Topology Optimization Using Convolutional Neural Network

Baki Harish, Kandula Eswara Sai Kumar, and Balaji Srinivasan

Abstract Topology optimization is a method to find the optimal material distribution of a structure by minimizing the objective function under the design and limit constraints. In this paper, we developed a deep learning-based machine learning algorithm to get the optimized structure for the given input conditions of a structure. We trained convolutional neural network (CNN)-based encoder–decoder architecture using the existing dataset as target images and input conditions modeled as input images. The target images are the optimized structures, developed using the MATLAB open-source topology optimization code, generated by varying the volume fraction from 5 to 95% with an increment of 5% and Poisson's ratio varied from 0.01 to 0.49. The input conditions, i.e., the volume fraction and Poisson's ratio are modeled as input images. In the present study, four types of input images and two encoder–decoder architectures are developed, and their performance is studied using identity mapping to obtain the optimized structure of a cantilever beam which is fixed at one end and a constant load is applied at the other end.

Keywords Topology optimization · Machine learning · Convolution–deconvolution network

1 Introduction

Structural topology optimization finds the optimal material distribution of a structure by minimizing the objective function under the design constraints. This method is widely used to design the low-weight structures which increase the efficiency of the

B. Harish (✉) · K. Eswara Sai Kumar · B. Srinivasan
Indian Institute of Technology Madras, Chennai, Tamil Nadu 600030, India
e-mail: harish.iiitn@gmail.com

K. Eswara Sai Kumar
e-mail: me15d417@smail.iitm.ac.in

B. Srinivasan
e-mail: sbalaji@iitm.ac.in

R. R. Salagame et al. (eds.), *Advances in Multidisciplinary Analysis and Optimization*, Lecture Notes in Mechanical Engineering,
https://doi.org/10.1007/978-981-15-5432-2_26

system. Sigmund et al. [1] provided free MATLAB source code to solve the topology optimization problem for compliance minimization.
Bruns et al. [2] first introduced the solid isotropic material with penalization (SIMP) method in topology optimization. The basic principle behind the SIMP method is it uses a density design variable that depends on material constitutive law and penalizes the intermediate density material in combination with an active volume constraint.
Sosnovik et al. [3] first introduced the deep learning-based convolution–deconvolution network model to topology optimization and significantly improved the efficiency of the optimization process by considering the problem as an image segmentation task. In his model, intermediate results of SIMP are taken as inputs and final optimized structures are taken as targets. His model successfully mapped the intermediate results of SIMP with optimized final structures and decreased the computation time and cost significantly. However, his work did not consider the initial conditions for topology optimization, and the accuracy of the result heavily relies on the first few iterations. Thus, this method cannot completely eliminate the traditional SIMP method.
Banga et al. [4] have developed 3D convolution–deconvolution network, which takes intermediate 3D images from traditional solvers as inputs and final optimal 3D designs as targets. This method also reduced the time consumption, but this method also did not start from initial conditions.
In the present work, two types of deep learning-based encoder–decoder networks are developed, and four types of input images are modeled from input conditions. The present work focused on modeling the input conditions as input images to the proposed encoder–decoder network and mapping these input images with final optimal designs.

2 Methodology

2.1 Topology Optimization

The structural topology optimization is used to solve compliance minimization, eigenfrequency maximization, and design of compliant mechanisms of the structures. The compliance minimization problem is defined as:

$$\min \quad C(\mathbf{x}) \tag{1}$$

$$\text{Subjected to} \quad \mathbf{Ku} = \mathbf{f} \tag{2}$$

$$V(\mathbf{x}) \leq V^* \tag{3}$$

$$0.001 \leq \mathbf{x} \leq 1 \tag{4}$$

where $\mathbf{x}$ is the design variable vector, $\mathbf{K}$ is stiffness matrix, $\mathbf{u}$ is displacement vector, $\mathbf{f}$ is force vector, $V(\mathbf{x})$ is the design volume, and V^* is required volume.

