be seen that our method is more effective than SVM. Consequently, since the proposed method achieves the best performance, the effectiveness of the CSDRN is verified.

We calculated the training times and test times for all of the methods as shown in Tables 8 and 9. The computational cost of MCDL is $7.53 \times 10^3$ sec, and the costs of the fine-tuned CNNs (comps. 7 and 20-24) are about $2.00 \times 10^4$ sec. Note that comps. 20-24 are constructed on the basis of DenseNet201, InceptionResNet-v2, Inception-v3, ResNet50 and Xception, respectively. In addition, since our method outperforms these fine-tuned CNNs, it is verified that effective dictionary filters can be calculated with low computational cost. However, the total cost of our method is higher than that of the fine-tuned CNNs. Since it is necessary to solve the ADMM for calculating MCDL and CSC, the calculational cost becomes high. In the test phase, CSC-based approaches have higher costs than the costs of other methods. This is the drawback of our method. There are some parameters determined experimentally in the proposed method. Among them, the number of dictionaries and dictionary filter size have an influence on the computational costs. Therefore, reduction of the computational costs is expected by determining these parameters optimally. We

should search these parameters in future works. However, MCDL can optimize parameters more effectively than can transfer learning-based approaches, and the proposed method is superior to these methods in terms of classification performance.

There is little difference between the performance of comp. 1 and that of our method. In order to statistically validate the effectiveness of the proposed method, we applied Welch's t-test (Welch, 1938) to the results of 100 trials by using comp. 1 and the proposed method as shown in Table 10. Since it was statistically significant (significance level at 0.01), the proposed method is effective. Verification using other datasets will be performed in future works.

### 4.3 Discussion

We discuss the effectiveness of our feature extraction via CSC in this subsection. An original training image and sparse feature maps of the image are shown in Fig. 5. Figure 5 (a) is an original image, and Figs. 5 (b), (c) and (d) are sparse feature maps obtained from CSC and from the first and second LRN layers, respectively. Specifically, the $m$-th feature map $x_{n,m}$ of the training image n is estimated via CSC by using $m$-th dictionaries $d_{c,m}$. We manually select the map that has the largest number of non-zero pixels among all maps (right images in Figs. 5 (b), (c) and (d)). Furthermore, we subjectively select the feature map significantly representing cracks among the feature maps (left images in Figs. 5 (b), (c) and (d)). Since something similar to a heat map of a CNN does not exist in CSC, images in which distress regions had been activated were manually determined as an effective feature map. It was verified that some feature maps calculated from only a small number of images via CSC can extract characteristics of distress images. This provides interpretation of classification results. It is thought that automatic determination of effective feature maps will become feasible by improving CSC approaches.

Feature maps in Fig. 5 show the merit of pooling. The left images of Figs. 5 (c) and (d) can represent characteristics of cracks more strongly than can the left image of Fig. 5 (b). Thus, the use of pooling and LRN procedures is effective for obtaining characteristics of distresses. Firstly, there is one feature map that has many non-zero elements in all of the maps estimated via the CSC layer as shown in Fig. 5 (b). Since the 30th feature map contributes to minimization of the difference between the original image and the reconstructed image by considering the whole image, this map cannot represent detailed characteristics of distresses. On the other hand, the others can represent these characteristics. For example, Fig. 5 (b) shows that not the 30th map but the 18th map represents the crack. However, since almost all of the feature maps are too sparse to represent the characteristics of distresses, the number of feature maps representing these characteristics is small. Thus, we apply pooling and normalization to feature maps obtained from the CSC layer.



(a) Original image



(b) Examples of feature maps estimated via the CSC layer



(c) Examples of feature maps calculated via the first LRN layer



(d) Examples of feature maps calculated via the second LRN layer
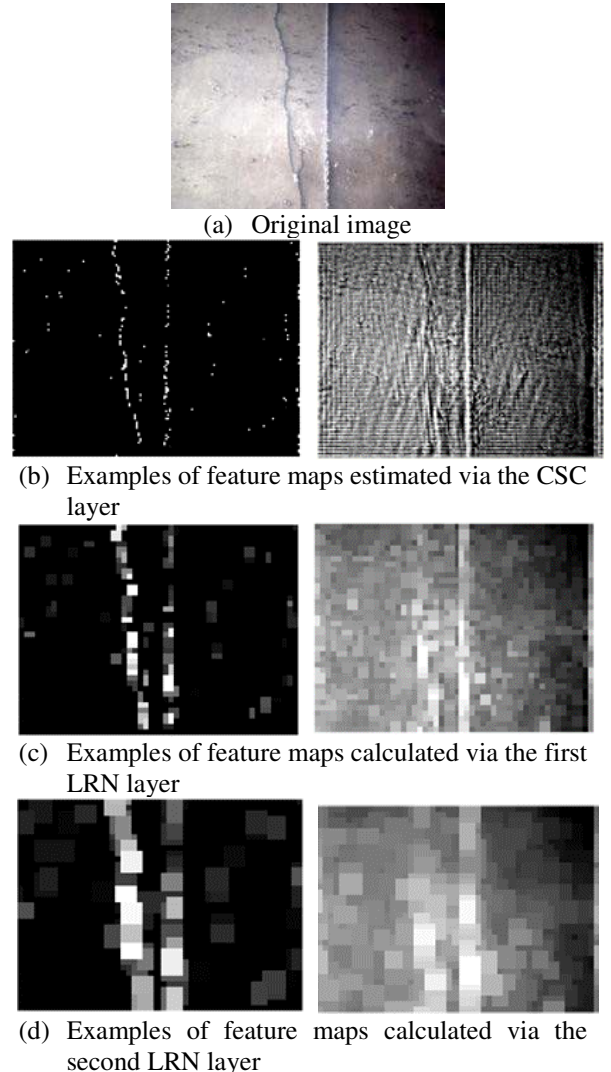
**Figure 5** Examples of an original training image and sparse feature maps of the image. The left images of (b), (c) and (d) are the 18th feature maps. The right images of (b), (c) and (d) are the 30th feature maps.

As shown in Figs. 5 (c) and (d), the feature maps can clearly represent distresses. Furthermore, by comparing Fig. 5 (c) with Fig. 5 (d), it is confirmed that feature maps in Fig. 5 (d) calculated from two sets of pooling and LRN layers can emphasize pixels of these feature maps. Therefore, the use of multiple sets of these layers is effective for improving the performance of distress classification.

Secondly, it is confirmed that feature maps representing the characteristics of distresses are different depending on the kind of distress as shown in Fig. 6. We explain their differences below.

· Crack (Figure 6 (a))
  The 17th - 23rd feature maps have characteristics of cracks such as long and faint patterns.

