https://doi.org/10.18321/ectj1471

Reduction of Material Usage in 3D Printable Structures Using Topology Optimization Accelerated with U-Net Convolutional Neural Network

J. Rasulzade[1], Y. Maksum[2,3]*, M. Nogaibayeva[2], S. Rustamov[1], B. Akhmetov[2,4]

[1]Department of Computer Science and Information Sciences, ADA University,
61 Ahmadbey Aghaoglu str., Baku, 1008, Azerbaijan
[2]Department of Mechanical Engineering, Satbayev University, 22 Satbaev str., Almaty, 050013, Kazakhstan
[3]Department of Chemical Engineering, University of Birmingham, Birmingham, B15 2TT, UK
[4]Department of Mechanical and Aerospace Engineering, Nanyang Technological University,
50 Nanyang Ave., 639798, Singapore

Article info

Received:
14 May 2022

Received in revised form:
26 June 2022

Accepted:
6 August 2022

Keywords:
Material reduction, 3D printing, Topology optimization, Convolutional neural network.

Abstract

Today's 3D printers allow the creation of very advanced structures from various materials, starting from simple plastics up to metal alloys. Since the printing time and amount of material used to create structures are considered very important in terms of cost and energy consumption, it is better to optimize structures for that particular application taking into account all the conditions. In the current work, U-Net convolutional neural network-based topology optimization method (TO) that allows to reduce the material usage and eventually reduces the cost of 3D printing is introduced. The results showed that the accuracy of the method is highly reliable and can be used for designing various 3D printable structures and it applies to any type of materials since properties of materials can be included in TO.

1. Introduction

With the rise of 3D printing technologies, possibilities for optimization of structures – made of different materials, such as plastics, composites, metals and alloys – moved to the next level [1]. As shown in Fig. 1, 3D printers are highly efficient in reducing material cost, due to their layered deposition of material approach compared to the traditional approach [2]. Although, when the latter is coupled with a mathematical method – topology optimization (TO), the reduction of material usage is further decreased, while keeping the stiffness of 3D printed structures at a required level.

TO is a method for optimizing the distribution of material within a given space for a given set of loads and boundary conditions. The purpose of TO is to determine the optimal distribution of the material in the design area, thus reducing the weight of the structure and material cost (i.e. filament cost) during 3D printing. This method is based on repeated steps of design analysis and updating [3], which finally provides the best layout possible. Although, when the design of solid structures is complicated and has multiple initial and boundary conditions, the computational time of TO drastically increases and it takes to get optimized results up to several hours. Such computational difficulty is mainly due to the iterative nature of solvers designed for a large system of linear equations obtained from finite element analysis [4].

There are several methods to improve the computational time of TO. One way is to apply parallel computing based on multi-core central processing units (CPUs) [5–7], and graphics processing units (GPU) [8–10]. Thus, the computationally expensive portions of TO algorithm, especially the parts related to finite element analysis (FEA), are run in parallel among the cores of several CPU or GPU

*Corresponding author.
E-mail: yxm099@bham.ac.uk

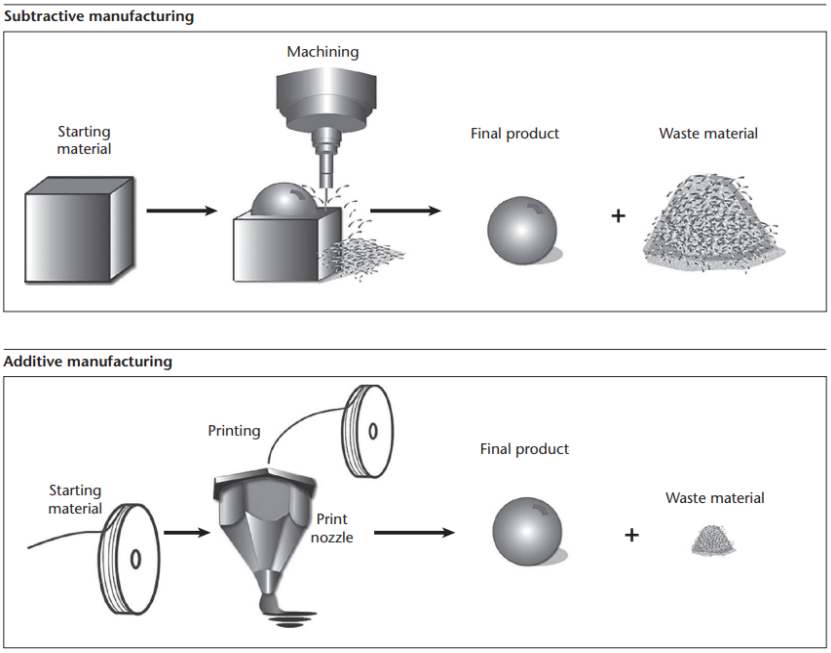


Fig. 1. Additive manufacturing vs subtractive manufacturing [2].

