Authors' Accepted Manuscript

Journal: Engineering Structures

Volume 260, 1 June 2022, 114227

https://doi.org/10.1016/j.engstruct.2022.114227

Structural symmetry recognition in planar structures using Convolutional Neural Networks

Pei Zhang [a,1], Weiying Fan [b,1], Yao Chen [b,*], Jian Feng [b], Pooya Sareh [c]

[a] College of Civil and Transportation Engineering, Hohai University, Nanjing 210098, China
[b] Key Laboratory of Concrete and Prestressed Concrete Structures of Ministry of Education, and National Prestress Engineering Research Center, Southeast University, Nanjing 211189, China
[c] Creative Design Engineering Lab (Cdel), Department of Mechanical, Materials, and Aerospace Engineering, School of Engineering, University of Liverpool, Liverpool, L69 3GH, UK

ARTICLE INFO

Keywords:
Deep learning
Planar structure
Pictures
Symmetry classification
Symmetry order

ABSTRACT

In both natural and man-made structures, symmetry provides a range of desirable properties such as uniform distributions of internal forces, concise transmission paths of forces, as well as rhythm and beauty. Most research on symmetry focus on natural objects to promote the developments in computer vision. However, countless engineering structures also contain symmetry elements since ancient times. In fact, many scholars have investigated symmetry in engineering structures, but most of them are based on analytical methods which require tedious calculations. Inspired by the application of deep learning in image identification, in this paper, we use two Convolutional Neural Networks (CNNs) to respectively identify the symmetry group and symmetry order of planar engineering structures. To this end, two different datasets with labels for symmetric structures are created. Then, the datasets are used to train and test the constructed network models. For symmetry classification, it achieves 86.69% accuracy, which takes about 0.006 s to predict one picture. On the other hand, for symmetry order recognition, it reaches 92% accuracy, which expends about 0.005 s to identify an image. This method provides an efficient approach to the exploration of structural symmetry, which can be expanded and developed further toward the identification of symmetry in three-dimensional structures.

1. Introduction

Symmetry refers to the regular repetition of certain elements in a figure, object, or structure. The exploration of symmetry and exploitation of its rules have been of attraction to many scholars and practitioners in various fields such as computer vision [1–3], structural mechanics and engineering [4–6][43,46] and mathematics [7–11] [45,47]. Countless natural and man-made structures [12] contain symmetry properties, which provide them with plenty of advantages in terms of both structural mechanics and aesthetics [13]. Relatively uniform distributions of internal forces, concise transmission paths of forces [14], as well as rhythm and beauty, are some generally desirable characteristics of symmetric structures.

Over the past few decades, many studies have been conducted on symmetry detection and recognition, especially for two-dimensional (2D) symmetric objects. Cicconet *et al.* [1] established a framework based on complex-valued wavelet convolutions to discern the lines of 2D reflective symmetry. They observed that the accuracy of this process was affected by three parameters α, β, and d of symmetric wavelet orbit pairs. To compensate for this defect, a mirror symmetry coefficient was introduced. Funk and Liu [15] built a supervised deep neural network to discover reflective axes and rotational centers in images which contained multiple out-of-plane symmetries. The artificial symmetry label was replaced by the symmetry heatmap as the output. Kondra *et al.* [16] improved their method, studied the correlation measurements of the symmetry detection algorithm, and demonstrated its applicability to a range of texture patterns. Dalitz *et al.* [17] employed a symmetry score to evaluate the symmetry properties of grayscale images. In order to recognize the reflective symmetry plane of three-dimensional (3D) objects, Nagar and Raman [3] translated this matter into an optimization problem, while Gao *et al.* [18] proposed an unsupervised 3D Convolutional Neural Network (CNN) to address the issue of poor time-

* Corresponding author.
E-mail address: chenyao@seu.edu.cn (Y. Chen).
[1] The authors Zhang and Fan contribute equally to this paper.

efficiency. Furthermore, Sun and Sherrah [2] exploited Extended Gaussian Image to explore the symmetries of images from a diverse range of formats. This approach not only identified reflection symmetries, but also determined the principal axes and orders of rotational symmetries. In addition, Aguilar and Bribiesca [19] introduced four theorems based on the orthogonal direction change chain code to detect the reflective and rotational symmetries of 3D curves and trees. It was able to acquire the number of symmetry axes and detect local symmetries. Since the process of symmetry detection can be influenced by noise, Chang *et al.* [20] proposed an Artificial Neural Network (ANN) for black and white pictures to weaken this impact; however, only simple pictures were trained and tested in that network.

Considerable progress has been recently made in the area of symmetry detection and utilization of civil engineering structures . Zingoni [23] first proposed an automatic procedure for the systematic search and identification of the symmetric structures. Inspired by the pioneering work of Zingoni [21], Chen *et al.* [22,23] proposed a group-theoretical method to identify cyclie and cubic symmetries for both 2D and 3D structures. Chen et al. [24] demonstrated that the mobility and geometric stability of the skeleton structures are enhanced by symmetry. Origami structures with rotational symmetry were used to cushion the collision of lightweight robotic rotorcraft [25]. Isomorphic and non-isomorphic symmetric descendants were designed based on the Miura origami pattern [9,10,26,27], where certain derivatives were proved to be geometrically impossible [44]. A combination of symmetry analysis and particle swarm optimization was used for the form-finding of tensegrity structures [14]. Vibration mode calculations for layered space grids were simplified based on their symmetries [28]. The key quest of these group-theoretic methods was to determine whether the structure has certain symmetry properties, followed by determining the highest-order symmetry group to which it belong. During this process, many iterative calculations were required to determine whether a structure remained invariant under each symmetry operation. Although many approaches to symmetry detection have been developed by various scholars, most of them are based on analytical methods applied to image features used to locate the reflective symmetry axes or the rotational symmetry centers. However, being time-consuming is an inevitable disadvantage of such analytical methods, especially when applied to structures with a lot of nodes and components.

Given these challenges, this study is devoted to identifying symmetry properties where a deep learning method will be adapted to improve the computational efficiency of analytical methods. So far, the popularity of deep learning, especially Convolutional Neural Network (CNN), in image recognition provides a new clue for symmetry detection. A trained CNN model is able to accurately identify the category of pictures in less than 0.1 s. For example, De Luca *et al.* [29] proposed a machine learning algorithm to classify artificial pictures into rotational, translational, and reflective symmetries with 99% accuracy. In addition, the maturity of the third-party library makes the construction and application of the CNN model convenient, so that people with basic machine learning knowledge could master it.

In this paper, two CNN models are proposed to detect the symmetry group and order of planar structures. Considering that there is no existing dataset to use, two datasets will be created autonomously. Subsequently, the datasets are employed to train and test the constructed network model. Three different optimizers are used to control the training process. The performances of the CNN models driven by different optimizers were compared and analyzed.

2. Theory and evaluation criteria

2.1. Symmetry operations and cyclic symmetries

This section introduces the characteristics of various symmetry operations and the properties of cyclic symmetries. For a symmetric structure, performing several independent linear transformations (i.e., symmetry operations) results in a transformed structure that perfectly overlaps the original one. This study mainly focuses on the recognition of the symmetry group and symmetry order of finite planar structures. For convenience, a three-dimensional (3D) Cartesian coordinate system *x-y-z* is established, supposing that two-dimensional (2D) structures are planar figures on the *x-y* plane. Then symmetry operations of a finite planar structure can be divided into the following three categories [12,30]:

(a) The *identity*,*E*, which is found in all structures and can be expressed as follows

$$E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{1}$$

(b) The *reflection*,σ_v, which denotes reflection about the vertical plane containing the principle axis (z axis), expressed as

$$\sigma_v = \begin{bmatrix} cos(2\alpha_r) & sin(2\alpha_r) \\ sin(2\alpha_r) & -cos(2\alpha_r) \end{bmatrix} \tag{2}$$

where α_r is the angle between the symmetry plane and the *x-z* plane.

(c) The *rotation*, C_n^i, which is a rotational transformation by angle $2\pi i/n$ around the axis of symmetry, where $n \geqslant 2$ and $i \in [1, n-1]$ are both integers. It can be mathematically represented as [31]

$$C_n^i = \begin{bmatrix} cos(2\pi i/n) & -sin(2\pi i/n) \\ sin(2\pi i/n) & cos(2\pi i/n) \end{bmatrix} \tag{3}$$

Group theory has been used to describe the symmetric properties of engineering structures [32–34]. The symmetry group of a structure is the group of all transformations under which the structure remains unchanged [35,36]. In this study, we focus on planar structures from four cyclic groups: (1) asymmetric structures of group C_1, (2) reflectively symmetric structures of group C_s, (3) rotationally symmetric structures of group C_n, and (4) rotationally-and-reflectively symmetric structures of group C_{nv}. Knowing that the *order*, n_r, of a symmetry group is determined by the number of its symmetry operations, the symmetry orders of groups C_1, C_s, C_n, and C_{nv} are 1, 2, n, and $2n$, respectively. If a structure belongs to any one of the two symmetry groups C_1 and C_s, its symmetry order would be known. For structures of groups C_n and C_{nv}, the symmetry order depends on n, which is to be determined in this research.

2.2. Evaluation principle of classification

Classification principles attempt to associate input variables with discrete categories. According to this definition, the study presented here can be a classification task. In order to evaluate the proposed deep learning model, three general classification criteria, namely, (1) accuracy *Acc*, (2) precision *P*, and (3) recall *R*, were selected. These variables can be expressed as

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$P = \frac{TP}{TP + FP} \tag{5}$$

$$R = \frac{TP}{TP + FN} \tag{6}$$

where *TP*, *TN*, *FP*, and *FN* denote the number of positive classes predicted as positive classes, the number of negative classes predicted as negative classes, the number of negative classes predicted as positive classes, and the number of positive classes predicted as negative classes, respectively.