**Table 7** Classification performance of all methods.

| | Proposed method | | | Comp. 1 | | | Comp. 2 | | | Comp. 3 | | | Comp. 4 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Distresses | Re | Pr | Fm | Re | Pr | Fm | Re | Pr | Fm | Re | Pr | Fm | Re | Pr | Fm |
| Crack | 0.911 | 0.906 | 0.908 | 0.905 | 0.895 | 0.900 | 0.911 | **0.916** | **0.913** | 0.888 | 0.879 | 0.883 | 0.894 | 0.865 | 0.879 |
| Failure | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** |
| Free lime | 0.828 | **0.882** | **0.854** | 0.801 | 0.868 | 0.833 | 0.801 | 0.838 | 0.819 | 0.790 | 0.846 | 0.817 | 0.773 | 0.843 | 0.806 |
| Exfoliation | **0.894** | 0.839 | **0.866** | 0.888 | 0.823 | 0.854 | 0.861 | 0.813 | 0.837 | 0.855 | 0.807 | 0.830 | 0.842 | 0.795 | 0.817 |
| Corrosion | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | 0.994 | 0.997 | **1.000** | **1.000** | **1.000** |
| Average | **0.926** | **0.925** | **0.925** | 0.918 | 0.917 | 0.917 | 0.914 | 0.913 | 0.913 | 0.906 | 0.905 | 0.905 | 0.902 | 0.900 | 0.900 |

| | Comp. 5 | | | Comp. 6 | | | Comp. 7 | | | Comp. 8 | | | Comp. 9 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Distresses | Re | Pr | Fm | Re | Pr | Fm | Re | Pr | Fm | Re | Pr | Fm | Re | Pr | Fm |
| Crack | 0.888 | 0.860 | 0.874 | 0.900 | 0.866 | 0.882 | 0.855 | 0.855 | 0.855 | 0.816 | 0.880 | 0.847 | 0.777 | 0.765 | 0.771 |
| Failure | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | 0.952 | 0.975 | **1.000** | **1.000** | **1.000** | 0.968 | 0.981 | 0.974 |
| Free lime | 0.767 | 0.812 | 0.789 | 0.790 | 0.812 | 0.801 | 0.767 | 0.847 | 0.805 | 0.773 | 0.790 | 0.782 | 0.696 | 0.707 | 0.701 |
| Exfoliation | 0.789 | 0.769 | 0.779 | 0.802 | 0.813 | 0.807 | 0.782 | 0.815 | 0.798 | 0.782 | 0.704 | 0.741 | 0.717 | 0.668 | 0.692 |
| Corrosion | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | 0.918 | 0.957 | **1.000** | **1.000** | **1.000** | 0.923 | 0.975 | 0.948 |
| Average | 0.889 | 0.888 | 0.888 | 0.898 | 0.898 | 0.898 | 0.881 | 0.877 | 0.878 | 0.874 | 0.875 | 0.874 | 0.816 | 0.819 | 0.817 |

| | Comp. 10 | | | Comp. 11 | | | Comp. 12 | | | Comp. 13 | | | Comp. 14 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Distresses | Re | Pr | Fm | Re | Pr | Fm | Re | Pr | Fm | Re | Pr | Fm | Re | Pr | Fm |
| Crack | 0.872 | 0.892 | 0.882 | 0.872 | 0.902 | 0.887 | 0.9 | 0.870 | 0.885 | 0.894 | 0.865 | 0.879 | 0.872 | 0.853 | 0.862 |
| Failure | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** |
| Free lime | 0.779 | 0.787 | 0.783 | 0.767 | 0.798 | 0.783 | 0.795 | 0.867 | 0.829 | 0.767 | 0.837 | 0.801 | 0.762 | 0.821 | 0.790 |
| Exfoliation | 0.822 | 0.796 | 0.809 | 0.815 | 0.756 | 0.784 | 0.868 | 0.825 | 0.846 | 0.842 | 0.800 | 0.820 | 0.815 | 0.770 | 0.792 |
| Corrosion | **1.000** | 0.994 | 0.997 | **1.000** | 0.994 | 0.997 | **1.000** | 0.994 | 0.997 | **1.000** | 0.994 | 0.997 | 0.994 | 0.994 | 0.994 |
| Average | 0.894 | 0.894 | 0.894 | 0.891 | 0.890 | 0.890 | 0.912 | 0.911 | 0.911 | 0.900 | 0.899 | 0.899 | 0.888 | 0.887 | 0.887 |

| | Comp. 15 | | | Comp. 16 | | | Comp. 17 | | | Comp. 18 | | | Comp. 19 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Distresses | Re | Pr | Fm | Re | Pr | Fm | Re | Pr | Fm | Re | Pr | Fm | Re | Pr | Fm |
| Crack | 0.888 | 0.869 | 0.879 | 0.872 | 0.887 | 0.879 | 0.861 | 0.911 | 0.885 | 0.888 | 0.903 | 0.896 | 0.883 | 0.850 | 0.866 |
| Failure | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | 0.993 | 0.996 | **1.000** | **1.000** | **1.000** |
| Free lime | 0.779 | 0.787 | 0.783 | 0.812 | 0.790 | 0.801 | 0.828 | 0.824 | 0.826 | **0.856** | 0.842 | 0.849 | 0.751 | 0.819 | 0.783 |
| Exfoliation | 0.782 | 0.793 | 0.788 | 0.789 | 0.800 | 0.794 | 0.861 | 0.813 | 0.837 | 0.848 | 0.848 | 0.848 | 0.776 | 0.732 | 0.753 |
| Corrosion | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | 0.994 | **1.000** | 0.997 | 0.994 | **1.000** | 0.997 |
| Average | 0.890 | 0.890 | 0.890 | 0.894 | 0.895 | 0.895 | 0.910 | 0.909 | 0.909 | 0.917 | 0.917 | 0.917 | 0.881 | 0.880 | 0.880 |

| | Comp. 20 | | | Comp. 21 | | | Comp. 22 | | | Comp. 23 | | | Comp. 24 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Distresses | Re | Pr | Fm | Re | Pr | Fm | Re | Pr | Fm | Re | Pr | Fm | Re | Pr | Fm |
| Crack | 0.855 | 0.793 | 0.823 | 0.900 | 0.760 | 0.824 | **0.916** | 0.778 | 0.841 | 0.927 | 0.625 | 0.747 | 0.866 | 0.764 | 0.812 |
| Failure | **1.000** | **1.000** | **1.000** | **1.000** | 0.975 | 0.987 | **1.000** | 0.946 | 0.972 | **1.000** | 0.869 | 0.930 | **1.000** | 0.969 | 0.984 |
| Free lime | 0.834 | 0.736 | 0.782 | 0.823 | 0.741 | 0.780 | 0.740 | 0.817 | 0.776 | 0.563 | 0.822 | 0.668 | 0.834 | 0.736 | 0.782 |
| Exfoliation | 0.519 | 0.822 | 0.637 | 0.546 | **0.932** | 0.688 | 0.644 | 0.875 | 0.742 | 0.493 | 0.789 | 0.607 | 0.532 | 0.890 | 0.666 |
| Corrosion | **1.000** | 0.903 | 0.949 | **1.000** | 0.965 | 0.982 | **1.000** | 0.913 | 0.954 | **1.000** | 0.982 | 0.991 | **1.000** | 0.954 | 0.976 |
| Average | 0.841 | 0.851 | 0.838 | 0.853 | 0.875 | 0.852 | 0.860 | 0.866 | 0.857 | 0.796 | 0.817 | 0.788 | 0.846 | 0.863 | 0.844 |

· Failure (Figure 6 (b))

Failure often occurs in drainpipes. Many feature maps represent these target objects.

· Free lime (Figure 6 (c))

Compared with cracks, failures and exfoliation, free lime

occurs in a large area. These characteristics can be monitored in the 4th, 11th, 25th and 26th feature maps.

· Exfoliation (Figure 6 (d))

Exfoliation often occurs in a small area, and it is represented in the 4th, 8th, 11th, 12th and 26th feature maps.