processors. Such an approach was well-known and applied intensively until artificial intelligence (AI) methods started to advance. Thus, in the latest trends, we can notice that AI technologies and open source AI platforms, machine learning (ML) methods in advanced manufacturing processes have expanded significantly [11, 12]. This trend is also noticeable in the field of topology optimization, where, with the help of trained neural networks, structural topology and properties such as strength, modulus of elasticity, strain, and stress fields are predicted [13–15]. In the work [14], a convolutional neural network (CNN) was utilized to train the intermediate topology layouts obtained by SIMP-based TO to solve a solid mechanics problem. To forecast the optimum structures using AI method, the TO solver was stopped at an intermediate point after a few iterations and further fed into the CNN model. Comparing the trained CNN model to the conventional SIMP method, it was discovered that the trained CNN model could predict the final topology optimized structures up to 20 times faster with a few uncommon pixel-wise alterations.

Wang et al. [15] introduced a CNN with perceptible generalization ability for TO. Encoding and decoding components of the neural network that was proposed by the authors enable down- and up-sampling operations. They compared their results with SIMP method and showed that they can achieve about 96% accuracy in predicting optimized topology with the increasing number of CNN training samples.

Nie et al. [16] proposed a new data-driven TO model, named TopologyGAN, which uses physical fields computed in the original material domain as inputs to a conditional generative adversarial network (cGAN) generator. They showed that TopologyGAN is much more effective in reducing the mean square error and mean absolute error compared to cGAN. In the generator, they also introduced a hybrid network, particularly, U-SE (Squeeze-and-Excitation)-ResNet. The results showed good accuracy but some structures were unphysical, therefore their model needed some improvements at the neural network level.

In the current work, we develop a CNN using U-net and test them for predicting the topology of a cantilever beam. We compare the computational cost of the proposed method compared to the conventional SIMP method. We also consider the proposed TO approach, in terms of reducing the material cost for 3D printing applications. Thus, the further sections are developed in the following order: firstly, we briefly discuss TO and its role in 3D printable design samples. Furthermore, we introduce CNN method used in the current study and continue with the results and discussion section. Lastly, conclusive remarks are provided and future works are discussed.

2. Topology optimization and structural design

2.1. General knowledge

Topology optimization is usually achieved

through numerical calculations, where the design domain is discretized by finite elements. In other words, the finite element method (FEM) is the main numerical tool. FEM-based topological optimization is classified as Isotropic Solid/Empty (ISE), Anisotropic Solid/Empty (ASE), and Isotropic Solid/Empty/Porous (ISEP) topologies. Among them, the most practical class is ISE, where FEM elements are considered either filled with a selected isotropic material or void [17, 18]. There are several works that compare different methods in terms of computational efficiency [3, 18–22]. The use of solid structures based on isotropic materials is widespread in all sectors of the manufacturing and construction industries. Similarly, 3D printing of structures using a single material is the most reliable and practical, because a stronger bond between printed layers is easier to achieve compared to a multi-material approach.

TO problem can be described as a search for a material distribution that minimizes the objective function F under a volume constraint G_i. The distribution of the material is described by a variable density $\rho(x)$, which can take on the value 0 (void) or 1 (solid material) at any point in the design plane Ω. The optimization problem can be written in mathematical form as [3]:

$$\begin{aligned} \min_{x} \quad & :F = F(\boldsymbol{u}(\rho),\rho) = \int_{\Omega} f(\boldsymbol{u}(\rho),\rho)dV \\ \text{subject to} \quad & :G_0(\rho) = \int_{\Omega} \rho(\boldsymbol{x})dV - V_0 \le 0 \\ & :G_i(\boldsymbol{u}(\rho),\rho) \le 0,\ i = 1,\dots,M \\ & :\rho(x) = 0\ or\ 1, \forall \boldsymbol{x} \in \Omega \end{aligned} \quad (1)$$

where $\boldsymbol{u}$ corresponds to the state equation. For simplicity of further notation, it can be assumed that the objective function can be calculated as an integral over a local function $f(\boldsymbol{u}(\rho),\rho)$. Also, since in real conditions there are several restrictions, M additional restrictions are included in the general formulation.