3. Methodology

A classic CNN model consists of an input layer, hidden layers

(including convolutional, pooling, and fully-connected layers), and an output layer. With the help of convolutional and pooling layers, a CNN can capture the features of images after multiple iterations. Because the information associated with a typical symmetric structure could be presented in the form of pictures, here we employ the CNN model to replace conventional theoretical approaches to structural symmetry recognition. It aims to detect the symmetry group of two-dimensional structures and identify the n_r of C_n and C_{nv} ($n \in [2, 10]$), where two datasets, namely, Dataset 1 and Dataset 2, are required. The details of this procedure are presented in Fig. 1.

3.1. Building datasets

The objects discussed before were mostly continuous rather than discrete. However, the elements of structures in civil engineering are generally discrete due to functional requirements and economic goals. There is no existing database that can be used to optimize the parameters of CNN in this work. Therefore, many structural forms are independently generated to serve the symmetry recognition of the deep learning models.

Supervised deep learning models are adapted here, so that the data sets need to be classified and labeled in advance. In the process of generating pictures, we put pictures belonging to the same category in the same folder. Dataset 1 is prepared to perform the classification of symmetry group (i.e., groups C_1, C_s, C_n, and C_{nv}). The number of pictures in each category is the same, with a total of 6800 pictures. Then, Dataset 2 is employed to identify the order of symmetry, which is a fourteen-classification task ($n_r \in [2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 18, 20]$). Dataset 2 contains a total of 18,000 pictures. When a category contains both C_n and C_{nv}, the number of pictures in this category is twice that of others. In short, there are 1000 pictures for which of $n_r \in [2,3,5,7,9,12,14,16,18,20]$, while there are 2000 structures for other symmetry orders. For the division of a dataset, 60% of the samples are selected as the training set, 20% as the validation set, and the rest as the test set. Since there are no available datasets for symmetric structures, it brings a great challenge for the application of this method. To solve this problem, many pictures are created through Python. The general principles followed by all the pictures are as follows. The picture is a black figure on a white background with a pixel size of 1000 × 1000, bit depth of 8, and format of '.png'. Each picture consists of solid dots and lines, in which lines represent structural components, and dots denote the connection mode between components. On the one hand, each dot is connected to at least 2 adjacent dots. On the other hand, the periphery of the structure is closed. After preprocessing, the picture pixels are changed to 200 × 200 and then transmitted to the CNN model. The following presents the details of the generation process for different groups of pictures.

For asymmetric structures (i.e., of group C_1), as depicted in Fig. 2, the dots are evenly distributed on a ring, but the connections among them are arbitrary and have no symmetry properties. To improve the robustness of the CNN model, a structure that is highly similar, but not identical, to the reflectively symmetric one (i.e., of group C_s), is also created. For example, in such highly similar structures, except for one line, all other lines and dots satisfy the requirements of reflective symmetry.

Fig. 3 illustrates some examples of the reflectively symmetric structures of group C_s. The generation of dots and lines for this kind of structures follow Eq. (2). The angles α_r between the symmetry axis and the horizontal line are 0, π/6, π/4, π/3, π/2, 2π/3, and 3π/4. There is a total of 1,700 pictures for reflective symmetry.

The positions of the dots and lines for structures of group C_n are determined by Eq. . To ensure the universality of the order of rotational symmetry, a rotationally symmetric structure with the order from 2 to 10 is generated. The configurations of some structures of group C_n are represented in Fig. 4.

Fig. 5 shows the structures with rotational and reflective symmetry properties. The initial dots and lines are transformed using Eq. (7) to generate new dots and lines.

$$C_{nv}^{i} = \begin{bmatrix} cos(2\alpha_r + 2\pi i/n) & sin(2\alpha_r + 2\pi i/n) \\ sin(2\alpha_r + 2\pi i/n) & -cos(2\alpha_r + 2\pi i/n) \end{bmatrix} \tag{7}$$

where α_r, i, and n represent the same parameters as in Eqs. (1) to (3).

3.2. The CNN model adopted for the experiment

The Convolutional Neural Network (CNN) [37–41], a type of deep learning algorithm, has been popular in a wide range of fields including computer vision and natural language processing. Two individual CNN models are prepared here separately for Dataset 1 and Dataset 2. Fig. 6 shows the CNN-I model adopted by Dataset 1. It is composed of one input layer, three 2D convolution layers (activation function = *ReLU*),

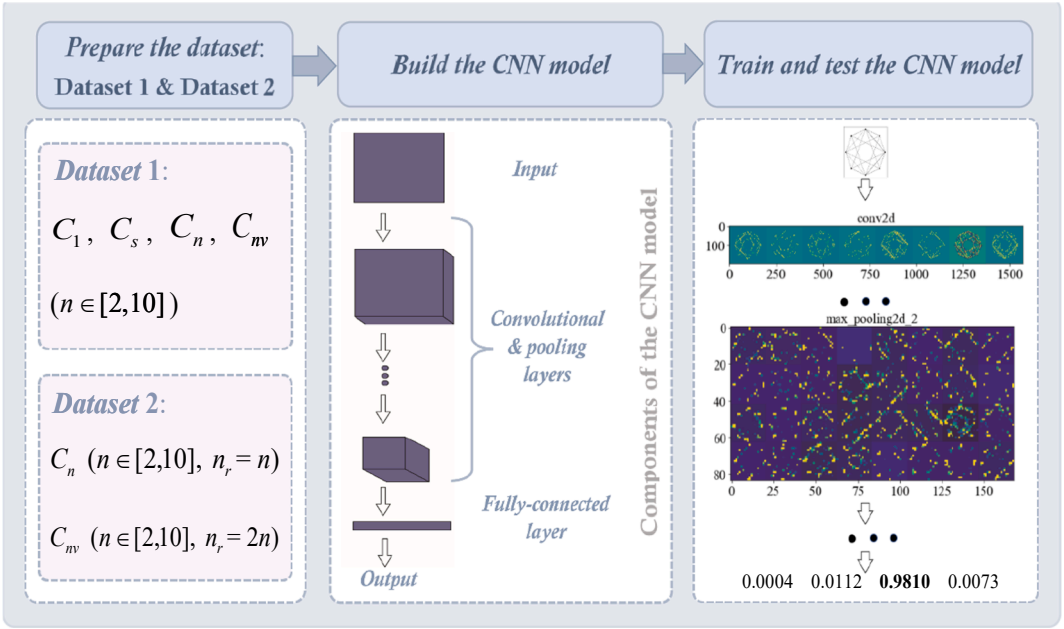

Fig. 1. Procedure of planar symmetry detection using Convolutional Neural Network (CNN).

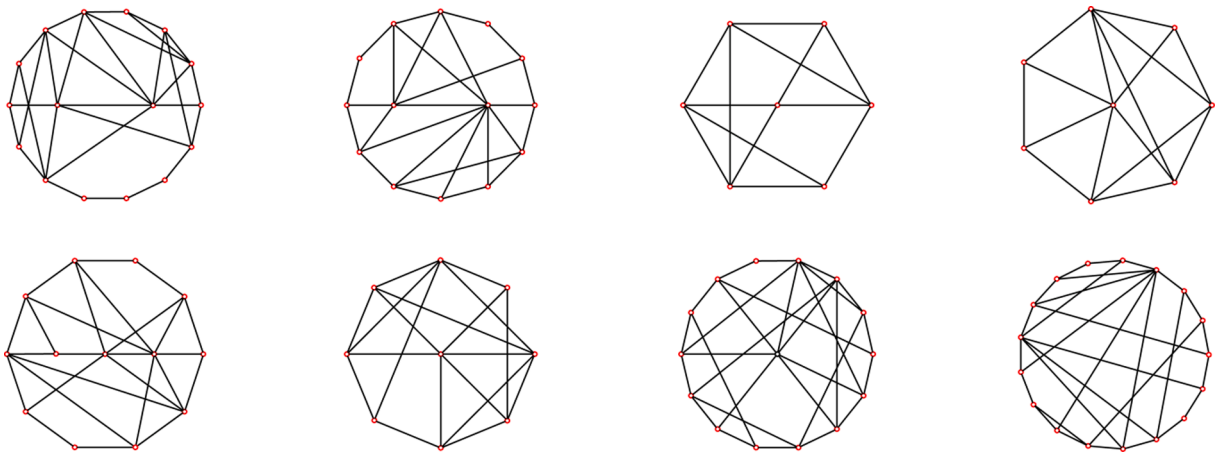

Fig. 2. Asymmetric planar structures (C_1).

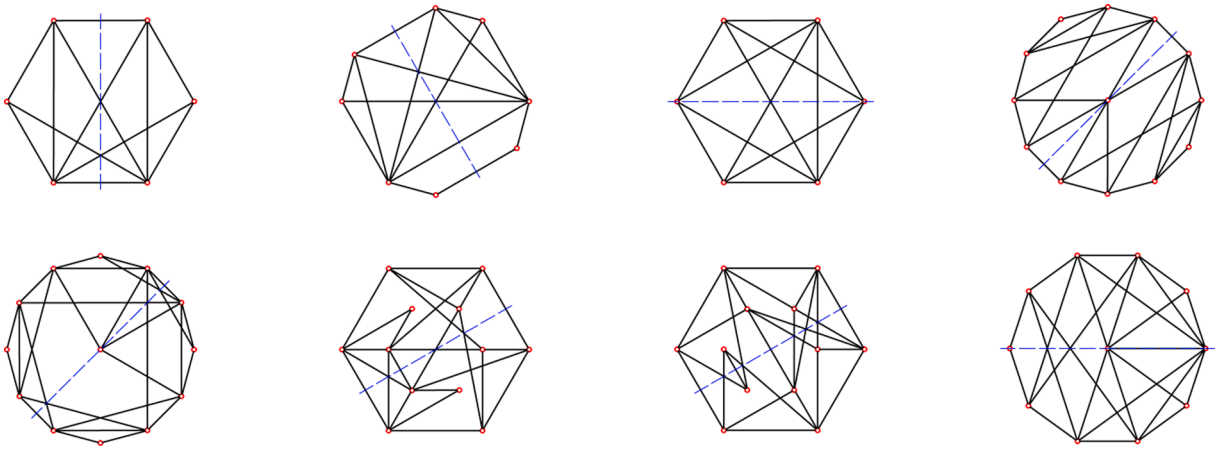

Fig. 3. Planar structures with reflective symmetry (C_s).