**Table 8** Computational costs (sec) of the training procedure. "PM" and "C1" mean the proposed method and comparative method 1, respectively.

| Methods | MCDL | CSC | P+N | LRF | Text | CNN feat | RVFL | KRVFL | DRN | Fine-tune | Sum |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PM | $7.53\times10^3$ | $3.13\times10^4$ | $2.55\times10^2$ | - | - | - | - | - | $3.91\times10^1$ | - | $3.91\times10^4$ |
| C1 | $7.53\times10^3$ | $3.13\times10^4$ | $2.14\times10^2$ | - | - | - | - | - | $1.58\times10^3$ | - | $4.06\times10^4$ |
| C2 | $7.53\times10^3$ | $3.13\times10^4$ | $2.14\times10^2$ | - | - | - | $1.12\times10^2$ | - | - | - | $3.91\times10^4$ |
| C3 | - | - | - | $2.53\times10^2$ | - | - | - | - | $2.86$ | - | $2.55\times10^2$ |
| C4 | - | - | - | $2.47\times10^2$ | - | - | - | $2.86\times10^{-1}$ | - | - | $2.47\times10^2$ |
| C5 | - | - | - | $1.94\times10^2$ | - | - | $8.69\times10^1$ | - | - | - | $2.81\times10^2$ |
| C6 | - | - | - | - | - | $5.72\times10^1$ | $1.06\times10^2$ | - | - | - | $1.63\times10^2$ |
| C7 | - | - | - | - | - | - | - | - | - | $1.21\times10^4$ | $1.21\times10^4$ |
| C8 | - | - | - | - | $9.73\times10^{-2}$ | - | $1.35$ | - | - | - | $1.44$ |
| C9 | - | - | - | - | - | $1.68\times10^2$ | $1.70\times10^2$ | - | - | - | $3.39\times10^2$ |
| C10 | - | - | - | - | - | $1.64\times10^2$ | $6.29\times10^1$ | - | - | - | $2.27\times10^2$ |
| C11 | - | - | - | - | - | $6.71\times10^1$ | $6.61\times10^1$ | - | - | - | $1.33\times10^2$ |
| C12 | - | - | - | - | - | $6.21\times10^1$ | $1.34\times10^2$ | - | - | - | $1.96\times10^2$ |
| C13 | - | - | - | - | - | $7.14\times10^1$ | $1.57\times10^2$ | - | - | - | $2.29\times10^2$ |
| C14 | - | - | - | - | - | $1.68\times10^2$ | - | - | $4.34$ | - | $1.73\times10^2$ |
| C15 | - | - | - | - | - | $1.64\times10^2$ | - | - | $1.74$ | - | $1.66\times10^2$ |
| C16 | - | - | - | - | - | $7.41\times10^1$ | - | - | $2.98$ | - | $7.70\times10^1$ |
| C17 | - | - | - | - | - | $6.71\times10^1$ | - | - | $3.63$ | - | $7.05\times10^1$ |
| C18 | - | - | - | - | - | $6.21\times10^1$ | - | - | $5.60$ | - | $6.77\times10^1$ |
| C19 | - | - | - | - | - | $7.14\times10^1$ | - | - | $4.26$ | - | $7.57\times10^1$ |
| C20 | - | - | - | - | - | - | - | - | - | $2.06\times10^4$ | $2.06\times10^4$ |
| C21 | - | - | - | - | - | - | - | - | - | $2.62\times10^4$ | $2.62\times10^4$ |
| C22 | - | - | - | - | - | - | - | - | - | $1.58\times10^4$ | $1.58\times10^4$ |
| C23 | - | - | - | - | - | - | - | - | - | $1.70\times10^4$ | $1.70\times10^4$ |
| C24 | - | - | - | - | - | - | - | - | - | $2.30\times10^4$ | $2.30\times10^4$ |

**Table 9** Computational costs (sec) of the test procedure for all test images. "PM" and "C1" mean the proposed method and comparative method 1, respectively.

| Methods | CSC | P+N | LRF | Text | CNN feat | RVFL | KRVFL | DRN | Fine-tune | Sum |
|---|---|---|---|---|---|---|---|---|---|---|
| PM | $9.33\times10^3$ | $7.54\times10^1$ | - | - | - | - | - | $3.57\times10^{-1}$ | - | $9.41\times10^3$ |
| C1 | $9.33\times10^3$ | $6.33\times10^1$ | - | - | - | - | - | $4.92$ | - | $9.40\times10^3$ |
| C2 | $9.33\times10^3$ | $6.33\times10^1$ | - | - | - | $2.18$ | - | - | - | $9.40\times10^3$ |
| C3 | - | - | $7.35\times10^1$ | - | - | - | - | $4.90\times10^{-2}$ | - | $7.35\times10^1$ |
| C4 | - | - | $7.17\times10^1$ | - | - | - | $4.88\times10^{-2}$ | - | - | $7.17\times10^1$ |
| C5 | - | - | $5.69\times10^1$ | - | - | $8.90\times10^{-1}$ | - | - | - | $5.78\times10^1$ |
| C6 | - | - | - | - | $1.69\times10^1$ | $1.34$ | - | - | - | $1.82\times10^1$ |
| C7 | - | - | - | - | - | - | - | - | $2.11\times10^1$ | $2.11\times10^1$ |
| C8 | - | - | - | $2.87\times10^{-2}$ | - | $6.00\times10^{-2}$ | - | - | - | $8.87\times10^{-2}$ |
| C9 | - | - | - | - | $4.99\times10^1$ | $2.58$ | - | - | - | $5.24\times10^1$ |
| C10 | - | - | - | - | $4.85\times10^1$ | $6.50\times10^{-1}$ | - | - | - | $4.92\times10^1$ |
| C11 | - | - | - | - | $1.98\times10^1$ | $6.20\times10^{-1}$ | - | - | - | $2.04\times10^1$ |
| C12 | - | - | - | - | $1.83\times10^1$ | $1.67$ | - | - | - | $2.00\times10^1$ |
| C13 | - | - | - | - | $2.11\times10^1$ | $2.09$ | - | - | - | $2.32\times10^1$ |
| C14 | - | - | - | - | $4.99\times10^1$ | - | - | $5.19\times10^{-2}$ | - | $4.99\times10^1$ |
| C15 | - | - | - | - | $4.85\times10^1$ | - | - | $2.21\times10^{-2}$ | - | $4.85\times10^1$ |
| C16 | - | - | - | - | $7.41\times10^1$ | - | - | $3.74\times10^{-2}$ | - | $7.41\times10^1$ |
| C17 | - | - | - | - | $1.98\times10^1$ | - | - | $5.07\times10^{-2}$ | - | $1.98\times10^1$ |
| C18 | - | - | - | - | $1.83\times10^1$ | - | - | $7.08\times10^{-2}$ | - | $1.84\times10^1$ |
| C19 | - | - | - | - | $2.11\times10^1$ | - | - | $4.60\times10^{-2}$ | - | $2.11\times10^1$ |
| C20 | - | - | - | - | - | - | - | - | $6.37\times10^1$ | $6.37\times10^1$ |
| C21 | - | - | - | - | - | - | - | - | $8.76\times10^1$ | $8.76\times10^1$ |
| C22 | - | - | - | - | - | - | - | - | $5.69\times10^1$ | $5.69\times10^1$ |
| C23 | - | - | - | - | - | - | - | - | $2.82\times10^1$ | $2.82\times10^1$ |
| C24 | - | - | - | - | - | - | - | - | $3.34\times10^1$ | $3.34\times10^1$ |

・Corrosion (Figure 6 (e))

An area in which corrosion occurs is the largest among all distresses. Furthermore, corrosion occurs in various forms since the distress depends on the form of the structure. Thus, various feature maps are activated.

