15th World Congress on Computational Mechanics (WCCM-XV)
8th Asian Pacific Congress on Computational Mechanics (APCOM-VIII)
Virtual Congress: 31 July – 5 August 2022
S. Koshizuka (Ed.)

ACCELERATING SENSITIVITY ANALYSIS IN STRUCTURAL TOPOLOGY OPTIMIZATION USING DEEP NEURAL NETWORK

XIANGRUI. KONG [1]*, YU-CHING. WU [1]

[1]Department of Structural Engineering, College of Civil Engineering, Tongji University
Shanghai, 200092, PR China
*2033792@tongji.edu.cn

Key words: Topology Optimization, Deep Neural Network, Finite Element Method

Abstract. *In recent years, Topology Optimization (TO) has been increasingly gaining attention with the development of new constructing techniques. In the optimization process of a density-based TO method, the sensitivities of the design variables are often computed using numerical methods like Finite Element Method (FEM). Such operation is performed repetitively for tens or hundreds of iteration steps, therefore generating huge computational cost for large scale design scenarios. This paper proposes to accelerate TO by replacing the full-scale sensitivity analysis of FEM with a reduced-scale case, and adopting a deep neural network to map the reduced-scale sensitivity field back to fine scale. Three neural network models are trained and tested using training data generated by structures of three difference scales. The results show that the proposed network successfully reduced the time cost to a large extent, while preserving the topology of the optimized design.*

1 INTRODUCTION

Topology Optimization (TO) is the process of determining the material distribution in a design space given a specific design purpose (structural compliance, heat conduction, etc.). TO has been gaining considerable attention due to recent development of additive manufacturing. The most well known approach is the Solid Isotropic Material with Penalization (SIMP) method [1]. In this density-based method, the design space is first discretized into square elements subject to a set of displacement and load boundary conditions. Finite Element Method (FEM) is then performed iteratively to compute the quantity of interest and the sensitivity of these elements, after which the element densities are updated accordingly.

One significant constraint that holds back the broad application of TO is the expensive computational cost. Due to the repetitive evaluations of the objective function, the process is extremely time-consuming for large-scale design problems. In recent years, the rapid development of deep learning has prepared the researchers with an abundance of tools with great potential to tackle this problem in a data-driven manner. Many researchers attempted to develop surrogate models for the whole TO process using neural networks [2–5]. However, poor or restricted generalization ability and disconnections in the optimized result are reported. Another group of researchers focused on the most time-consuming part of the optimization process, which is the calculation of the objective function or sensitivity analysis [6–9].

This paper proposes to accelerate the TO process through speeding up the sensitivity calculation in each iteration step. This is achieved by training a neural network which infers the fine-scale sensitivity field from a coarse-scale sensitivity and density field. After the model is trained, it is used to replace the traditional full-scale sensitivity calculation with a coarse-scale FEM simulation and a neural network inference step, therefore reducing computational cost while preserving accuracy to a certain degree.

2 PROBLEM STATEMENT

In the SIMP method, each of the elements in the discretized domain is assigned to a continuous density value between 0 and 1. As the optimization process proceeds, most of the element densities in the design domain converge to 0 or 1, which stands for void or solid, respectively. This paper focuses on structural TO, where the most widely-used objective function is minimum compliance, and the corresponding problem statement is described in Eq.(1).

$$\begin{aligned} &\min_{x} : c(\mathbf{x}) = \mathbf{U}^{\mathrm{T}}\mathbf{K}\mathbf{U} = \sum_{\mathrm{e}=1}^{N} (x_{\mathrm{e}})^{p}\, \mathbf{u}_{\mathrm{e}}{}^{\mathrm{T}} \mathbf{k}_0 \mathbf{u}_{\mathrm{e}} \\ &s.t. : \frac{V(\mathbf{x})}{V_0} = f \\ &\mathbf{K}\mathbf{U} = \mathbf{F} \\ &0 < \mathbf{x} \le 1 \end{aligned} \tag{1}$$