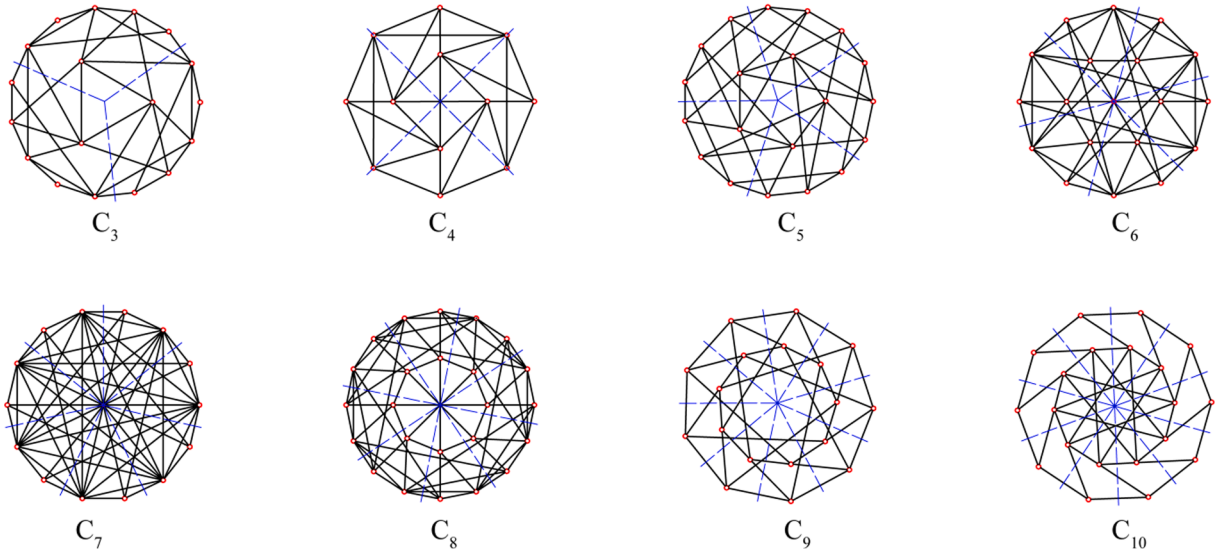

Fig. 4. Planar structures with rotational symmetry (C_n).

three 2D max-pooling layers, three fully-connected layers, and one output layer (activation function = *Softmax*). To prevent overfitting, the two fully-connected layers framed by the dashed line in Fig. 6 adopt dropout (dropout = 0.5). The difference between the two CNN models is that the output layer of the network model for Dataset 2 owns 14 neurons.

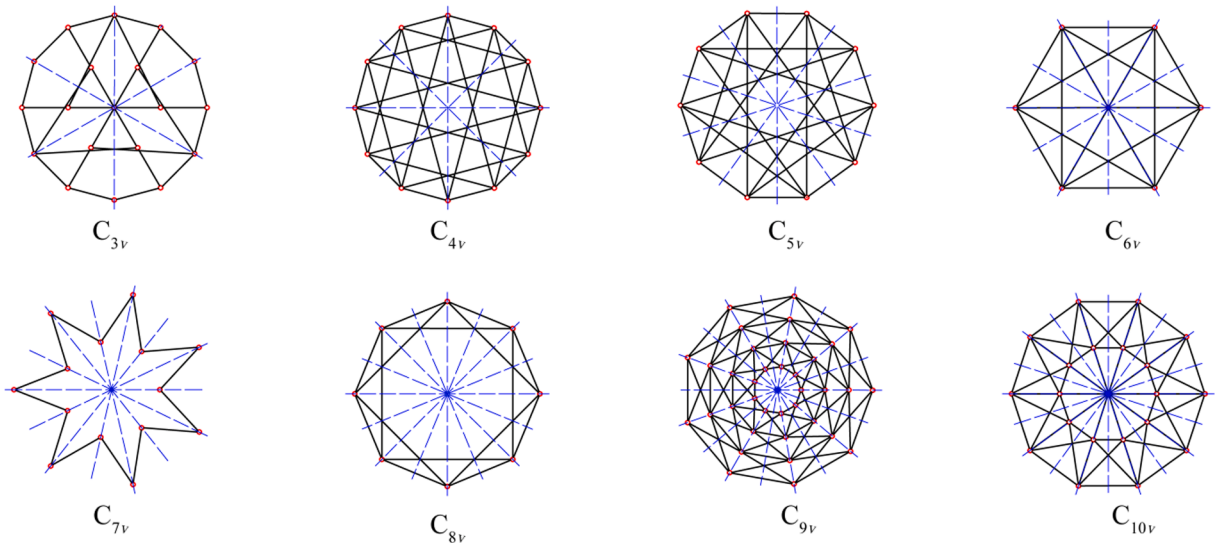

Fig. 5. Planar structures with rotational and reflective symmetries (C_{nv}).

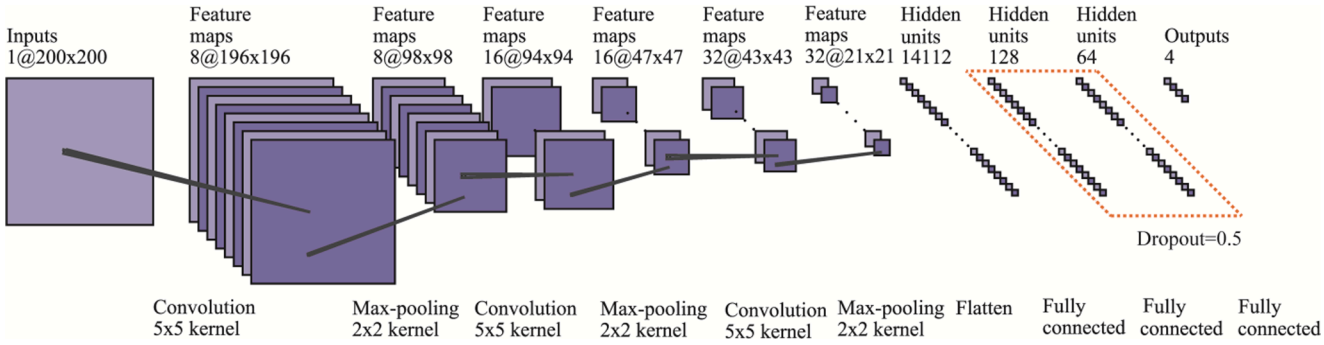

Fig. 6. Convolutional Neural Network-I (CNN-I) model used for Dataset 1.

The *ReLU* and *softmax* activation functions are given by Eqs. (8) and (9) [38], respectively:

$$f(\boldsymbol{x}) = max(0, \boldsymbol{x}) = \begin{cases} 0, & \boldsymbol{x} \leqslant 0 \\ \boldsymbol{x}, & \boldsymbol{x} > 0 \end{cases} \tag{8}$$

$$\sigma(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{9}$$

where $\boldsymbol{x}$ is the input variable, x_i denotes the i-th element of $\boldsymbol{x}$, where i is an integer from 1 to n, and n is the total number of elements of the input variable $\boldsymbol{x}$. Taking Dataset 1 as an example, if the input is a picture, CNN will output four probability values and choose the category with the highest probability for the picture. Since these are multi-classification tasks, the sparse categorical cross-entropy function below is taken as a *loss* function:

$$loss = -\sum_{i=1}^{n} \widehat{y}_{i1} logy_{i1} + \widehat{y}_{i2} logy_{i2} + \cdots + \widehat{y}_{im} logy_{im} \tag{10}$$

where n is the size of the sample; m indicates the number of classifications; and y_{i1} and $\widehat{y}_{i1}$ denote true value and predicted value, respectively.

'Python' builds up a strong platform to realize machine learning or deep learning with the help of various third-party libraries. The CNN models adopted in this work are created using the TensorFlow library supported by Python.

3.3. Training the model

This experiment is carried out on a computer with Intel (R) Core (TM) i5-9400 CPU, 2.90 GHz CPU frequency and 8 GB memory. The training set is used to fit the model. The validation set is prepared for adjusting the hyperparameters of the CNN model when one epoch is completed. For the black-and-white image input into the model, the pixel size is 200 × 200, and it is normalized to avoid the influence of noise points. During training, the accuracy and loss are monitored, and the accuracy of the validation set is taken as the standard to save the network model. The model with a higher validation accuracy will be saved in time. If the accuracy of the validation set is no longer improved after five epochs, the model will stop training.

In this work, the convergence of loss and accuracy means that the model has achieved convergence. In order to ensure that the model converges at the end of the set number of epochs, a larger epoch is selected as much as possible. If the model fails to converge, the last model saved previously is applied, then epoch and record will be added. Considering that each sample is fully utilized, the batch size of Dataset 1 is 34 and that of Dataset 2 is 32. In order to investigate the effect of optimizers on the convergence and performance of the CNN model, three different optimizers (SGD, Adam, RMSprop) are selected herein.