From the above, it is confirmed that feature maps representing cracks, free lime and exfoliation are different as shown by color rectangles in Fig. 6. Since something similar to a heat map of a CNN does not exist in CSC, we manually determine feature maps that are activated in distress regions as effective feature maps. It was verified that some feature maps calculated from only a small number of images via CSC can extract characteristics of distress images. This provides interpretation of classification results. On the other hand, failure and corrosion are represented by various feature maps. Thus, the CSC-based feature extraction contributes to improvement in the performance of classification of cracks, free lime and exfoliation; that is, dictionaries effectively representing distresses are different for each distress as shown in Fig. 7. Figure 7 shows dictionaries of a red channel obtained via MCDL, and effective dictionaries are surrounded by color rectangles by reference to Fig. 6. In free lime and exfoliation, the 4th, 11th and 26th feature maps are activated as shown in Fig. 6. Although their corresponding dictionaries represent both characteristics of free lime and exfoliation, it is considered that CSC cannot provide strongly discriminant features that distinguish these distresses. Therefore, misclassification of these distresses occurs. Thus, the performance for these distresses is lower than that for other distresses. Although there is no remarkable difference between feature maps of failure and corrosion and those of other distresses, the proposed method maintains high performance. Focusing on the results of comp. 3 in Table 7, LRF-based features have an adverse effect on classification performance for corrosion. There is a possibility that the CSC-based features have an affinity for the text features. Therefore, by considering the above discussion, CSC-based feature extraction is effective. Distress size depends on imaging processes, and the tendency for exfoliation to be small and corrosion to be large is sometimes not observed. However, since the imaging process is unified among all inspectors, that tendency can be seen in many images. In future works, it is necessary to consider the influence of the imaging processes.

The proposed method misclassified some test images in the experiment. We explain why these images were misclassified in terms of CSC-based feature extraction. Figure 8 shows examples of misclassified images. Crack, free lime and exfoliation images were misclassified as exfoliation, crack and free lime, respectively. Although there is a crack in the center of the original image in Fig. 8 (a), feature maps do not represent any characteristics of a crack. Generally, the aim of CSC is minimization of the error between original and reconstructed images. Since the original image in Fig. 8 (a) has various patterns and darkening of concrete, CSC estimates

sparse maps so that it effectively reconstructs these patterns and darkening. Thus, the image was misclassified as another distress. Furthermore, as shown in feature maps in Fig. 8 (b), there are long and faint lines such as cracks, but the image in Fig. 8 (b) was misclassified as crack. Moreover, as shown in Fig. 8 (c), the 4th, 11th and 26th maps have characteristics of exfoliation, but they also represent other textures in a large area. Thus, this image was misclassified as free lime. The proposed model depends on the quality of images. This is a limitation of the proposed method. On the other hand, in this situation, not only visual features but also multimodal features such as text features are effective. In the next step in our research, we will consider image quality. On the other hand, examples of a part of the classification results obtained from comps. 1 and 18 and the proposed method show that the performance of these comparative methods is similar to that of the proposed method. However, a comparison with training results obtained from the training data is meaningless. Therefore, we compared the results of the test. Figure 9 shows examples of images classified by these three methods. The proposed method correctly classified all images, and comp. 18 incorrectly classified all images. In addition, the image shown in Fig. 9 (c) was incorrectly classified by comp. 1. As shown in Figs. 9 (a) and (c), CSC-based methods can classify slight cracks and complex exfoliation. Since Fig. 9 (b) is a corrosion image, text features are effective for classification. However, comp. 18 misclassified the image. Thus, it was verified that the use of a combination of text features and Inception-v3-based visual features is not suitable.

From the above discussion, the effectiveness of CSC-based feature extraction is verified. However, in future works, we will try to extend this feature extraction approach in order to reconstruct distress images with focus on target distresses even if there are various patterns and noise in original images. Specifically, construction of discriminative CSC by introducing class information into CSC would be effective for solving the problem.

## 5 CONCLUSIONS

A novel distress classification method via a CSDRN is proposed in this paper. The CSDRN can extract visual features with higher representation ability since MCDL can estimate optimal dictionary filters from distress images. Furthermore, by training the DRN classifier based on the obtained effective visual features, a novel neural network can be constructed. According to the results of experiments, the CSDRN has higher computational costs than those of comparative methods including advanced deep learning methods. For example, fine-tuned CNNs are trained about twice as fast as the proposed method. Furthermore, in the test phase, the costs of CSC-based methods are much higher than those of other methods, and this is a drawback of the proposed method. However, MCDL can optimize parameters more effectively than can transfer learning-based approaches, and