where c represents the compliance; $\mathbf{U}$ is the global displacement factor, $\mathbf{K}$ is the global force vector, $\mathbf{F}$ is the global force vector. $\mathbf{u}_{\mathrm{e}}$ represents the element displacement vector, $\mathbf{k}_0$ represents the general element stiffness matrix, $\mathbf{x}$ represents the design variable vector, and x_{e}

is density for a specific element. N is the total number of elements in the discretized domain. p is the penalty factor introduced to achieve 0-1 convergence, and a typical value of 3 is adopted in this work. $V(\mathbf{x})$ and V_0 are the material volume and design domain volume, and f is the prescribed volume fraction.

The optimization problem described by Eq.(1) is solved by the classic optimality criteria method presented in the 88-line SIMP Matlab code paper [1], as shown in Eq.(2).

$$\begin{cases} \max(x_{\min}, x_e - m), & x_e B_e^{\eta} \le \max(x_{\min}, x_e - m) \\ x_e B_e^{\eta} & \max(x_{\min}, x_e - m) \le x_e B_e^{\eta} \le \min(1, x_e + m) \\ \min(1, x_e + m) & \min(1, x_e + m) \le x_e B_e^{\eta} \end{cases} \tag{2}$$

where m is a positive move limit, η is a numerical damping coefficient, and updating factor B_e is derived from the optimality condition, as shown in Eq.(3).

$$B_e = \frac{-\dfrac{\partial c}{\partial x_e}}{\lambda \dfrac{\partial V}{\partial x_e}} \tag{3}$$

where λ is the Lagrangian multiplier and can be solved with a bisection algorithm. The element sensitivities of the compliance and material volume is given in Eqs.(4) and (5). The sensitivities is solved with FEM and is the most time-consuming part of the whole optimization process, especially for large-scale element discretization cases. Therefore, this paper attempts to relieve the expensive computational cost of sensitivities in a data-driven manner.

$$\frac{\partial c}{\partial x_e} = -p(x_e)^{p-1} \mathbf{u}_e{}^{\mathrm{T}} \mathbf{k}_0 \mathbf{u}_e \tag{4}$$

$$\frac{\partial V}{\partial x_e} = 1 \tag{5}$$

3 METHODOLOGY

This paper proposes a deep learning model to accelerate typical TO process in structural mechanics. The model is a Convolutional Neural Network (CNN), and will be addressed as Finite Element Super Resolution (FESR) for simplicity. FESR takes in the sensitivity analysis result of a reduced-scale structure and the difference between the fine and coarse density field, then outputs the full-scale sensitivity field.

3.1 Sensitivity calculation with FESR model

In the sensitivity analysis part of each optimization step, the fine-scale structure is first reduced to coarse-scale by performing an average down-sampling operation to the density field, as shown in Fig. 1. The boundary conditions are reduced proportionally as well, where the finite element analysis is performed to obtain a corresponding coarse-scale sensitivity field. The coarse-scale sensitivity and density are then up-sampled back to fine-scale through

bi-linear interpolation. Taking into account the loss of information caused by the down-sampling operation, the difference of the fine-scale density and the up-sampled density is calculated and combined with the up-sampled sensitivity field to form a two-channel input to the FESR model for inference. The model then outputs the final fine-scale sensitivity field for the optimality criteria to update the density field.