In the present study, a 2D cantilever beam fixed at one end and a constant load applied at other end is considered to work on. The beam will have unique optimized design corresponding to volume percentage, Poisson's ratio, fixed boundary conditions, and application of force location. For the sake of simplicity, only volume percentage and Poisson's ration are considered as variables for optimization.

2.2 Architecture of Encoder–Decoder Network

The encoder–decoder network is a deep neural network, which consists of two parts:

1. **Encoder network**: The input image is down-sampled by a series of convolution and pooling operations to a final vector of size 1024. The input image is reduced almost 20 times.

2. **Decoder network**: The output vector from encoder part is up-sampled by a series of deconvolution and upsampling operations to a final image of size same as the target image.

We have developed two types of encoder network. First network has two convolution operations and one fully connected operation as shown in Fig. 1. Second network has three convolution operations and one fully connected operations as shown in Fig. 2. In both networks, decoder network is the same. The convolution filters are squared matrices which are the weights we acquire after training. The pooling operation which we used here is average pooling, it reduces the convolved layer to the target size by doing averaging of elements. The convolution and pooling operations are said to be one convolution layer. The convolution and upsampling operations together said to be one deconvolution layer. In the upsampling operation, the convolved layer is expanded to a target size. Though explicitly not mentioned, ReLU is the activation function in every convolution and deconvolution operation and the mean square error is the cost function.

2.3 Preparing Target Data Set

We have generated optimized structures of the cantilever beam by varying the aspect ratio from 5 to 95% with an increment of 5% and Poisson's ration from 0.01 to 0.49 with an increment of 0.01 using the MATLAB code. These images are used as target dataset for encoder–decoder network. Each image in the target dataset has a size of (100×100).