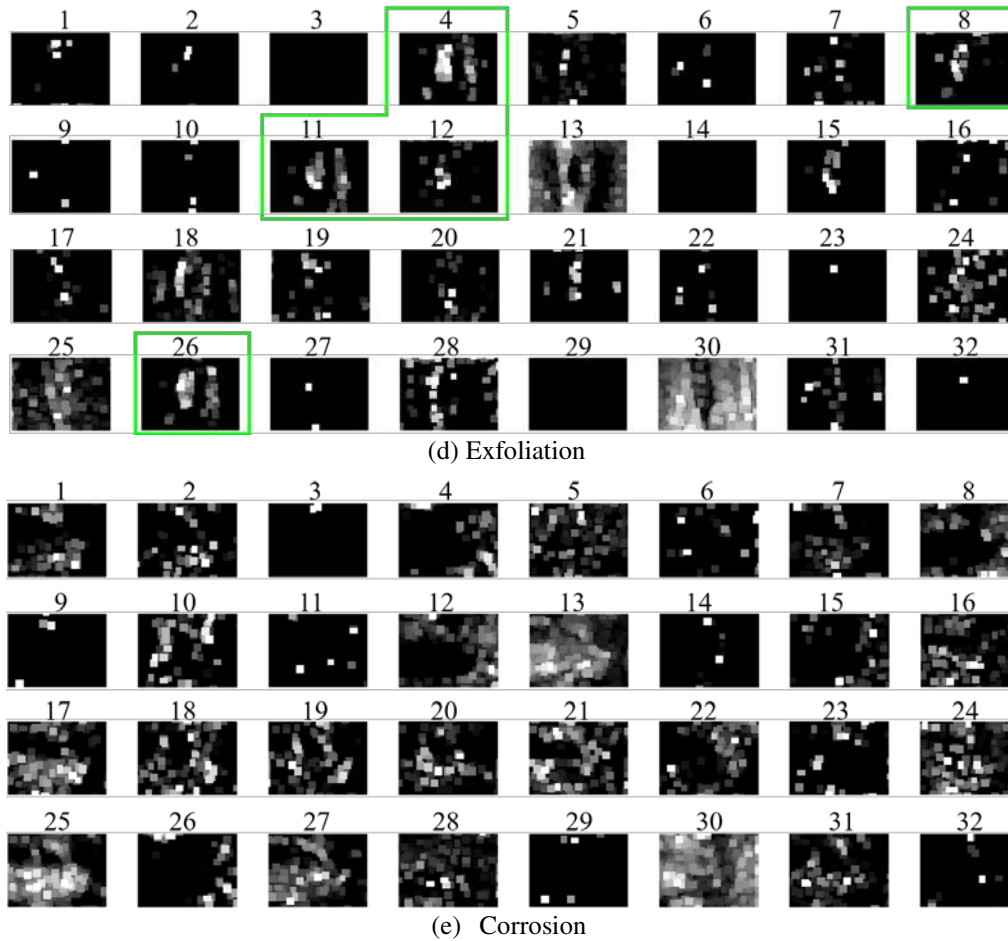(a)   Crack



(b)   Failure



(c)   Free lime

(d) Exfoliation



(e)　Corrosion
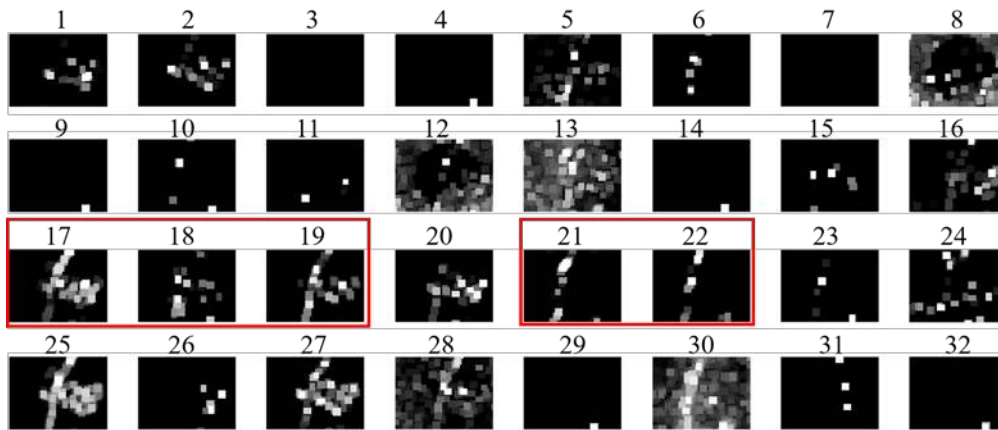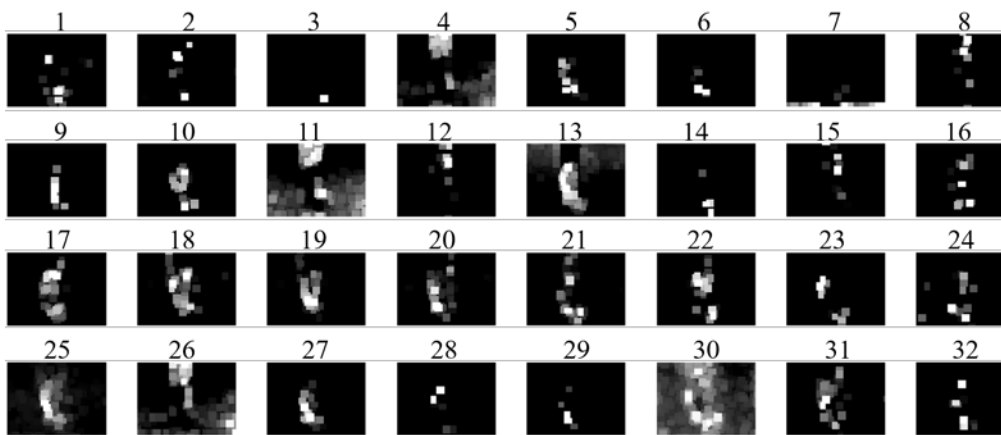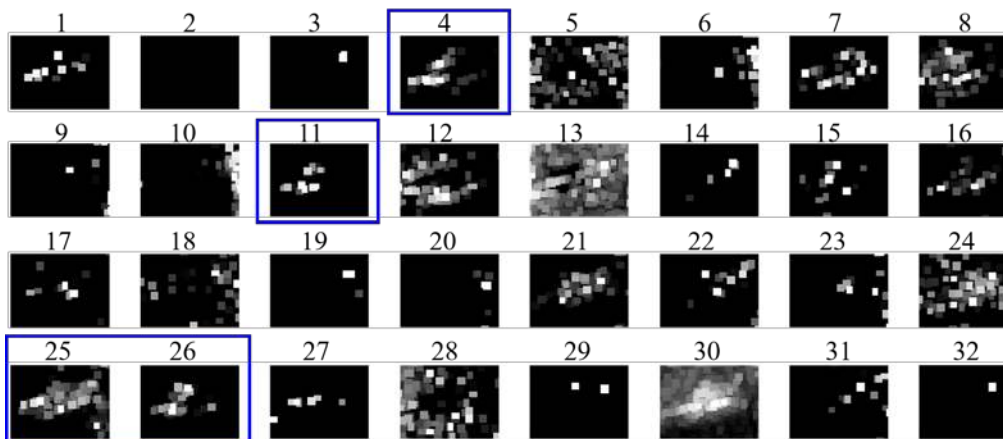
**Figure 6** Feature maps calculated via the second LRN layer. The images in (a), (b), (c), (d) and (e) correspond to those in Fig. 1. Effective feature maps are surrounded by color rectangles. The red rectangles in (a), blue rectangles in (c), and green rectangles in (d) correspond to crack, free lime and exfoliation, respectively.
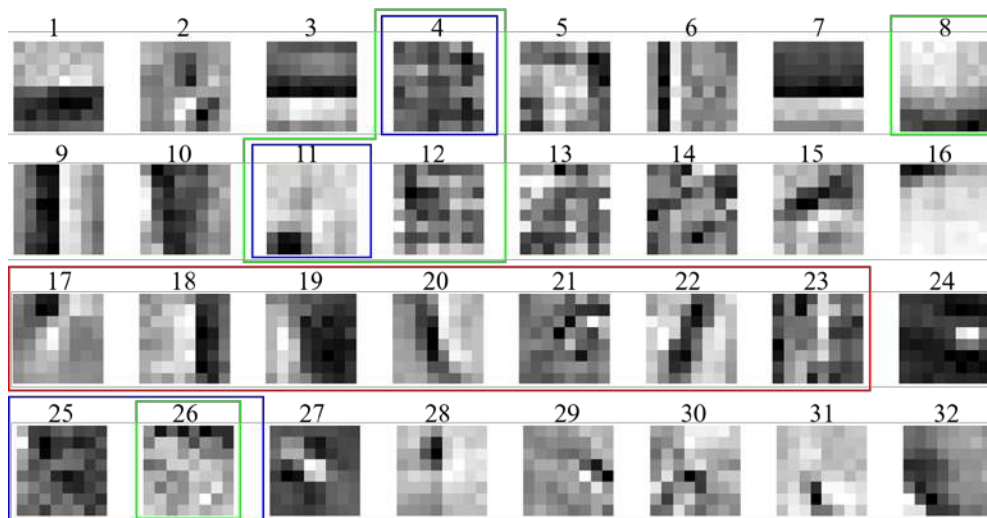


**Figure 7** Dictionaries (red channel) obtained via MCDL. The color rectangles indicate the effective dictionaries for each distress. Red, blue and green correspond to crack, free lime and exfoliation, respectively.

(a) Crack image misclassified as exfoliation



(b) Free lime image misclassified as crack



(c) Exfoliation image misclassified as Free lime

**Figure 8** Examples of misclassified test images. Original images (left column) and their corresponding feature maps calculated via the second LRN layer (right column).



(a) Crack      (b) Corrosion      (c) Exfoliation
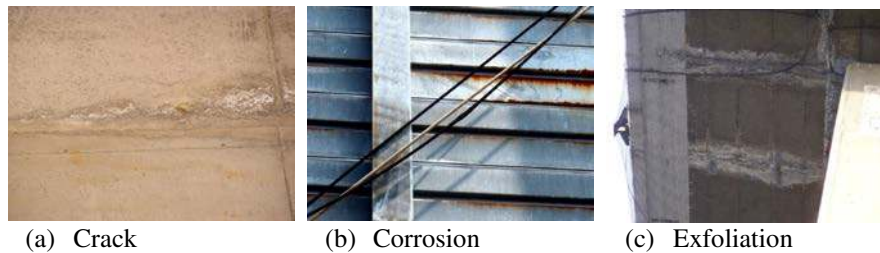
**Figure 9** The proposed method correctly classified all images, and comp. 18 incorrectly classified all images. In addition, (c) was incorrectly classified by comp. 1.

the CSC-based approaches provide interpretation of classification results. In addition, the proposed method is superior to these methods in terms of classification performance. Consequently, the experimental results show that the CSDRN enables effective training from a small number of training images with diverse visual characteristics.