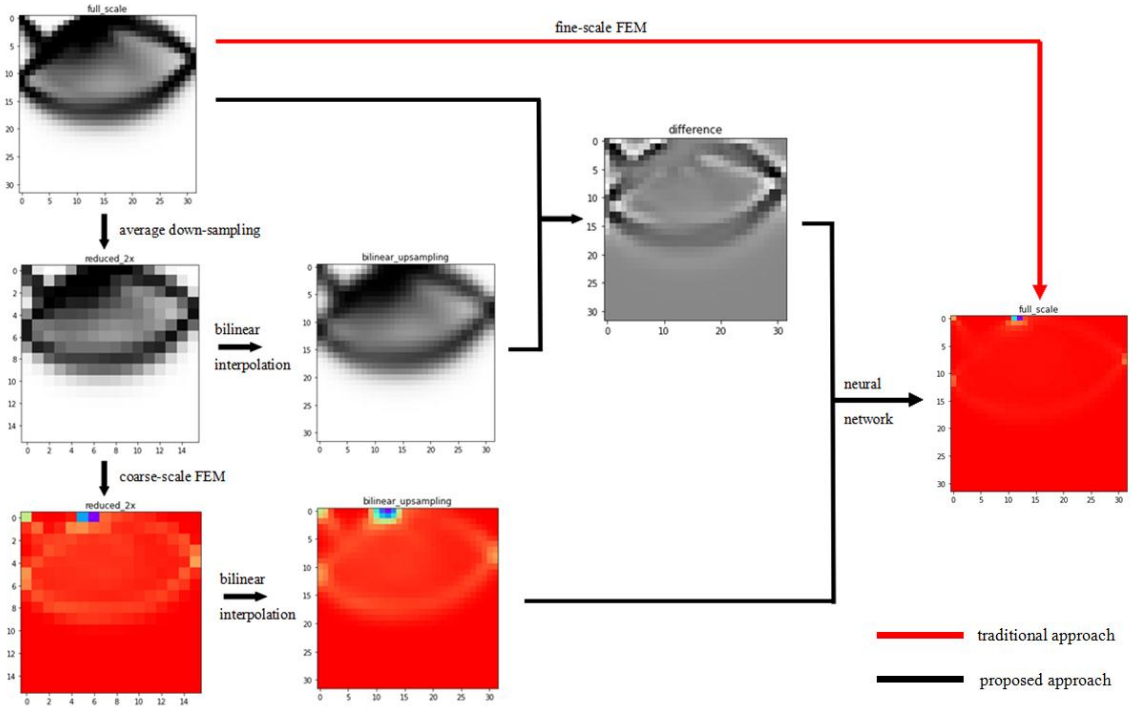

Figure 1: A comparison between the traditional approach of sensitivity analysis and the proposed approach

3.2 FESR model architecture

The FESR neural network architecture adopts the encoder-decoder architecture with SE-Res blocks, which has been successfully applied in relevant research [7,10]. The architecture of the deep neural network is shown in Fig. 2, which contains three convolution layers, five SE-Res blocks, and two de-convolution layers followed by a final convolution layer with kernel size of 1x1 and linear active function to produce outputs of desired range and shape. The detailed structure of the SE-Res block is shown in Fig. 3. In the previous work, the stride of the convolution operation is set to (2,2) for some of the layers. However, from experimental observation, all convolution operations in the presented work adopt a stride of (1,1) to achieve better performance. To give the proposed model a certain degree of transferability, the input size is not fixed to a specific scale.

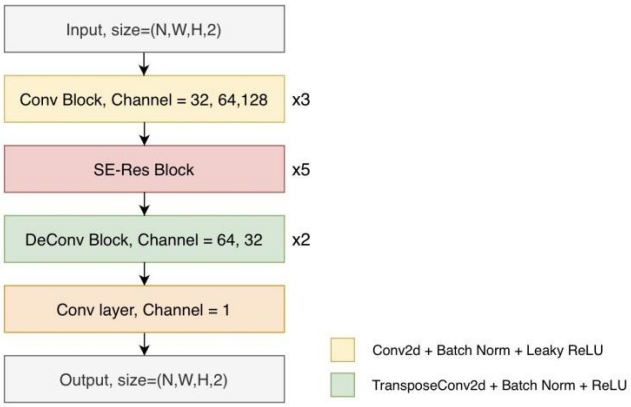

Figure2: The neural network architechture of FESR model.