To understand the working principle of CNN, Fig. 7 shows the information collected from the convolutional layer and pooling layer. Fig. 7 illustrates the feature maps in the convolutional layers and pooling layers for a set of pictures. The best models after training are selected here. The visualization of the 2D feature layer for the CNN-I model built by Dataset 1 is shown in Fig. 7(a), while the features

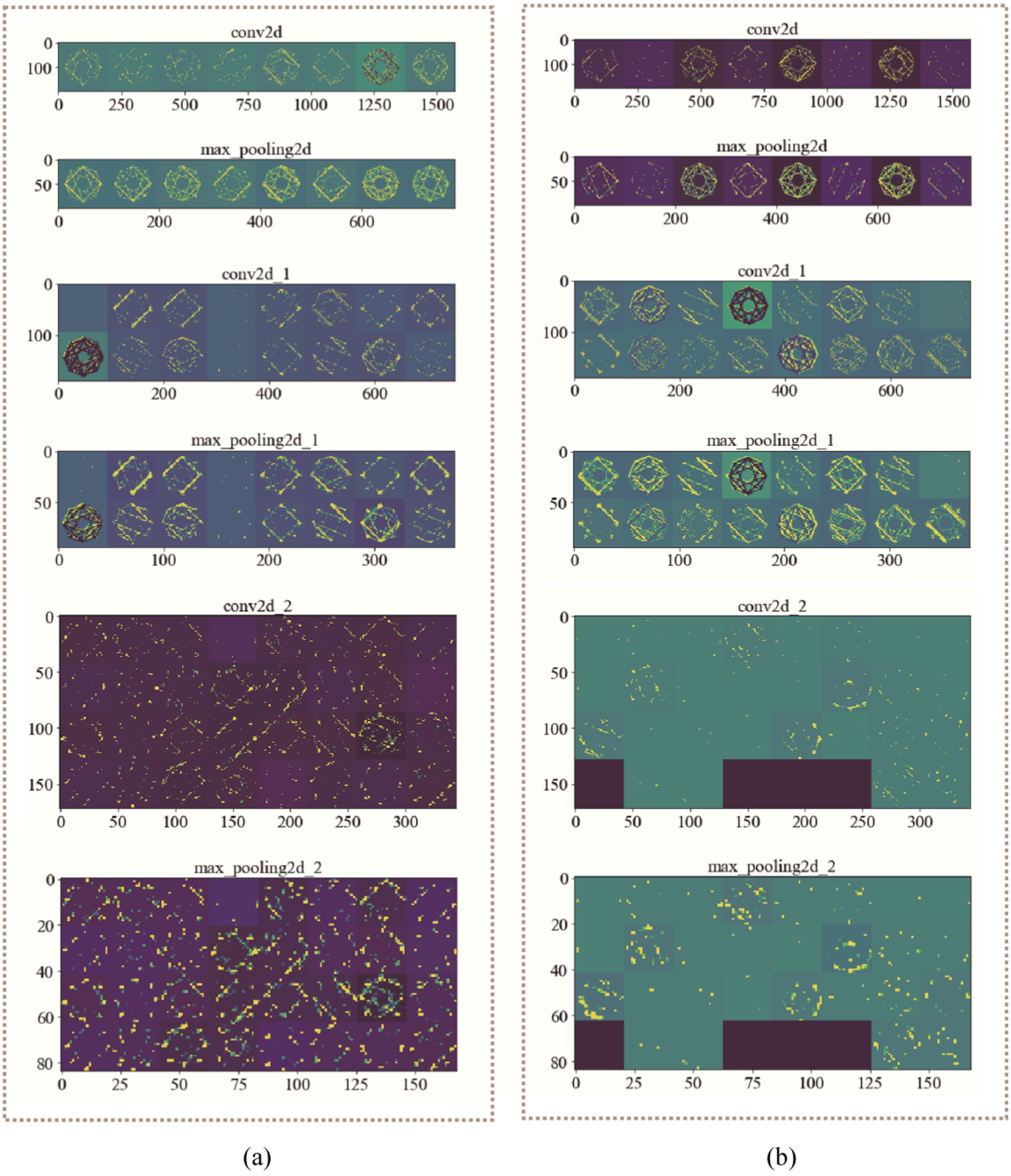

Fig. 7. Visualization of feature maps for Convolutional Neural Network model: (a) Dataset 1; (b) Dataset 2.

obtained by the CNN-II model of Dataset 2 are illustrated in Fig. 7(b). For the two classification tasks, the input pictures are exactly the same, and the structures of the constructed CNN models are almost identical. However, Fig. 7 shows that the extracted features are considerably different. Moreover, the extracted features are gradually reduced and become more abstract with the deepening of the convolution layer. The original content of the image is no longer visible.

4. Results and discussion

4.1. Detecting symmetry groups over Dataset 1

The structures in Dataset 1 are artificially divided into four categories in advance, namely C_1, C_s, C_n, and C_{nv}. The responsibility of the CNN-I model is to update the weight of the network through training until it can accurately distinguish the categories of pictures.

The training results of CNN-I with three different optimizers are shown in Fig. 8, Fig. 9, and Table 1. Fig. 8 shows that the accuracy of the training set is lower than that of the validation set at the beginning, but it can approach or even exceed the latter with the development of the process. When epoch is equal to 20, the accuracy of the validation set is 0.45 for the SGD optimizer, 0.75 for the Adam optimizer, and 0.85 for the RMSprop optimizer, which indicates that the convergence rate of the RMSprop optimizer is faster than that of the other optimizers.

The change of loss during training is given in Fig. 9. Moreover, when the training is completed, the accuracy or loss of the validation set and training set tend to converge. Table 1 lists the key parameters and results of the constructed CNN-I model.

Fig. 10 shows the confusion matrix of the test set. The data in this figure represents the number of samples that satisfy both the predicted

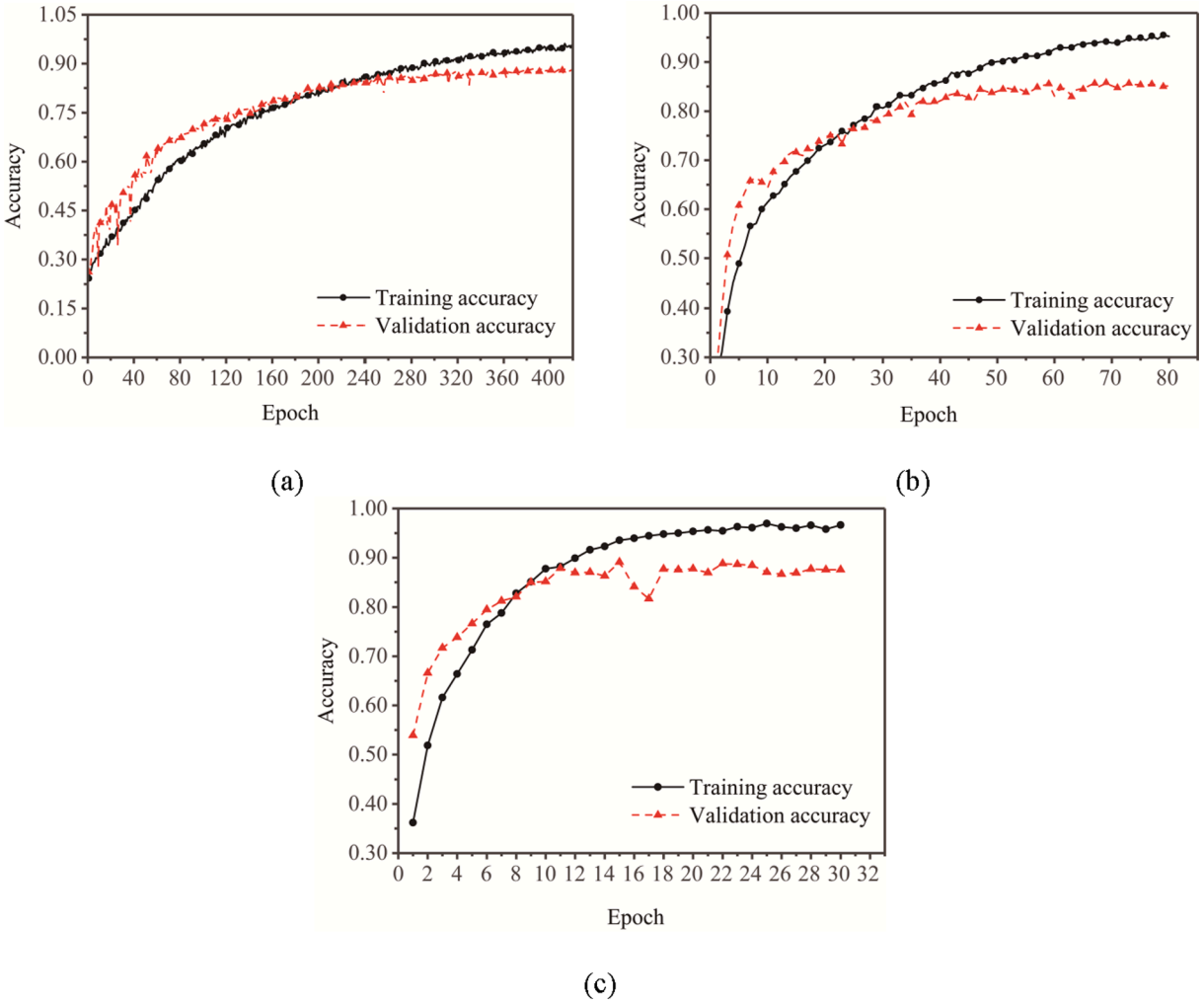

Fig. 8. The accuracy in symmetry group classification on dataset 1: (a) SGD; (b) Adam; (c) RMSprop.

label and the true label. The precision and recall presented in Table 2 can be determined from the corresponding data. No matter which optimizer is selected, the recall and precision of C_n and C_{nv} are relatively low. Besides, there are more samples of mutual confusion between C_n and C_{nv}. Therefore, it is necessary to divide the two categories further. Comparing the relative error between the accuracy of the test set, validation set, and training set of the three optimizers, RMSprop wins with errors of 4.375% and 7.333%.

4.2. *Identifying symmetry order* n_r *over Dataset 2*

Dataset 2 is exploited to identify the symmetry orders of C_n and C_{nv}. It is divided into fourteen categories, so the output layer of the corresponding CNN-II model has 14 neurons. The orders matching the labels of the samples are provided in Table 3.

Table 4 states the performance of the CNN-II model on Dataset 2. It can be seen from Table 4 that the accuracy of the training set and validation set achieves 93%, and the accuracy of the test set exceeds 91%. On the other hand, the generalization ability of the CNN-II driven by SGD optimizer is higher in terms of the accuracy of the test set. Comparing the relative error between the accuracy of the test set, validation set, and training set of the three optimizers for Dataset 2, the CNN-II driven by SGD wins with errors of 0.5214% and 2.085%. More importantly, the approach described here takes about 0.05 s per picture, which saves a lot of computing time and resources.

Fig. 11 shows the changes of classification accuracy on Dataset 2 during the training process. Specifically, the accuracy of the training set as well as the validation set are growing with the increase of the epoch, and finally, tend to converge about 90%. The CNN-II driven by Adam and RMSprop optimizers are becoming steady after 10 epochs. However, the CNN-II driven by SGD optimizers become steady after 40 epochs. So that the convergence rates of the Adam and RMSprop optimizers are better than that of SGD.