**Table 10** Statistical verification of the proposed method and comp. 1.

|  | *PM* | *Comp. 1* |
|---|---|---|
| Recall | $92.2 \pm 0.47\%$ | $92.0 \pm 0.51\%$ |
| Precision | $92.1 \pm 0.47\%$ | $91.9 \pm 0.52\%$ |
| F-measure | $92.1 \pm 0.47\%$ | $91.9 \pm 0.52\%$ |

CNNs can extract effective features automatically by merely determining the input and output and they realize highly accurate classification. They are very powerful without feature design based on experts' knowledge and experience. In fact, Cha et al. proposed that a CNN is more effective than a traditional crack detection method (Cha et al., 2017) and also realizes location detection for multiple distresses (Cha et al., 2018) by Faster R-CNN (Girshick, 2015). These methods had a great impact on computing in the field of civil engineering. On the other hand, a CNN realizes highly accurate classification, but the poor interpretability of a CNN has been a major problem in recent years. Various approaches by CSC have been proposed as a solution (Papyan et al., 2016). Since filters and feature maps calculated by CSC are created by clear objective functions, interpretation of the results is possible. Figures 6 and 7 show that the activated maps are different for each distress. This is one of the strengths of the proposed method. In future works, it will be necessary to expand the proposed method to a method that can detect the location while maintaining high interpretability.

We should consider various factors in future works to improve the proposed method. From experimental results, we should deal with reduction of computational costs by improving CSC. The detection techniques enable complete automation. Since accurate classification is realized by the proposed method, it is necessary to consider the detection scheme. If the number of images used in the proposed method is small, representation capability of the visual features is not sufficient. On the other hand, if the number of images is large, the calculation cost is increased. There is a need to verify the optimal number of images. For evaluating the robustness of the proposed method, we should verify the effectiveness of our method by using various kinds of distress including leakage; that is, we should apply the proposed method to other datasets. The proposed method does not consider a distress-free situation; in other words, detection of distresses is not realized in the proposed method. We should consider the detection problem in future works.

**ACKNOWLEDGMENTS**

**APPENDIX**

**Details of DRN**

Calculation of DRN is conducted via two steps.

**(I)**    $k = 2, 3, \ldots, K$

DRN calculates the output matrix $V^k = [v_1^k, v_2^k, \ldots, v_N^k]$ of the $k$-th hidden layer as $v_n^k = sig\left(\boldsymbol{\beta}^k v_n^{k-1}\right)$. Note that $sig$ means a sigmoid function. Furthermore, $\boldsymbol{\beta}^k \in \mathbb{R}^{L^k \times L^{k-1}}$ is a weight matrix between the $k$-th and $(k-1)$-th hidden layers and is calculated via the RVFL-based auto-encoder consisting of three layers. The number of input and output nodes of the auto-encoder is $L^{k-1}$, and that of hidden nodes is $L^k$. When an input vector $v_n^{k-1} \in \mathbb{R}^{L^{k-1}}$ is given, the outputs $r_n^k \in \mathbb{R}^{L^k}$ of the auto-encoder's hidden layers can be obtained as $r_n^k = sig(W^k v_n^{k-1} + b^k)$. Note that $(b^k)^\top b^k = 1$ and $W^k(W^k)^\top = I$. In the auto-encoder, $W^k = [w_1^k, w_2^k, \ldots, w_{L^k}^k]^\top$ is an orthogonal random weight, and $b^k = [b_1^k, b_2^k, \ldots, b_{L^k}^k]^\top$ is a random bias. Finally, the output weight $\boldsymbol{\beta}^k$ can be obtained as

$$\boldsymbol{\beta}^k = \left(\frac{I}{C_1}\sum_{l^k=1}^{L^k} \text{KL}\left(\eta || \hat{\eta}_{l^k}\right) + (R^k)^\top R^k\right)^{-1} (R^k)^\top (V^{k-1})^\top. \quad (10)$$

Note that $V^{k-1}$ and $R^k = [r_1^k, r_2^k, \ldots, r_N^k]^\top \in \mathbb{R}^{N \times L^k}$ are an input and a hidden layer's output matrices of the auto-encoder, respectively. Furthermore, $C_1$ is a regularization parameter, $\text{KL}\left(\eta || \hat{\eta}_{l^k}\right) = \eta \log(\eta/\hat{\eta}_{l^k}) + (1-\eta)\log((1-\eta)/(1-\hat{\eta}_{l^k}))$ is the KL divergence, $\eta$ is a sparsity parameter, and $\hat{\eta}_{l^k}$ is the average activation of a node $l^k$ of the auto-encoder's hidden layer.

**(II)**    $k = K + 1$

The output weight $\boldsymbol{\beta}^{K+1} \in \mathbb{R}^{\Phi \times L^k}$ is calculated by RVFL. We minimize the output weights and the training error $\xi_n = [\xi_{n,1}, \xi_{n,2}, \ldots, \xi_{n,\Phi}]^\top$ ($\Phi$ being the number of classes) as

$$\min_{\boldsymbol{\beta}^{K+1}} \frac{1}{2}\left\|\boldsymbol{\beta}^{K+1}\right\|_F^2 + \frac{C_2}{2}\sum_{n=1}^N \|\boldsymbol{\xi}_n\|^2$$

$$\text{s.t.} \quad \boldsymbol{\beta}^{K+1} v_n^K = \boldsymbol{t}_n - \boldsymbol{\xi}_n, \quad (11)$$

where $C_2$ is a regularization parameter. Furthermore, $\boldsymbol{t}_n = [t_{n,1}, t_{n,2}, \ldots, t_{n,\Phi}]^\top$ is a binary vector ($t_{n,\phi} \in \{1,0\}$). If the true class is $\phi$, the $\phi th$ element of the vector is 1, while the other elements become 0. By defining $\boldsymbol{T} = [\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_N]$, the optimal weight of $\boldsymbol{\beta}^{K+1}$ can be calculated as

$$\beta^{K+1} = T(V^K)^\top \left(\frac{I}{C_2} + V^K(V^K)^\top\right)^{-1}. \qquad (12)$$

In both cases (I) and (II), we have to solve the least square error problems.

## REFERENCES

Amezquita-Sanchez, J. & Adeli, H. (2015a), Feature Extraction and Classification Techniques for Health Monitoring of Structures, *Scientia Iranica. Transaction A, Civil Engineering*, **22**(6), 1931.

Amezquita-Sanchez, J. P. & Adeli, H. (2015b), Synchrosqueezed Wavelet Transform-fractality Model for Locating, Detecting, and Quantifying Damage in Smart Highrise Building Structures, *Smart Materials and Structures*, **24**(6), 065034.

Bergquist, B. & Söderholm, P. (2015), Data Analysis for Condition-based Railway Infrastructure Maintenance. *Quality and Reliability Engineering International*, **31**(5), 773−781.

Bristow, H., Eriksson, A. & Lucey, S. (2013), Fast Convolutional Sparse Coding, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 391−398.

Cambria, E., Huang, G. B., Kasun, L. L. C., Zhou, H., Vong, C. M., Lin, J., Yin, J., Cai, Z., Liu, Q., et al. (2013), Extreme Learning Machines [trends & controversies], *IEEE Intelligent Systems*, **28**(6), 30−59.