So far, several methods have been developed for solving TO problems: the density approach, the level-set approach, the phase field approach, and the discrete approaches [3]. Several papers compare the advantages and disadvantages of each of these methods in terms of their computational efficiency [5–8]. Among them, the density approach method, particularly SIMP method (solid penialized isotropic material) is commonly used by researchers/engineers and it is now commonly used in commercial software such as SolidWorks, COMSOL and ANSYS. Thus, a common TO problem based on the SIMP approach, where the goal is to minimize the degree of conformity (i.e., maximize the rigidity of the design), can be written as [23]:

$$\begin{aligned} \min_{x} \quad & :C(x) = \boldsymbol{U}^T\boldsymbol{K}\boldsymbol{U} = \sum_{e=1}^{N} (x_e)^p u_e^T k_0 u_e \\ \text{subject to} \quad & :\frac{V(x)}{V_0} = f \\ & :\boldsymbol{K}\boldsymbol{U} = \boldsymbol{F} \\ & :0 < x_{min} \le x \le 1 \end{aligned} \quad (2)$$

where C – compliance; $\boldsymbol{U}$ and $\boldsymbol{F}$ – global displacement and force vectors; $\boldsymbol{K}$ – global stiffness matrix; u_e and k_e – elements displacement and stiffness vectors; x – design variable vector; x_{min} – minimum relative densities; N – number of FEA elements, p – penalty factor, $V(x)$ and V_0 – volume of material and design domain, and f – volume fraction of material. The optimization problem can be solved using various iterative methods such as Optimality Criteria (OC) method, Sequential Linear Programming (SLP) method, or Moving Asymptotes (MMA) method [24].

Within the last decade, TO method has been significantly improved [3], and its application to solve complex problems has expanded significantly [25]. Various multi-physics problems such as enhancing heat transfer processes [26, 27], reducing liquid and solid interactions [28], optimizing the thermoelastic behaviour of structures [29], and improving the geometry of electro-thermal mechanical drives [30, 31] are addressed using TO approaches, where given geometries are numerically discretized in 2D or 3D domains and optimized using TO for better performance. Although, the application of TO on such problems with multiple interacting physical properties in 2D or 3D domains leads to a noticeable increase in computational complexity, and hence an increase in the computation time of TO analysis. Therefore, in the current work, we address this use and try to improve it using machine learning methods.

2.2. Design example for 3D printing

As an example, in the current research work, we decided to study the TO of the Messerschmitt-Bölkow-Blohm (MBB) beam as shown in Fig. 2. In fact, this is one of the well-known structures that

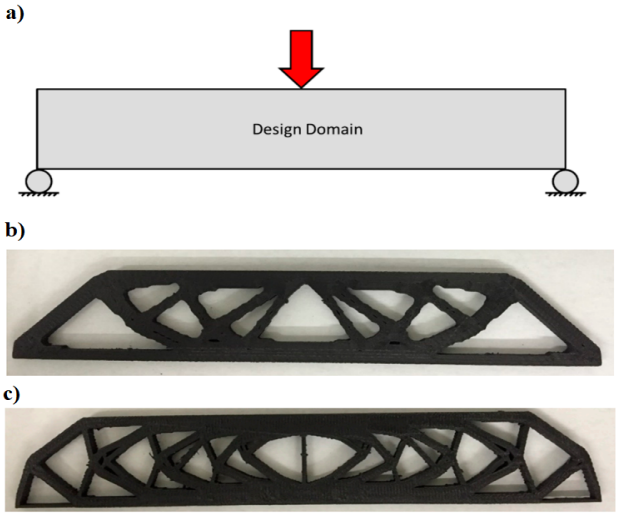


Fig. 2. Messerschmitt-Bölkow-Blohm (MBB) beam: a) – boundary conditions continuous fiber angle optimization; b) – continuous fiber angle optimization; c) fixed horizontal print direction [32].

are previously studied in most of the works that include machine learning studies [14, 16, 17]. In this example, continuous fiber angle optimization was considered together with TO so that 3D printed structure has a strength depending on the printing direction. The structure can be further tested using a mechanical machine as shown in Fig. 3 [32]. In the particular example, the volume fraction *f* was set to 0.4, meaning 60% of the material compared to the original MBB beam was saved. It should be noted that, in the current studies, the main purpose is to use ML methods for acceleration of TO, and reducing the computational time. Furthermore, we discuss the material cost reduction that can be achieved by TO approach.

Here, we consider the half of MBB beam due to its symmetricity and apply TO on it to develop an