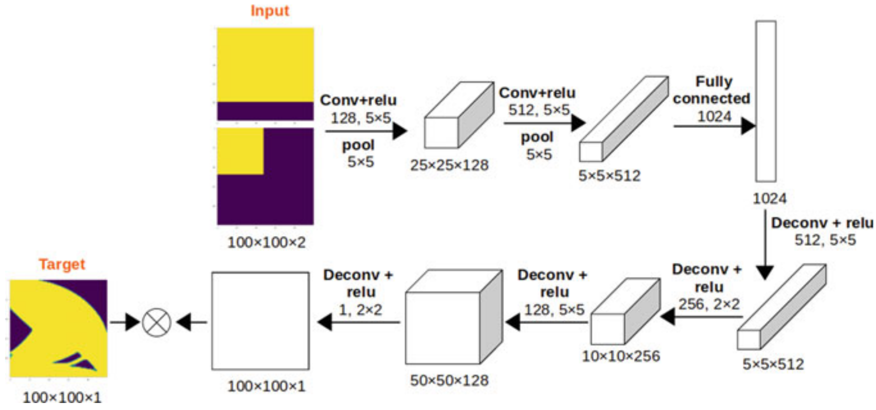

Fig. 1 1st encoder–decoder architecture

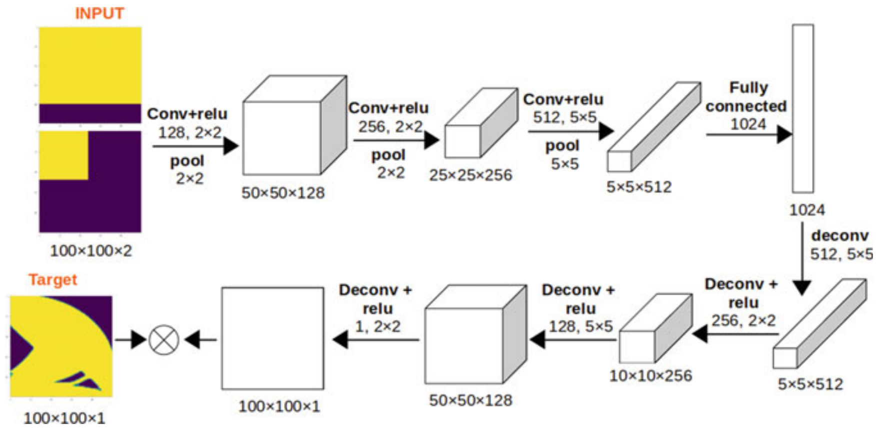

Fig. 2 2nd encoder–decoder architecture

2.4 Modeling the Input Images

We have modeled four types of input images by combining volume percentage and Poisson's ratio in different combinations.

Input type 1 images consist of two layers. First layer is (100×100) size image, has volume percentage information, modeled by filling volume percentage number of rows with ones completely and rest by zeros. Second layer is (100×100) size image, has Poisson's ratio information, modeled by filling $100 \times$ Poisson's ratio of rows and columns with ones and rest by zeros. Refer to Fig. 3a.

Input type 2 images consist of two layers. First layer has volume percentage information, modeled similar to Input type 1. Second layer is (100×100) size image,

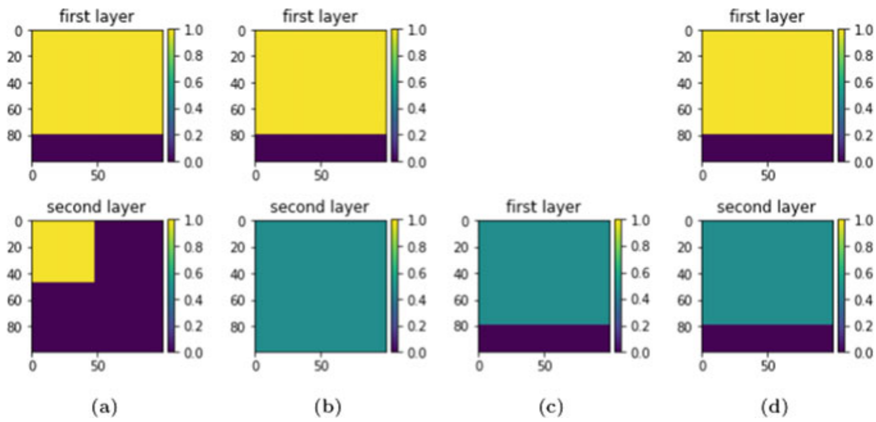

Fig. 3 Features represented as inputs for volume percentage of 80% and Poisson's ration of 0.49

has Poisson's ratio information, modeled by filling entire layer with Poisson's ratio. Refer to Fig. 3b.

Input type 3 images consist of one layer, which has both volume percentage information and Poisson's ratio information, modeled by filling volume percentage of rows with Poisson's ratio completely and rest by zeros. Refer to Fig. 3c.

Input type 4 images consist of two layers. First layer has volume percentage information, modeled similar to Input type 1. Second layer is (100 × 100) size image, has Poisson's ratio information, modeled by filling volume percentage of rows with Poison's ratio completely and rest by zeros. Refer to Fig. 3d.

2.5 Training the Network

The input and target datasets have total 931 images, in which 70% of data is taken as training set, 20% of data is taken as cross-validation set, and 10% of data is taken as testing set. Training the first network took 6 h for 2000 epochs, whereas second network took 12 h approximately.