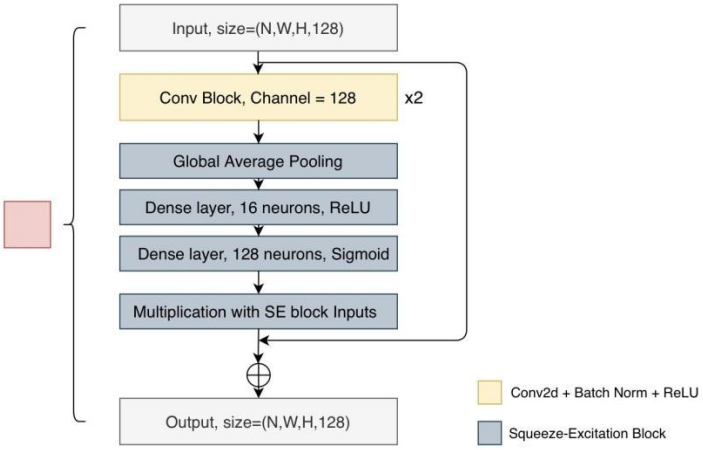

Figure3: The detailed structure of SE-Res block

3.3 Preparation for model training

During the raw data preparation phase, this paper adopts a similar approach with Ref. [9]. 200 random structures with mesh size 128 × 128 are generated. Each structure are assigned with two or three fixed peripheral boundaries on both x and y axis, and two or three unit forces on negative x or y direction are applied on the peripheral boundary. Each structure then go through 20 iteration steps of TO using the SIMP method to obtain full-scale density field. Figure 4 shows two examples of the optimized structure after 20 iterations.

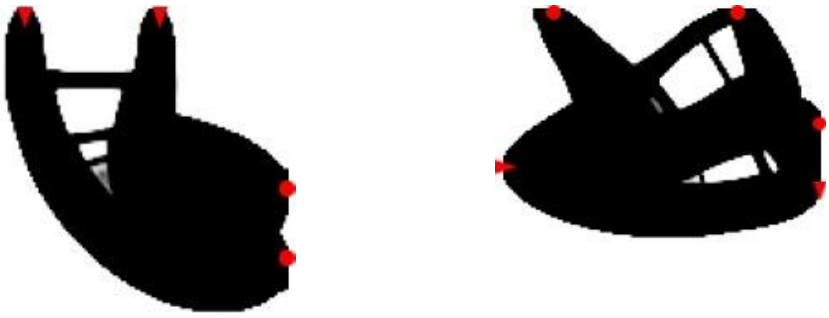

Figure4: Examples of the generated structures for training. Note: Red dot stands for fixed boundary, triangle stands for force position and direction.

After obtaining raw data from TO process, the structure and density field are down-scaled 1, 2 and 3 times (scale reduced by 2×, 4×, 8×) and analyzed with FEM to produce reduced-scale sensitivity fields, as described previously in Section 3.1. The original dataset contains 4000 data for each of the three scales. To prevent extreme value points (often found near the fixed boundary condition point) from scaling the data distribution, the sensitivity fields with a mean value of less than -1 in the 8× case are discarded. The same data case in the 2× and 4× dataset are also discarded to achieve consistency. Data normalization is performed before the data are split up to 80% training data and 20% testing data. The Adam optimizer with a learning rate of 0.001 is chosen, and the loss function is Mean Absolute Error (*MAE*). For each of the three scales, the training undergoes 100 epochs with a batch size of 32.

4 RESULTS AND DISCUSSION

The data generation, model training/inference and TO process are all performed on the cloud platform Google Colab, with Tesla P100 16G-RAM GPU. Each of the three datasets produces a trained neural network with the same FESR architecture, named model-2x, model-4x, model-8x, respectively.