The goal of training a CNN is to make the value of the loss function as small as possible. The smaller the loss value, the more accurate the classification of the CNN. Variations in the value of the loss function during training for CNN-II driven by different optimizers are presented in Fig. 12. In general, the value of the loss function decreases rapidly at the beginning, and it float around 0.25 after a certain epoch. Obviously, the loss on the validation set is smaller than the training set in the initial stage. Gradually, the classification effect of the validation set becomes worse, and the loss is greater than the training set. However, the losses on both training and validation sets tend to converge eventually.

The specific results of the CNN-II model on the test set for Dataset 2 are shown in Fig. 13. Dataset 2 has fourteen categories, and it is difficult to display the number of samples. Therefore, the confusion matrix is normalized. The sum of the probabilities of each row in Fig. 13 is equal to 1.00, and the diagonal entry represents the recall rate of each category. Parts (a), (b), and (c) in Fig. 13 are similar. When n_r is equal to 10 (label = 0), the recall rates of different optimizers vary greatly.

Comparing the optimal models selected for Dataset 1 and Dataset 2, it can be proved that the best optimizer varies from problem to problem.

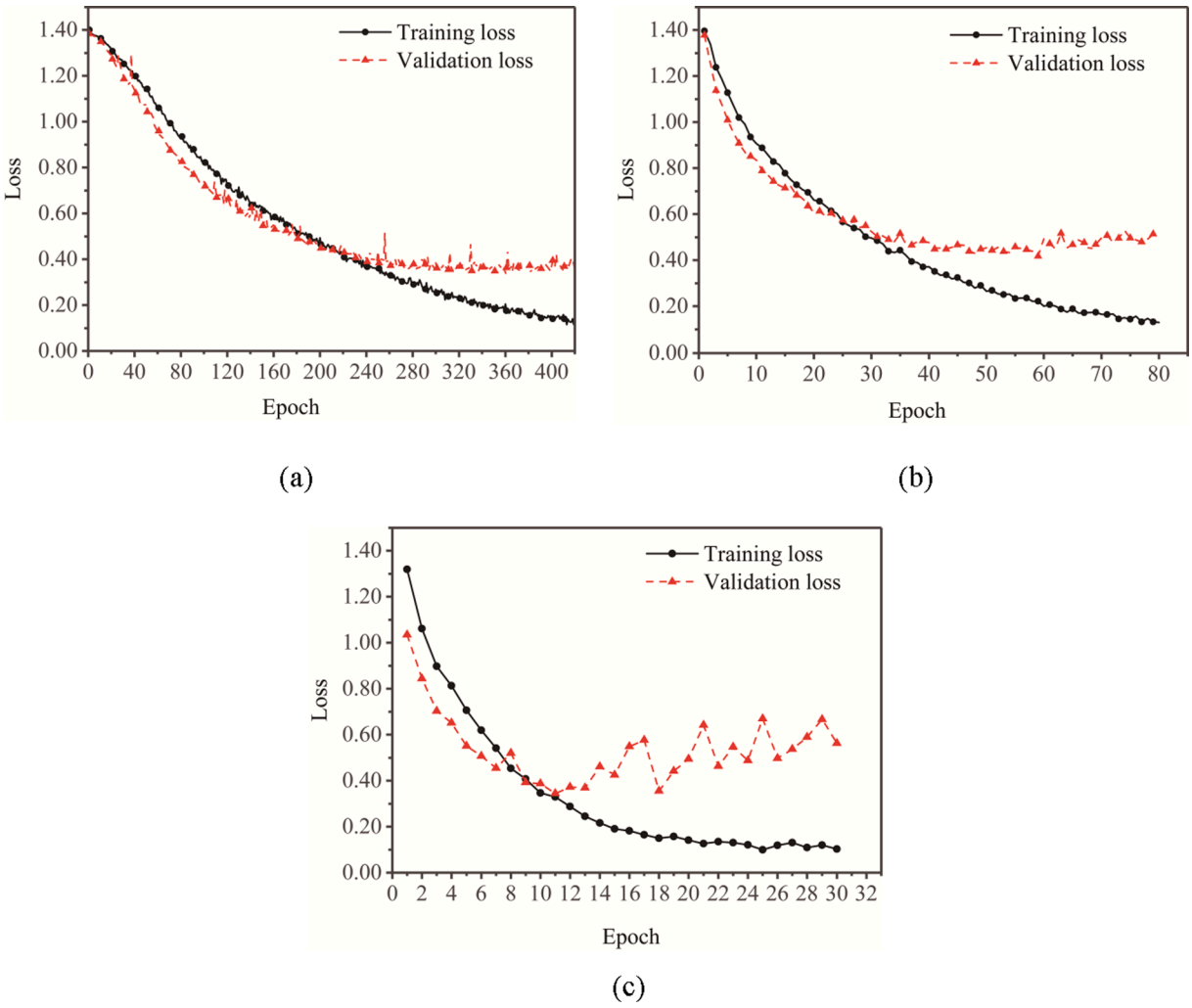

Fig. 9. The loss in symmetry group classification on Dataset 1: (a) SGD; (b) Adam; (c) RMSprop.

Table 1
The performance of the CNN-I model on Dataset 1.

Optimizer	Learning rate	Epoch	*Acc* of training set	*Acc* of validation set	*Acc* of test set	Running time (*h*)	Prediction efficiency (*s*)
SGD	0.001	420	0.9316	0.8831	0.8324	20.0300	0.0055
Adam	0.0001	80	0.9417	0.8529	0.8529	2.6691	0.0068
RMSprop	**0.001**	**30**	**0.9355**	**0.8912**	**0.8669**	**0.8221**	**0.0059**

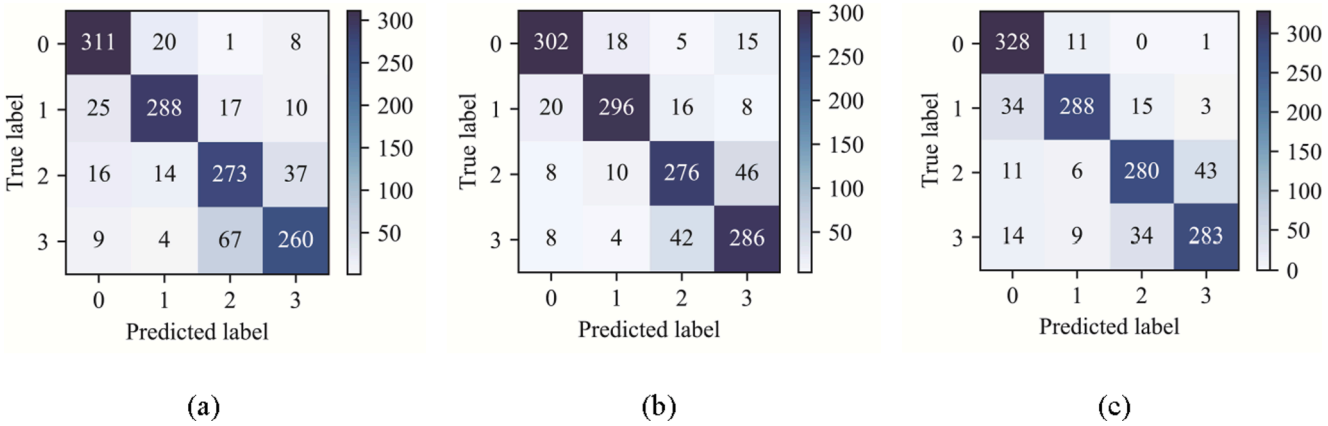

Fig. 10. The confusion matrix of the CNN-I model on Dataset 1: (a) SGD; (b) Adam; (c) RMSprop.

Table 2
The precision and recall of the test set on Dataset 1.

Optimizer	SGD				Adam				RMSprop			
Label	0	1	2	3	0	1	2	3	0	1	2	3
Group	C_1	C_s	C_{nv}	C_n	C_1	C_s	C_{nv}	C_n	C_1	C_s	C_{nv}	C_n
P	0.8615	0.8834	**0.7626**	0.8254	0.8935	0.9024	0.8142	0.8056	0.8475	**0.9172**	0.8511	0.8576
R	0.9147	0.8471	**0.7647**	0.7647	0.8882	0.8706	0.8118	0.8412	**0.9647**	0.8471	0.8235	0.8324

Table 3
The label of the CNN-II model and the matching order on Dataset 2.

Label	0	1	2	3	4	5	6	7	8	9	10	11	12	13
n_r	10	12	14	16	18	20	2	3	4	5	6	7	8	9

Table 4
The performance of the CNN-II model on Dataset 2.