Cecotti, H. (2016), Deep Random Vector Functional Link Network for Handwritten Character Recognition, In *Proc. International Joint Conference on Neural Networks*, pp. 3628−3633.

Cha, Y.-J., Choi, W. & Büyüköztürk, O. (2017), Deep Learning-based Crack Damage Detection Using Convolutional Neural Networks. *Computer-Aided Civil and Infrastructure Engineering*, **32**(5), 361−378.

Cha, Y.-J., Choi, W., Suh, G., Mahmoudkhani, S., & Büyüköztürk, O. (2018), Autonomous Structural Visual Inspection Using Region-based Deep Learning for Detecting Multiple Damage Types. *Computer-Aided Civil and Infrastructure Engineering*, **33**(9), 731−747.

Chen, B., Li, J., Ma, B. & Wei, G. (2016), Convolutional Sparse Coding Classification Model for Image Classification, In *Proc. IEEE International Conference on Image Processing*, pp. 1918−1922.

Chen, F.-C. & Jahanshahi, M. R. (2018), Nb-cnn: Deep Learning-based Crack Detection Using Convolutional Neural Network and Naive Bayes Data Fusion, *IEEE Transactions on Industrial Electronics*, **65**(5), 4392−4400.

Chollet, F. (2017), Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv preprint arXiv:1610.02357*.

Cortes, C. & Vapnik, V. (1995), Support-vector Networks, *Machine learning*, **20**(3), 273−297.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E. & Darrell, T. (2014), Decaf: A Deep Convolutional Activation Feature for Generic Visual Recognition, In *Proc. IEEE International Conference on Machine Learning*, vol. 32, pp. 647−655.

Doulamis, A., Doulamis, N., Protopapadakis, E., & Voulodimos, A. (2018), Combined Convolutional Neural Networks and Fuzzy Spectral Clustering for Real Time Crack Detection in Tunnels, In *2018 25th IEEE International Conference on Image Processing*, pp. 4153−4157.

Feng, C., Liu, M. Y., Kao, C. C., & Lee, T. Y. (2017), Deep active learning for civil infrastructure defect detection and classification, *Mitsubishi Electric Research Laboratories*.

Gabay, D. & Mercier, B. (1975), A Dual Algorithm for The Solution of Non Linear Variational Problems via Finite Element Approximation, *Institut de recherche dʹinformatique et dʹautomatique*.

Gao, Y. & Mosalam, K. M. (2018), Deep Transfer Learning for Image-based Structural Damage Recognition, *Computer-Aided Civil and Infrastructure Engineering*, **33**(9), 748−768.

Gibert, X., Patel, V. M. & Chellappa, R. (2017), Deep Multitask Learning for Railway Track Inspection, *IEEE Transactions on Intelligent Transportation Systems*, **18**(1), 153−164.

Girshick, R. (2015), Fast R-cnn. In *Proc. IEEE international conference on computer vision*, pp. 1440−1448.

Gopalakrishnan, K., Gholami, H., Vidyadharan, A., Choudhary, A. & Agrawal, A. (2018), Crack Damage Detection in Unmanned Aerial Vehicle Images of Civil Infrastructure Using Pre-trained Deep Learning Model, *International Journal for Traffic and Transport Engineering*, **8**, 1-14.

Gopalakrishnan, K., Khaitan, S. K., Choudhary, A. & Agrawal, A. (2017), Deep Convolutional Neural Networks with Transfer Learning for Computer Vision-based Data-Driven Pavement Distress Detection, *Construction and Building Materials*, **157**, 322−330.

He, K., Zhang, X., Ren, S. & Sun, J. (2016), Deep Residual Learning for Image Recognition, In *Proc. IEEE conference on computer vision and pattern recognition*, pp. 770−778.

Heide, F., Heidrich, W. & Wetzstein, G. (2015), Fast and Flexible Convolutional Sparse Coding, In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5135−5143.

Hotelling, H. (1936), Relations between two sets of variates, *Biometrika*, **28(3/4),** 321−377.

Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K.

Q. (2017), Densely Connected Convolutional Networks. In *CVPR*, pp. 4700-4708.

Huang, G. B., Bai, Z., Kasun, L. L. C. & Vong, C. M. (2015), Local Receptive Field-based Extreme Learning Machine, *IEEE Computational Intelligence Magazine*, **10**(2), 18−29.

Huangpeng, Q., Zhang, H., Zeng, X. & Huang, W. (2018), Automatic Visual Defect Detection Using Texture Prior and Low-rank Representation, *IEEE Access*, **6**, 37965−37976.

Koch, C., Georgieva, K., Kasireddy, V., Akinci, B. & Fieguth, P. (2015), A Review on Computer Vision based Defect Detection and Condition Assessment of Concrete and Asphalt Civil Infrastructure, *Advanced Engineering Informatics*, **29**(2), 196−210.

Koziarski, M. & Cyganek, B. (2017), Image Recognition with Deep Neural Networks in Presence of Noise−dealing with and Taking Advantage of Distortions, *Integrated Computer-Aided Engineering*, **24**(4), 337−349.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012), Imagenet Classification with Deep Convolutional Neural Networks, In *Proc. Advances in neural information processing systems*, pp. 1097−1105.

Li, R., Yuan, Y., Zhang, W. & Yuan, Y. (2018), Unified Vision-based Methodology for Simultaneous Concrete Defect Detection and Geolocalization, *Computer-Aided Civil and Infrastructure Engineering*, **33**, 527−544.

Liang, X. (2018), Image-based Post-disaster Inspection of Reinforced Concrete Bridge Systems Using Deep Learning with Bayesian Optimization, *Computer-Aided Civil and Infrastructure Engineering*.

Liao, T.-Y. (2017), On-line Vehicle Routing Problems for Carbon Emissions Reduction, *Computer-Aided Civil and Infrastructure Engineering*, **32**(12), 1047−1063.

Lin, Y.-z., Nie, Z.-h., & Ma, H.-w. (2017), Structural Damage Detection with Automatic Feature-extraction through Deep Learning, *Computer-Aided Civil and Infrastructure Engineering*, **32**(12), 1025−1046.

Long, J., Shelhamer, E., & Darrell, T. (2015), Fully Convolutional Networks for Semantic Segmentation, In *Proc. IEEE conference on computer vision and pattern recognition*, pp. 3431−3440.

Luo, W., Li, J., Yang, J., Xu, W. & Zhang, J. (2017), Convolutional Sparse Autoencoders for Image Classification, *IEEE transactions on neural networks and learning systems*, (99):1−6.

Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., & Omata, H. (2018a), Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images, *Computer-Aided Civil and Infrastructure Engineering*, **0**, 1−15.

Maeda, K., Takahashi, S., Ogawa, T. & Haseyama, M.

(2017a), Automatic Estimation of Deterioration Level on Transmission Towers via Deep Extreme Learning Machine based on Local Receptive Field, In *Proc. IEEE International Conference on Image Processing*, pp. 2379−2383.

Maeda, K., Takahashi, S., Ogawa, T. & Haseyama, M. (2017b), Distress Classification of Road Structures via Adaptive Bayesian Network Model Selection. *Journal of Computing in Civil Engineering*, **31**(5), 04017044.

Maeda, K., Takahashi, S., Ogawa, T. & Haseyama, M. (2018b), Distress Classification of Class-imbalanced Inspection Data via Correlation-maximizing Weighted Extreme Learning Machine. *Advanced Engineering Informatics*, **37**, 79−87.

Nabian, M. A. & Meidani, H. (2018), Deep Learning for Accelerated Seismic Reliability Analysis of Transportation Networks. *Computer-Aided Civil and Infrastructure Engineering*, **33**(6), 443−458.