4.1 Training dataset test case

The first test case is a structure with a mesh size of 128 × 128 and the same boundary condition set up as the training dataset. The fine-scale structure is down scaled by 2×, 4× and 8×, which means the FESR model inference will be performed iteratively 1, 2 and 3 times correspondingly in a single iteration step. This set up is to test the possibility of stacking multiple model inferences in order to achieve further time reduction. In addition, two hybrid model cases are performed, which utilize the model on the corresponding scale that matches the mesh size on which the model was trained. For example, in the hybrid-4x model case, the structure will first be down-scaled to 32 × 32 to calculate coarse-scale sensitivity and density difference Then model-4x will be used to infer the 64 × 64 sensitivity field, after which model-2x will combine the sensitivity field generated by model-4x and the 64 × 64 density difference as its input, and output the final full-scale sensitivity result. The density fields for the 50^{th} step of TO are shown in Fig. 5. The time cost for the whole 50 iteration is shown in Table 1.

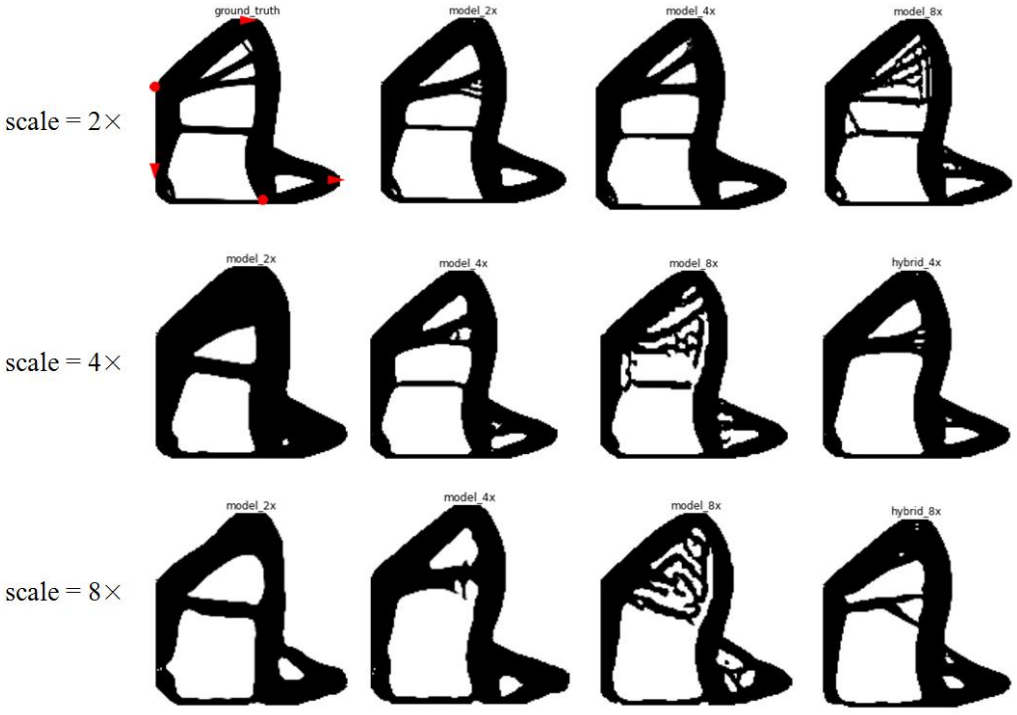

Figure5: Optimization results of a structure with 128 × 128 mesh size

Table 1: Time cost of different models and scales for 50 iteration steps with 128 × 128 mesh size(s)

Scale factor	Ground truth	Model			
		Model-2x	Model-4x	Model-8x	Hybrid model
2 ×	61.0	15.4	15.2	15.2	–
4 ×		7.34	6.96	7.08	6.94
8 ×		8.95	7.89	8.03	8.07

As for optimization quality, model-4x with a scale factor of 2 produces the most similar optimization structure to the ground truth. When the structure is down-scaled for more than 2×, the resulting structures tend to have vague boundaries and disconnections. As for time cost, the 2 × scale reduction brings about a 75% time save compared to the traditional SIMP method. Further scale reduction reasonably saves more time, but comes with the cost of severe quality loss. The similar time cost between 4 × and 8 × can be explained by the fact that the difference of computational cost between 32 × 32 and 16 ×16 mesh size is trivial compared to other optimization procedures.