Optimizer	Learning rate	Epoch	*Acc* of training set	*Acc* of validation set	*Acc* of test set	Running time (h)	Prediction efficiency (s)
SGD	0.001	200	**0.9396**	**0.9347**	**0.9200**	48.3195	0.0050
Adam	0.001	45	0.9410	0.9403	0.9169	7.5989	0.0051
PMSprop	0.001	45	0.9339	0.9302	0.9197	8.1878	0.0050

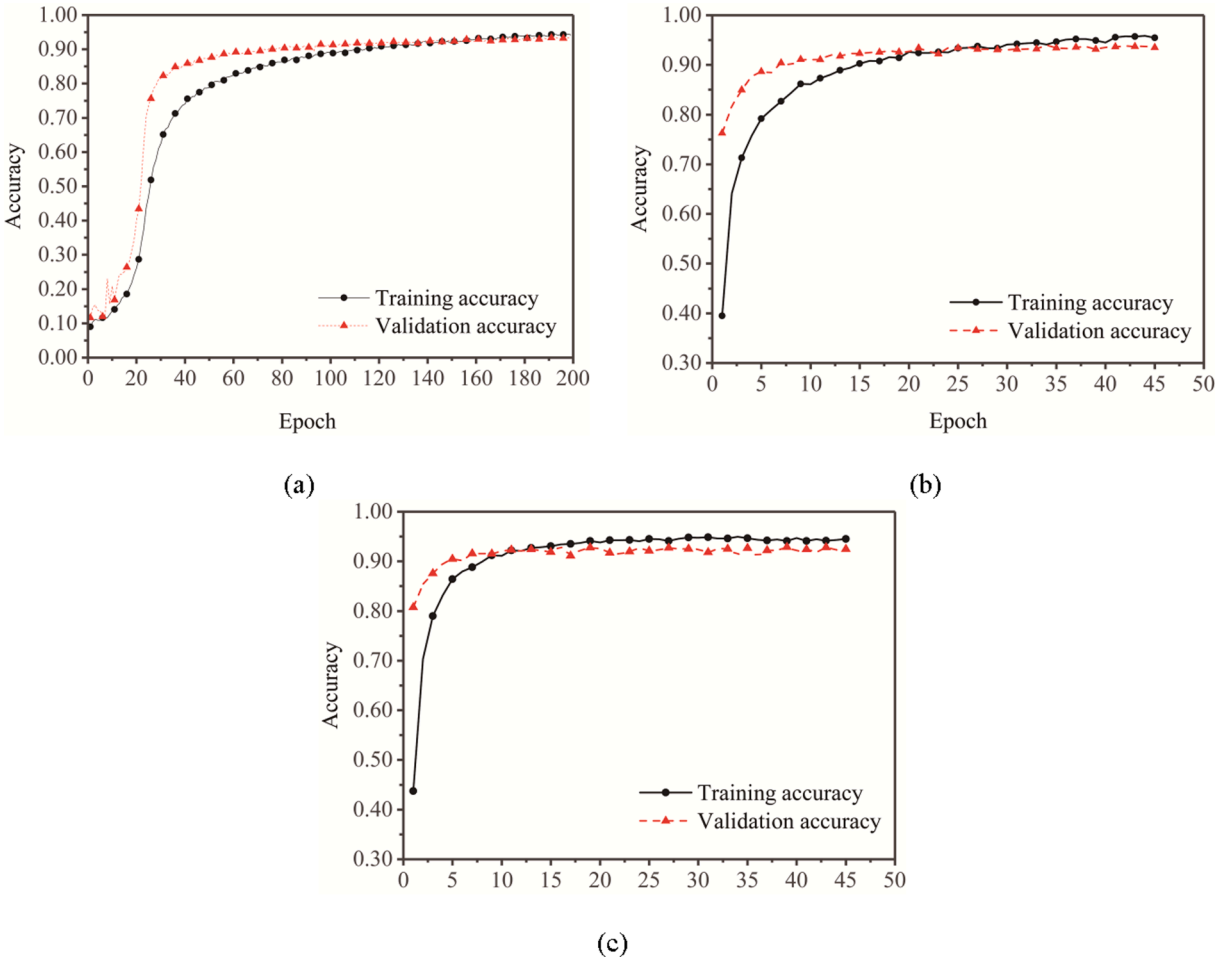

Fig. 11. The accuracy in symmetry order classification on Dataset 2: (a) SGD; (b) Adam; (c) RMSprop.

When the established model does not adapt to a conundrum, hyper-parameters such as the learning rate, the number of neurons, the number of neural layers, and the choice of optimizer can be adjusted properly. Although the update of CNN weights also relies on multiple iterations, this is a one-time task. If the training is completed, the model is in a position like an implicit function. When the required pictures are inputted, the results are outputted efficiently.

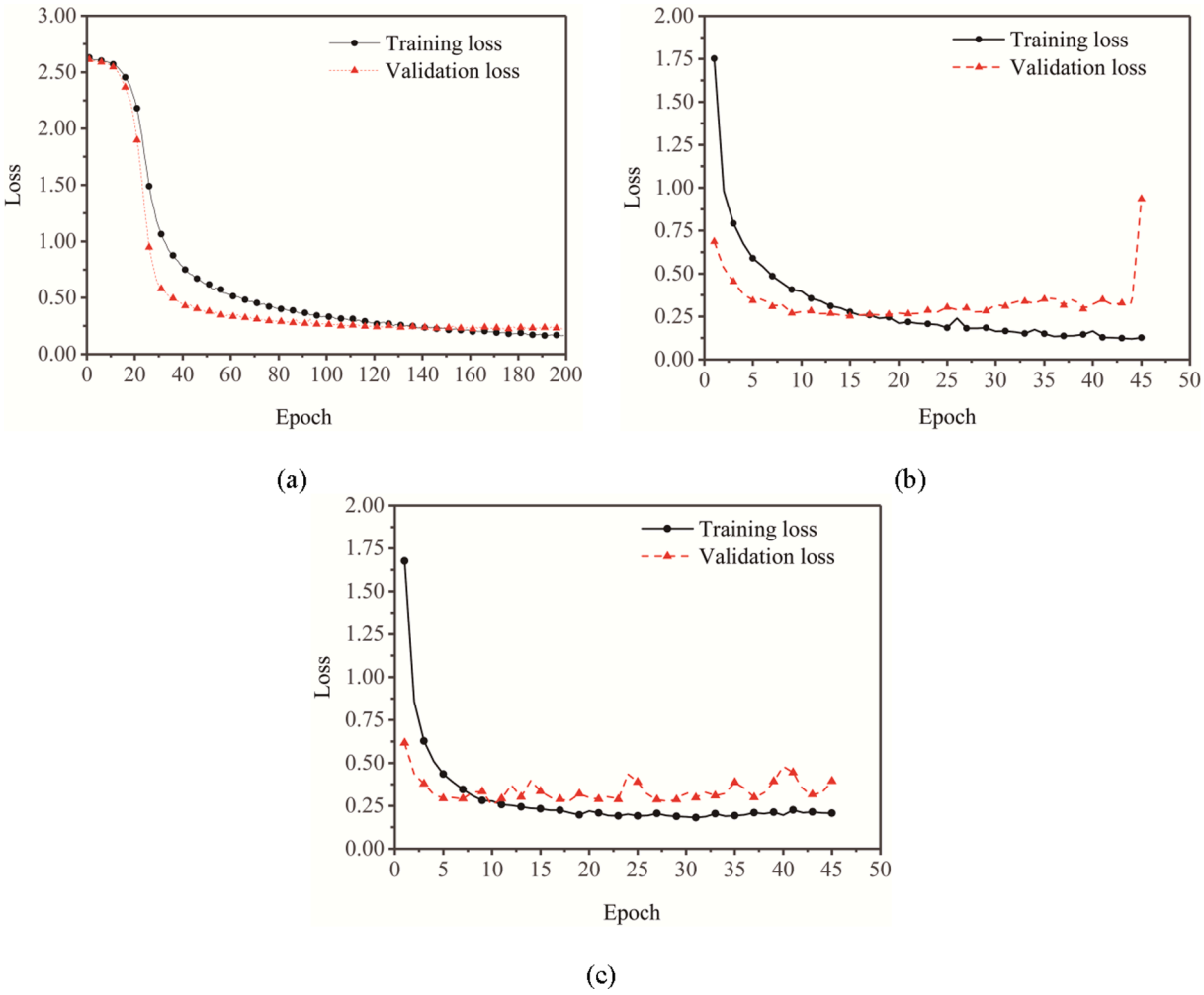

Fig. 12. The loss in symmetry order classification on Dataset 2: (a) SGD; (b) Adam; (c) RMSprop.

4.3. Illustrative examples

This section presents the results of CNNs for symmetry recognition of some engineering structures. We explore the symmetry identification of planar structures. Usually, the symmetry property of some three-dimensional structures can be seen from the plane projection. At this time, the symmetry property can be obtained by identifying the plane projection. Therefore, we choose some 2D and 3D structures to demonstrate the practicality of CNN-I and CNN-II.

Fig. 14 shows a two-layer circulant structure [42]. It can be seen that the two-layer structure has 4 cycles. It is a rotationally symmetric structure with the symmetry order of 4. First, the picture in Fig. 14c is input into CNN-I and CNN-II directly. The output of the CNN-I is 3, indicating that the structure owns rotational symmetry property. The output of the CNN-II is 8. We can notice from Table 3 that the symmetry order of this two-layer structure is 4. It is consistent with the inherent symmetry of this structure. Therefore, this two-layer circulant structure belongs to C_4 group.

Moreover, a three-dimensional dome structure [22] is illustrated in Fig. 15. It is a symmetric structure with rotational and reflective symmetries. We can put the picture in Fig. 15c into CNN-I and CNN-II. The output of the CNN-I is 2, indicating that the structure has rotational and reflective symmetry properties. The output of the CNN-I is 3, indicating that the structure has rotational symmetry properties. The output of the CNN-II is 3, which is consistent with its symmetry order. Therefore, this three-dimensional dome structure belongs to C_{8v} group.

Fig. 16 illustrates two planar structures. The structure in Fig. 16a keeps reflective symmetry, which is computed by CNN-I. The output of the CNN-I is 1, indicating that the structure has reflective symmetry properties. The structure in Fig. 16b shows a similar symmetry. However, when the component indicated by the red dash line is removed, CNN-I for this modified structure is 0. Thus, the structure without the dashed component become asymmetric and belongs to the lowest symmetry group C_1.

In conclusion, CNN-I can accurately identify the symmetry classes of the structures in Figs. 14-16. Furthermore, the structure in Fig. 16b would induce symmetry breaking if one component was removed. However, CNN-I can accurately identify the exact symmetry for both symmetric and asymmetric configurations, showing its robustness.

5. Conclusions and future work

In this study, CNN models were employed to recognize the global symmetry of various planar structures, which revealed the potential of deep learning for symmetry detection purposes. The accuracy of the test set for symmetry group classification reached 86.69%, while for symmetry order recognition reached 92.00%. In comparison with analytical methods, it turned out that by using CNN the classification speed is greatly improved as it takes less than 0.01 s to predict the category of an image.