Ortega-Zamorano, F., Jerez, J. M., Gómez, I. & Franco, L. (2017), Layer Multiplexing Fpga Implementation for Deep Back-propagation Learning. *Integrated Computer-Aided Engineering*, **24**(2), 171−185.

O 'Byrne, M., Schoefs, F., Ghosh, B. & Pakrashi, V. (2013), Texture Analysis based Damage Detection of Ageing Infrastructural Elements. *Computer-Aided Civil and Infrastructure Engineering*, **28**(3), 162−177.

Pao, Y.-H., Park, G.-H. & Sobajic, D. J. (1994), Learning and Generalization Characteristics of the Random Vector Functional-link Net. *Neurocomputing*, **6**(2), 163−180.

Papyan, V., Romano, Y. & Elad, M. (2016), Convolutional Neural Networks Analyzed via Convolutional Sparse Coding, *arXiv preprint arXiv:1607.08194*.

Poskus, E., Rodgers, G. W., Zhou, C. & Chase, J. G. (2018), Damage Identification for Hysteretic Structures Using A Mode Decomposition Method, *Computer-Aided Civil and Infrastructure Engineering*, **33**(2), 97−109.

Qarib, H. & Adeli, H. (2015), A New Adaptive Algorithm for Automated Feature Extraction in Exponentially Damped Signals for Health Monitoring of Smart Structures, *Smart Materials and Structures*, **24**(12), 125040.

Rafiei, M. H. & Adeli, H. (2017), A Novel Machine Learning-based Algorithm to Detect Damage in High-rise Building Structures, *The Structural Design of Tall and Special Buildings*, **26**(18), e1400.

Rafiei, M. H. & Adeli, H. (2018), A Novel Unsupervised Deep Learning Model for Global and Local Health Condition Assessment of Structures, *Engineering Structures*, **156**, 598−607.

Rafiei, M. H., Khushefati, W. H., Demirboga, R. & Adeli, H. (2017), Supervised Deep Restricted Boltzmann Machine for Estimation of Concrete, *ACI Materials Journal*, **114**(2).

Sarma, K. C. & Adeli, H. (1995), Effect of General Sparse

Matrix Algorithm on Optimization of Space Structures, *AIAA journal*, **33**(12), 2442−2444.

Sarma, K. C. & Adeli, H. (1996), Sparse Matrix Algorithm for Minimum Weight Design of Large Dtructures, *Engineering Optimization+ A35*, **27**(1), 65−85.

Schmidt, W. F., Kraaijveld, M. A. & Duin, R. P. (1992), Feedforward Neural Networks with Random Weights, In *Proceedings of IAPR International Conference on Pattern Recognition*, pp. 1−4.

Shen, Y., Chen, J. & Xiao, L. (2017), Supervised Classification of Hyperspectral Images Using Local Receptive Fields-based Kernel Extreme Learning Machine, In *Proc. IEEE International Conference on Image Processing*, pp. 3120−3124.

Simonyan, K. & Zisserman, A. (2014), Very Deep Convolutional Networks for Large-scale Image Recognition, *arXiv preprint arXiv:1409.1556*.

Šorel, M. & Šroubek, F. (2016), Fast Convolutional Sparse Coding Using Matrix Inversion Lemma, *Digital Signal Processing*, **55,** 44−51.

Szegedy, C., Ioffe, S., Vanhoucke, V. & Alemi, A. A. (2017), Inception-v4, Inception-resnet and The Impact of Residual Connections on Learning, In *AAAI*, vol. 4, pp. 4278-4284.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. (2016), Rethinking The Inception Architecture for Computer Vision, In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 2818−2826.

Tizani, W. & Mawdesley, M. J. (2011), Advances and Challenges in Computing in Civil and Building Engineering, *Advanced Engineering Informatics*, **25**(4), 569−572.

Wang, J., Yang, J., Yu, K., Lv, F., Huang, T. & Gong, Y. (2010), Locality-constrained Linear Coding for Image Classification, In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 3360−3367.

Wang, W., Zhang, A., Wang, K. C., Braham, A. F. & Qiu, S. (2018), Pavement Crack Width Measurement based on Laplace′s Equation for Continuity and Unambiguity, *Computer-Aided Civil and Infrastructure Engineering*, **33**(2), 110−123.

Welch, B. L. (1938), The Significance of The Difference between Two Means When The Population Variances Are Unequal, *Biometrika*, **29**(3/4), 350−362.

Wohlberg, B. (2016a), Convolutional Sparse Representation of Color Images, In *IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, pp. 57−60.

Wohlberg, B. (2016b), Efficient Algorithms for Convolutional Sparse Representations, *IEEE Trans. on Image Processing*, **25**(1), 301−315.

Wold, S., Esbensen, K. & Geladi, P. (1987), Principal Component Analysis, *Chemometrics and intelligent laboratory systems*, **2**(1-3), 37−52.

Woo, S., Chu, I., Youn, B. & Kim, K. (2016), Development of The Corrosion Deterioration Inspection Tool for Transmission Tower Members, *KEPCO Journal on Electric Power and Energy*, **2**(2), 293−298.

Xia, W. (2018), An Approach for Extracting Road Pavement Disease from HD Camera Videos by Deep Convolutional Networks, In *2018 International Conference on Audio, Language and Image Processing (ICALIP)*, pp. 418−422.

Xu, P., Sun, Q., Liu, R., Souleyrette, R. R. & Wang, F. (2015), Optimizing The Alignment of Inspection Data from Track Geometry Cars, *Computer-Aided Civil and Infrastructure Engineering*, **30**(1), 19−35.

Xue, Y. & Li, Y. (2018), A Fast Detection Method via Region-based Fully Convolutional Neural Networks for Shield Tunnel Lining Defects, *Computer-Aided Civil and Infrastructure Engineering*, **33**(8), 638−654.

Yang, J., Yu, K., Gong, Y. & Huang, T. (2009), Linear Spatial Pyramid Matching Using Sparse Coding for Image Classification, In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1794−1801.

Yu, Y. and Sun, Z. (2017), Sparse Coding Extreme Learning Machine for Classification, *Neurocomputing*, **261**, 50−56.

Zalama, E., Gómez, G. B. J., Medina, R. & Llamas, J. (2014a), Road Crack Detection Using Visual Features Extracted by Gabor Filters, *Computer-Aided Civil and Infrastructure Engineering*, **29**(5), 342−358.

Zhang, A., Wang, K. C., Li, B., Yang, E., Dai, X., Peng, Y., Fei, Y., Liu, Y., Li, J. Q. & Chen, C. (2017a), Automated Pixel-level Pavement Crack Detection on 3d Asphalt Surfaces Using A Deep-learning Network, *Computer-Aided Civil and Infrastructure Engineering*, **32**(10), 805−819.

Zhang, L., Yang, F., Zhang, Y. D. & Zhu, Y. J. (2016), Road Crack Detection Using Deep Convolutional Neural Network, In *Proc. IEEE International Conference on Image Processing*, pp. 3708−3712.

Zhang, S., Wang, J., Tao, X., Gong, Y. & Zheng, N. (2017b), Constructing Deep Sparse Coding Network for Image Classification, *Pattern Recognition*, **64**, 130−140.

Zhang, Z., Xu, Y., Yang, J., Li, X. & Zhang, D. (2015), A Survey of Sparse Representation: Algorithms and Applications. *IEEE access*, **3**, 490−530.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A. & Oliva, A. (2014), Learning Deep Features for Scene Recognition Using Places Database, In *Proc. Advances in neural information*

*processing systems*, pp. 487–495.