4.2 Larger mesh size test case

The second test case is a structure with a mesh size of 256 × 256 and the same boundary condition set up as the training dataset, in order to test the generalization ability of the FESR model under the increase of mesh size. The 50^{th}-step optimization result is shown in Fig. 6,

and the time cost of different scales is compared in Table 2. Model-2x and model-4x display a certain degree of generalization ability, and more computational cost reduction is observed.

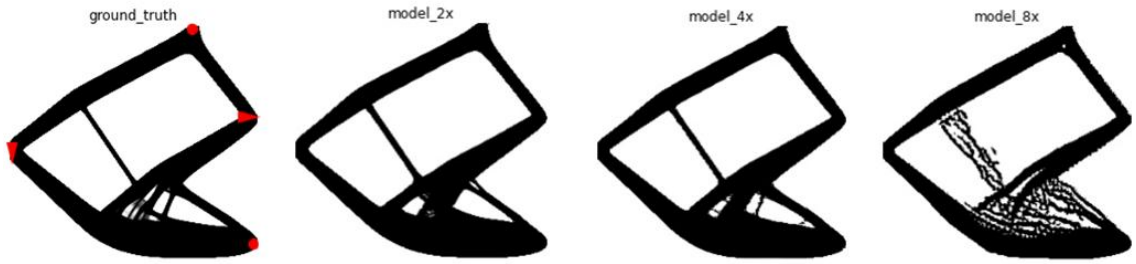

Figure6: Optimization results of a structure with 256 × 256 mesh size

Table 2: Time cost of different models and scales for 50 iteration steps with 256 × 256 mesh size (s)

Scale factor	Ground truth	Model-2x
2×	351.0	69.0
4×		25.1
8×		16.0

4.3 Classic problem test case

The third test case involves two classic TO problems, the Messerschmitt-Bölkow-Blohm (MBB) beam with a mesh size of 120 × 40 and the L-shape beam with a mesh size of 128 × 128. These two examples serve as a further confirmation of the transferability of the FESR model. Model-2x with a 2× scale reduction is tested in this case. The results are shown in Fig. 7.

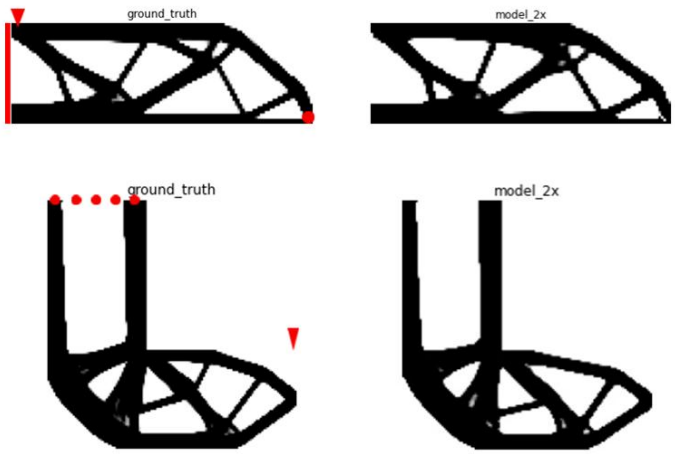

Figure7: Optimization results of two classical TO problems

4.4 Discussion

The numerical experiments show that the proposed model effectively relieves the computation expense to a certain degree. It is reasonable to believe structures with mesh size larger than the ones adopted in the experiment will benefit even more. However, when adopting a scale reduction of more than 2 ×, the accumulated sensitivity prediction error of the model and the resulting distortion of the updated density field lead to trivial structural topologies. The relationship among models trained on different scales and how to integrate them to achieve greater generalization ability and accuracy requires further investigation. Particularly, one research that addresses the super-resolution of whole slide imaging might be inspiring, where a Recurrent Convolutional Neural Network (RCNN) is proposed to string together different resolution versions of a single image in one model [11].