This simulation verifies that a structure with a certain configuration can be classified accurately after a deep learning classification task. However, there is a diverse range of structural forms in civil engineering. This method can be used to identify symmetry properties of cable-

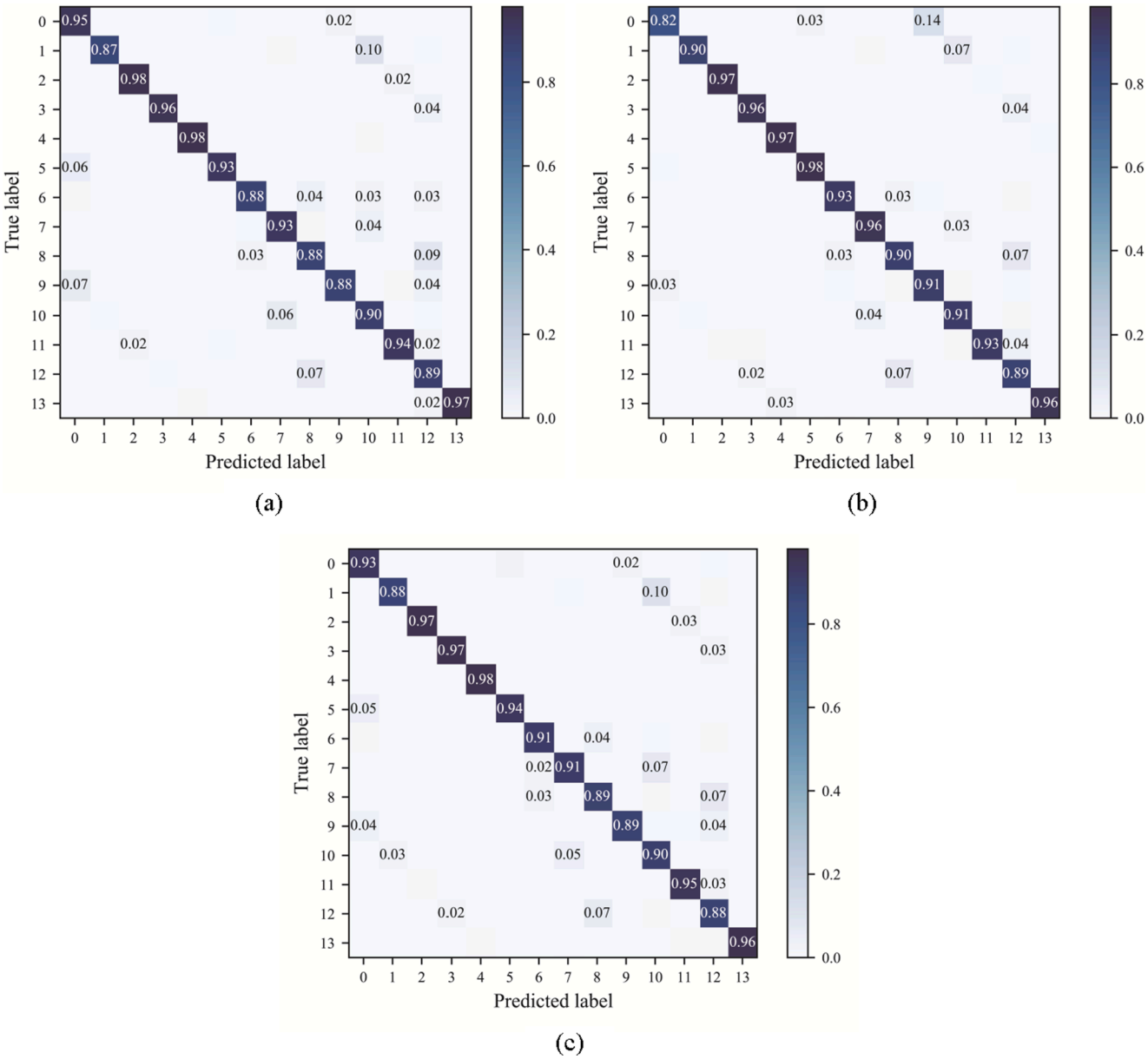

Fig. 13. The confusion matrix of CNN-II model on Dataset 2: (a) SGD; (b) Adam; (c) RMSprop.

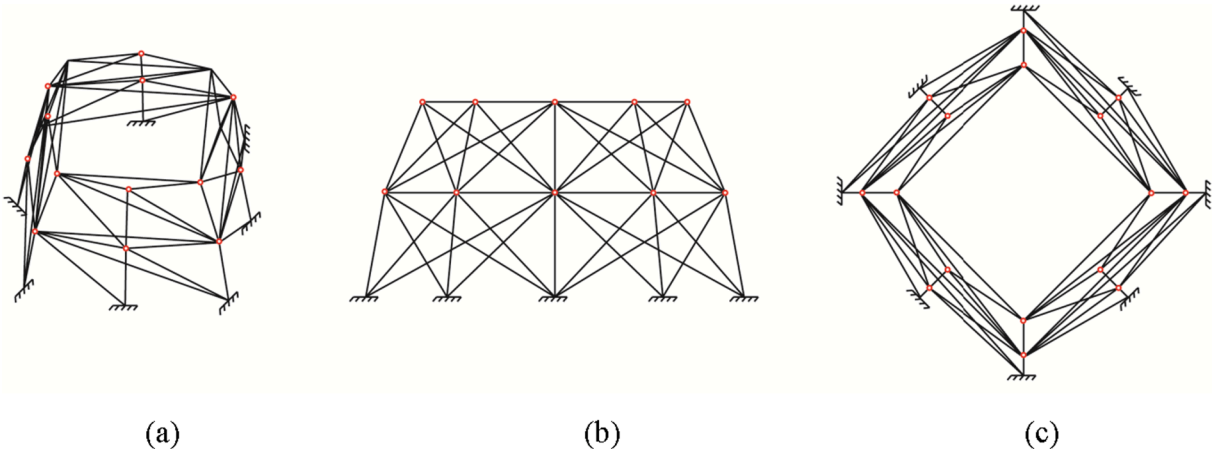

Fig. 14. A two-layer circulant structure: (a) three-dimensional view, (b) lateral view, (c) plan view.

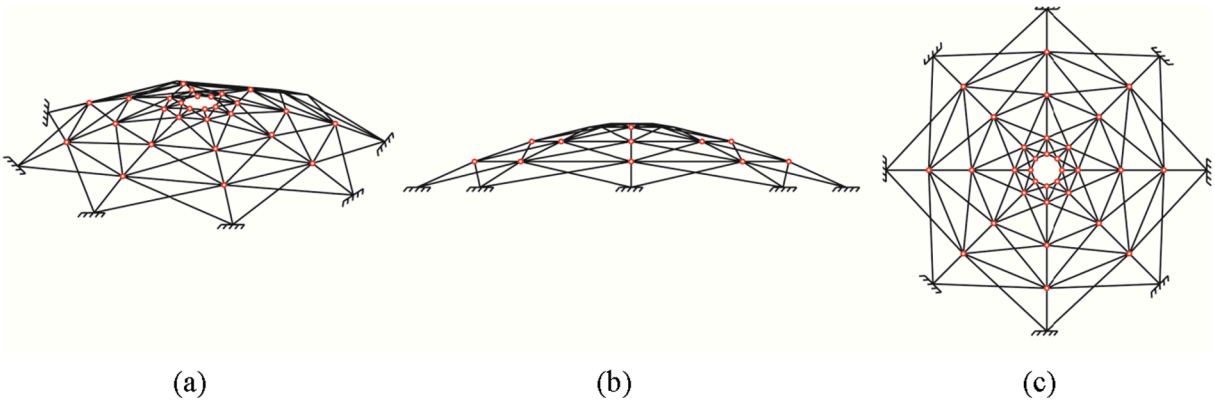

Fig. 15. A three-dimensional dome structure: (a) three-dimensional view, (b) lateral view, (c) plane view.

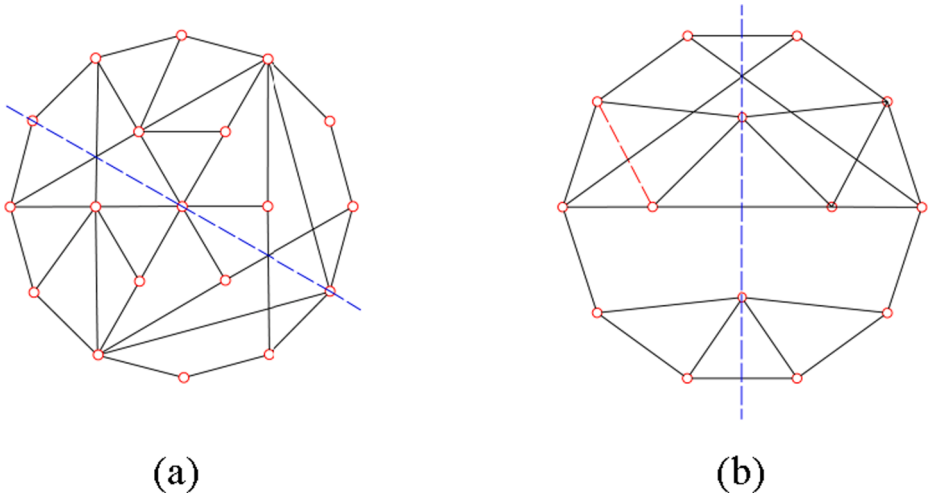

Fig. 16. Two planar structures: (a) C_s, (b) C_1

net structures, trusses, grids, reticulated shells, cable domes, chord domes, and so on. Through the symmetry of plane projections, the actual symmetry of the three-dimensional structures can be predicted. In addition, in the geometric model for each structure, different node symbols can be considered to represent different constraints or node masses, and different line types can be used to represent members with different axial stiffness or section shapes. Consequently, it is necessary to conduct further research to establish an accurate symmetry recognition system. In this study, we may encounter trouble when putting an actual structure in the RGB format into the CNN model, because we should not only consider the characteristics of the image itself (such as illumination, deformation, scale, and blur), but we need to also consider whether the CNN model is familiar with the structural system. It could be possible that with advances in various technologies, a comprehensive database, as well as a nearly perfect model will be established for the symmetry identification of structures. Furthermore, the framework reported in this paper could be expanded and further developed towards the recognition of symmetry in three-dimensional structures.