3 Results and Discussion

To demonstrate the results after training the encoder–decoder network with input features given as input images, we have taken input and target images for volume of 45, 55, 65%, and Poisson's ration of 0.2. Figure 4 shows the results of first encoder–decoder network, and Fig. 5 shows the results of second encoder-decoder network for all 4 types of inputs. From the figures, it is clear that the second encoder–decoder

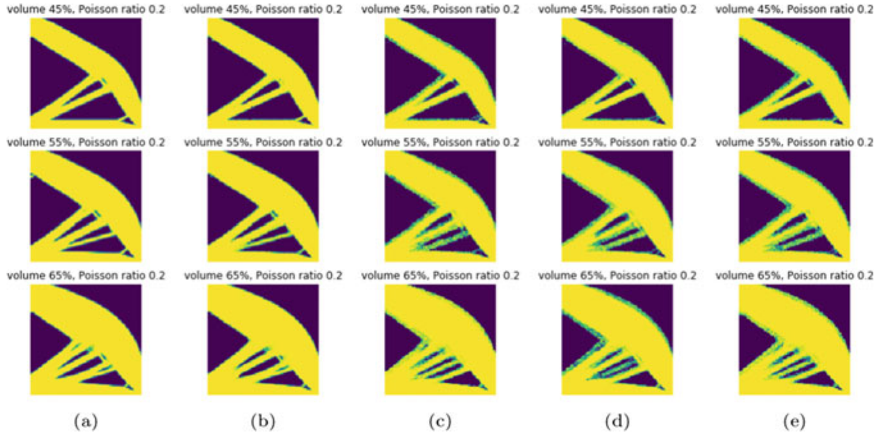

Fig. 4 Results of 1st encoder–decoder network for volume of 45, 55, 65%, and Poisson's ration of 0.2

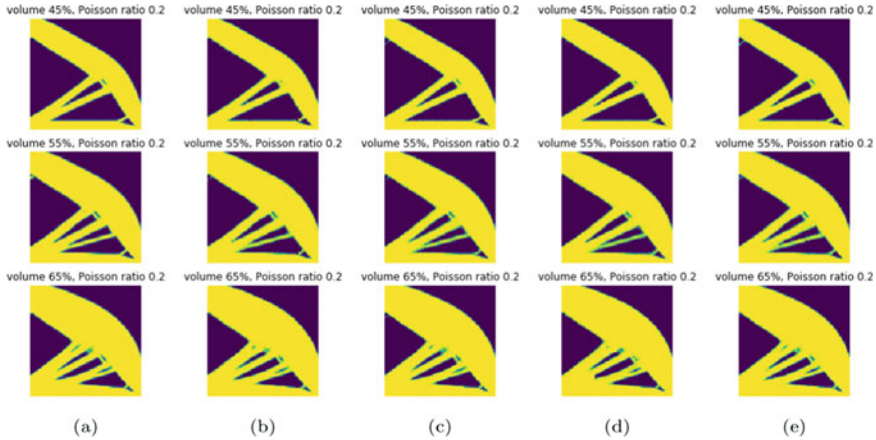

Fig. 5 Results of 2nd encoder–decoder network for volume of 45, 55, 65%, and Poisson's ration of 0.2

network combination with input type 4 captures very fine details, which are necessary to consider for the continuity in volume of the original beam. The MSE of 100 input type 4 images tested on the second encoder–decoder network reaches 0.025, which is far below than rest of the input types.

4 Conclusion

In the present work, we have developed a encoder–decoder network, which takes input conditions, i.e., volume fraction and Poisson's ratio as input images and optimized structures correspond to the same input conditions generated by MATLAB code as target images. While training takes 12 h, which is considerably huge, testing only takes seconds. Using MATLAB to generate one image takes 5 min, whereas using this proposed network takes only 10 s, which is 30 times faster than MATLAB code.

References

1. O. Sigmund, A 99 line topology optimization code written in matlab. Struct. Multidisc. Optim. **21**(2), 120–127 (2001)
2. T.E. Bruns, A reevaluation of the SIMP method with filtering and an alternative formulation for solidvoid topology optimization. Struct. Multidisc. Optim. **30**(6), 428–436 (2005)
3. I. Sosnovik, I. Oseledets, *Neural Networks for Topology Optimization*. arXiv preprint arXiv:1709.09578 (2017)
4. S. Banga et al., *3D Topology Optimization using Convolutional Neural Networks*. arXiv preprint arXiv:1808.07440 (2018)