5 CONCLUSIONS

- In this paper, a deep neural network is proposed to relieve the expensive computational cost of the sensitivity analysis in SIMP. The proposed neural network adopts the encoder-decoder architecture with SE-Res blocks. The model takes in the coarse scale sensitivity field and the difference between fine and coarse scale density field, and outputs a fine scale sensitivity field. By replacing the large scale FEM simulation with a coarse scale case plus one or several neural network inference operations, the optimization process is speeded up 75% (for 128 × 128 mesh size and 2 × scale reduction) and 80% (for 256 × 256 mesh size and 2 × scale reduction) while achieving consistency with respect to the final topology of the optimized structure.
- However, it is observed that although further acceleration could be achieved, the accuracy of the model will be undermined as the scaling factor reaches 4 × and 8 ×. Two different directions holds to potential of tackling this restriction. One is to improve the learning ability of a single deep learning model to directly output finer sensitivity results (4 ×, 8 ×, etc.), the other lies in the integration of models trained at different scale. Both directions might benefit from the research area of image super-resolution (SR), while further effort is still needed to fine-tune the image SR models for TO process.

REFERENCES

[1] Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B. S. and Sigmund, O. Efficient topology optimization in MATLAB using 88 lines of code. *Struct. Multidiscip. Optim.* (2011) **43**: 1–16.

[2] Yu, Y., Hur, T., Jung, J. and Jang, I. G. Deep learning for determining a near-optimal topological design without any iteration. *Struct. Multidiscip. Optim.* (2019) **59(3)**: 787–799.

[3] Li, B., Huang, C., Li, X., Zheng, S. and Hong, J. Non-iterative structural topology optimization using deep learning. *Comput. Aided Des.* (2019) **115**: 172–180.

[4] Nie, Z., Lin, T., Jiang, H. and Kara, L. B. TopologyGAN: topology optimization using generative adversarial networks based on physical fields over the initial domain. *J. Mech. Des.* (2021) **143(3):** 031715.
[5] Zheng, S., He, Z. and Liu, H. Generating three-dimensional structural topologies via a U-Net convolutional neural network. *Thin-Walled Struct.* (2021) **159**: 107263.
[6] Qian, C. and Ye, W. Accelerating gradient-based topology optimization design with dual-model artificial neural networks. *Struct. Multidiscip. Optim*. (2021) **63(4)**: 1687–1707.
[7] Tan, R. K., Qian, C., Wang, M. and Ye, W. An efficient data generation method for ANN-based surrogate models. *Struct. Multidiscip. Optim.*(2022) **65(3)**: 1–22.
[8] Qian, C., Xu, D. and Ye, W. An adaptive and scalable ANN-based model-order-reduction method for large-scale TO designs. *arXiv preprint* (2022): 2203.10515.
[9] Yue, T., Yang, H., Du, Z., Liu, C., Elkhodary, K. I., Tang, S. and Guo, X. A mechanistic-based data-driven approach to accelerate structural topology optimization through finite element convolutional neural network (FE-CNN). *arXiv preprint* (2021): 2106.13652.
[10] Nie, Z., Jiang, H. and Kara, L. B. Stress Field Prediction in Cantilevered Structures Using Convolutional Neural Networks. *J. Comput. Inf. Sci. Eng.* (2020) **20**: 1–8.
[11] Mukherjee, L., Bui, H. D., Keikhosravi, A., Loeffler, A. and Eliceiri, K. W. Super-resolution recurrent convolutional neural networks for learning with multi-resolution whole slide images. *J. Biomed. Opt.* (2019) **24(12)**: 126003.