CRediT authorship contribution statement

Pei Zhang: Methodology, Software, Data curation, Writing – original draft. **Weiying Fan:** Methodology, Software, Data curation, Writing – original draft. **Yao Chen:** Conceptualization, Methodology, Investigation, Supervision. **Jian Feng:** Supervision, Validation. **Pooya Sareh:** Supervision, Writing – review & editing.

Declaration of Competing Interest

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

Acknowledgements

This work has been supported by the National Natural Science Foundation of China (Grants No. 51978150 and 52050410334), Southeast University "Zhongying Young Scholars" Project, and the Fundamental Research Funds for the Central Universities. The corresponding author would like to acknowledge financial support from the Alexander von Humboldt-Foundation for his academic research at Max-Planck-Institut für Eisenforschung GmbH. The authors are grateful to the editors and anonymous reviewers for their professional comments and valuable suggestions which led to the improvement of the quality of the paper.

References

[1] Cicconet M, Birodkar V, Lund M, Werman M, Geiger D. A convolutional approach to reflection symmetry. Pattern Recogn Lett 2017;95:44–50.
[2] Sun C, Sherrah J. 3D symmetry detection using the extended Gaussian image. IEEE Trans Pattern Anal Mach Intell 1997;19:164–8.
[3] Nagar R, Raman S. 3DSymm: robust and accurate 3D reflection symmetry detection. Pattern Recogn 2020;107:107483. https://doi.org/10.1016/j.patcog.2020.107483.

[4] Chen Y, Yan J, Sareh P, Feng J. Nodal flexibility and kinematic indeterminacy analyses of symmetric tensegrity structures using orbits of nodes. Int J Mech Sci 2019;155:41–9.
[5] Nojima T. Modelling of folding patterns in flat membranes and cylinders by Origami. JSME International J Series C-Mechanical Systems Machine Elements and Manuf 2002;45(1):364–70.
[6] Miura K. Map fold a la Miura style, its physical characteristics and application to the space science. Proceedings of International Meeting of Origami Science & Technology 1989.
[7] Maurer PM. A universal symmetry detection algorithm. A universal symmetry detection algorithm. Springerplus. 2015;4(1). https://doi.org/10.1186/s40064-015-1156-7.
[8] Dong Ji-Yang, Zhang Jun-Ying. Detection of the permutation symmetry in pattern sets. Discrete Dynamics in Nat Soc 2006;2006:081503. https://doi.org/10.1155/DDNS/2006/81503.
[9] Sareh P, Guest SD. Design of isomorphic symmetric descendants of the Miura-ori. Smart Mater Struct 2015;24(8):085001. https://doi.org/10.1088/0964-1726/24/8/085001.
[10] Sareh P, Guest SD. Design of non-isomorphic symmetric descendants of the Miura-ori. Smart Mater Struct 2015;24(8):085002. https://doi.org/10.1088/0964-1726/24/8/085002.
[11] Sareh P, Guest SD. Tessellating variations on the Miura fold pattern. South Korea: IASS-APCS Symposium. Seoul; 2012.
[12] Grünbaum B. What symmetry groups are present in the Alhambra? Notices of the American Mathematical Society. 2009;53:670–3.
[13] Makovicky E. Symmetry: Through the eyes of old masters. Boston: Walter de Gruyter. Berlin; 2016.
[14] Chen Y, Yan J, Feng J, Sareh P. A hybrid symmetry-PSO approach to finding the self-equilibrium configurations of prestressable pin-jointed assemblies. Acta Mech 2020;231(4):1485–501.
[15] Funk C, Liu Y. Beyond planar symmetry: modeling human perception of reflection and rotation symmetries in the wild. 2017 IEEE International Conference on Computer Vision (ICCV)2017. p. 793-803.
[16] Kondra S, Petrosino A, Iodice S. Multi-scale kernel operators for reflection and rotation symmetry: further achievements. Proceedings of 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Portland2013.
[17] Dalitz C, Wilberg J, Jeltsch M. The gradient product transform: an image filter for symmetry detection. Image Processing on Line 2019;9:413–31.
[18] Gao L, Zhang L-X, Meng H-Y, Ren Y-H, Lai Y-K, Kobbelt L. PRS-Net: planar reflective symmetry detection net for 3D models. IEEE Trans Vis Comput Graph. 2021;27(6):3007–18.
[19] Aguilar W, Bribiesca E. Symmetry detection in 3D chain coded discrete curves and trees. Pattern Recogn 2015;48(4):1420–39.
[20] Chang W, Song HA, Oh S-H, Lee S-Y. Noise-robust detection of symmetric axes by self-correcting Artificial Neural Network. Neural Process Lett 2015;41(2):179–89.
[21] Zingoni A, Pavlovic MN, Zlokovic GM. A symmetry-adapted flexibility approach for multi-storey space frames. Part 2: symmetry-adapted loads. Structural Eng Rev 1995;7:121–30.
[22] Chen Y, Sareh P, Feng J, Sun Q. A computational method for automated detection of engineering structures with cyclic symmetries. Comput Struct 2017;191:153–64.
[23] Chen Y, Fan L, Feng J. Automatic and exact symmetry recognition of structures exhibiting high-order symmetries. J Computing in Civil Eng ASCE 2018;32(2):04018002.
[24] Chen Y, Sareh P, Feng J. Effective insights into the geometric stability of symmetric skeletal structures under symmetric variations. Int J Solids Struct 2015;69-70:277–90.
[25] Sareh P, Chermprayong P, Emmanuelli M, Nadeem H, Kovac M. Rotorigami: a rotary origami protective system for robotic rotorcraft. Sci Rob 2018;3(22). https://doi.org/10.1126/scirobotics.aah5228.
[26] Sareh P, Guest SD. A framework for the symmetric generalisation of the Miura-ori. Int J Space Struct 2015;30(2):141–52.
[27] Chen Y, Yan J, Feng J, Sareh P. Particle swarm optimization-based metaheuristic design generation of non-trivial flat-foldable origami tessellations with degree-4 vertices. J Mech Des 2021;143:011703.
[28] Zingoni A. On the symmetries and vibration modes of layered space grids. Eng Struct 2005;27(4):629–38.
[29] De Luca F, Hossain MI, Kobourov S. Symmetry detection and classification in drawings of graphs. In: Archambault D, editor. Tóth CD. Cham: Springer International Publishing; 2019. p. 499–513.
[30] Zingoni A. On the best choice of symmetry group for group-theoretic computational schemes in solid and structural mechanics. Comput Struct 2019;223:106101. https://doi.org/10.1016/j.compstruc.2019.106101.
[31] Healey TJ. A group-theoretic approach to computational bifurcation problems with symmetry. Comput Methods Appl Mech Eng 1988;67(3):257–95.
[32] Watada R, Ohsaki M, Kanno Y. Group theoretic approach to large-deformation property of three-dimensional bar-hinge mechanism. Jpn J Ind Appl Math 2019;36(1):177–208.
[33] Zingoni A. Group-theoretic vibration analysis of double-layer cable nets of D-4h symmetry. Int J Solids Struct 2019;176-177:68–85.
[34] Hamermesh M. Group theory and its application to physical problems. Addison-Wesley Pub. Co. 1962.
[35] Zingoni A. Group-theoretic insights on the vibration of symmetric structures in engineering. Philosophical Transactions of the Royal Soc A-Mathemat Phys Eng Sci 2014;372:20120037. https://doi.org/10.1098/rsta.2012.0037.
[36] Zingoni A. Use of symmetry groups for generation of complex space grids and group-theoretic vibration analysis of triple-layer grids. Eng Struct 2020;223:111177. https://doi.org/10.1016/j.engstruct.2020.111177.
[37] Chan JW, Kearney V, Haaf S, Wu S, Bogdanov M, Reddick M, et al. A Convolutional Neural Network algorithm for automatic segmentation of head and neck organs at risk using deep lifelong learning. Med Phys 2019;46(5):2204–13.
[38] Lei X, Pan H, Huang X. A dilated CNN model for image classification. IEEE Access 2019;7:124087–95.
[39] Hu J, Kuang Y, Liao B, Cao L, Dong S, Li P. A multichannel 2D Convolutional Neural Network model for task-evoked fMRI data classification. Computational Intelligence and Neurosci 2019;2019:1–9.
[40] Hsieh T-H, Kiang J-F. Comparison of CNN algorithms on hyperspectral image classification in agricultural lands. Sensors 2020;20(6):1734. https://doi.org/10.3390/s20061734.
[41] Mei S, Ji J, Hou J, Li Xu, Du Q. Learning sensor-specific spatial-spectral features of hyperspectral images via Convolutional Neural Networks. IEEE Trans Geosci Remote Sens 2017;55(8):4520–33.
[42] Kaveh A, Joudaki A. Efficient analysis of block circulant structures. Structures. 2021;34:738–47.
[43] Sareh Pooya. The least symmetric crystallographic derivative of the developable double corrugation surface: Computational design using underlying conic and cubic curves. Materials & Design 2019;183:108128. https://doi.org/10.1016/j.matdes.2019.108128.
[44] Sareh Pooya, Chen Yao. Intrinsic non-flat-foldability of two-tile DDC surfaces composed of glide-reflected irregular quadrilaterals. International Journal of Mechanical Sciences 2020;185:105881. https://doi.org/10.1016/j.ijmecsci.2020.105881.
[45] Talatahari S, Azizi M, Tolouei M, Talatahari B, Sareh P. Crystal structure algorithm (CryStAl): a metaheuristic optimization method. IEEE Access 2021;9:71244–61. https://doi.org/10.1109/ACCESS.2021.3079161.
[46] Talatahari B, Azizi M, Talatahari S, Tolouei M, Sareh P. Crystal structure optimization approach to problem solving in mechanical engineering design. Multidiscipline Modeling in Materials and Structures 2022;18(1):1–23. https://doi.org/10.1108/MMMS-10-2021-0174.
[47] Khodadadi N, Azizi M, Talatahari S, Sareh P. Multi-Objective Crystal Structure Algorithm (MOCryStAl): Introduction and Performance Evaluation. IEEE Access 2021;9:117795–812. https://doi.org/10.1109/ACCESS.2021.3106487.