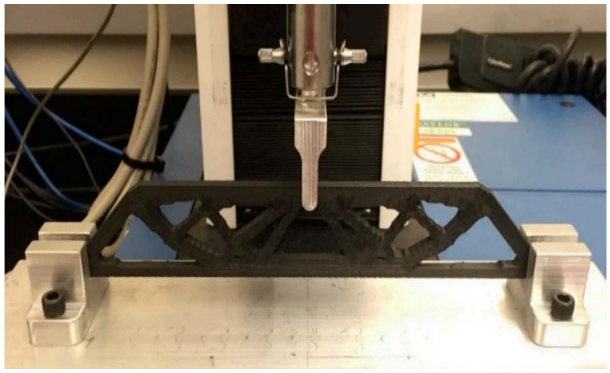
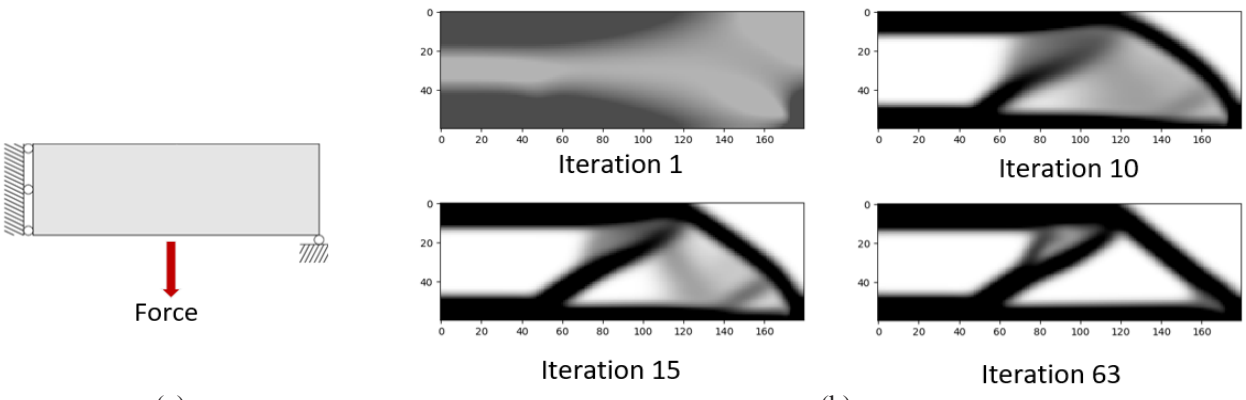


Fig. 3. Testing of 3D printed structure [32].

optimized material distribution for given boundary conditions. As can be seen from a) part of Fig. 4, there is a force (F) applied to the beam, while it is fixed at the right lower corner. The results obtained from TO using SIMP method show that the overall contour/shape of the possible structure is achieved after 15 iterations, while an additional 48 iterations are required to obtain the final finely tuned binary result. In other words, it is clear how the final structure would eventually get shaped after 15 iterations, but the method further has to go through more iterations (i.e. up to 60) to get the final reduced material distribution. According to our studies based on an open-source SIMP algorithm from [23], for a computer with Intel Core i7-4510 at 2.00 Ghz, the time for one iteration is around 0.88 sec. In total, to run 80 iterations to get the final TO, the overall computational time is about 70.4 sec. Once again, it should be mentioned that the mesh size in the current study is 120×40, which is 4800 FEM elements only. Usually, for complex structures, element size might go up to several thousand and millions. Therefore, the importance of accelerating the computational time is actual and needs proper studies.



(a) (b)

Fig. 4. Messerschmitt-Bölkow-Blohm (MBB) beam: (a) – design area and boundary conditions; (b) – intermediate and final TO results based on the SIMP method.

To accelerate this process of TO to obtain material distribution for a given boundary and initial constraints in a shorter time, the authors propose to use a machine learning algorithm using neural networks. In such a way, it is possible to improve the overall duration from designing structures to 3D printing process.

3. Machine learning approach

3.1. Main idea

U-net is a special type of convolutional neural network, which was introduced by the Faculty of Computer Science at the University of Freiburg, and is designed to solve problems related to image segmentation. It consists of two main stages: convolution and development. At each step of the first stage (convolution), the model generates N reduced (compressed or convolved) filtered versions of the image, and at each step, the number of filters will be doubled. The subsequent stage (sweep) is the inverse version of the convolution. At each step of this stage, the images will be increased in size and filtered, with the initial number of filters corresponding to the final number of filters of the 1st stage, and the number of filters is halved at each step. The key action in this model is to store and transfer intermediate images from the convolution stage to the unwrapping stage, with the result of the first convolution step being used in the last unwrapping step, the second in the penultimate one, and so on. The final solution to this problem is an image of the same size as the original one.

3.2. Intermediate stages, layers

As mentioned above, U-net is a special type of CNN, which consists of the following layers: convolution, pooling, dropout and up-sampling. The convolutional layer is a set of maps (a set of matrices) with trainable ts (in different sources it is called differently: a scanning core or a synaptic core), which have a small receptive field but pass through the entire depth of the input volume. Within the framework of this experiment, at each convolution step, the number of filters with the ReLU (Rectified Linear Unit) activation function will increase by two, with an initial number of sixteen. Further, at each sweep step, the number of filters will be halved. The last set of sixteen filtered images will be filtered one last time with the sigmoid activation function.

A pooling layer is a selective sampling process that aims to reduce the resolution of an input matrix. Within the framework of this experiment, horizontal and vertical steps equal to two (N = 2, M = 2) was used to select the area, and the maximum value of each area was assigned to the resulting matrix.

$$f'(x,y) = [\, f(N * x + i, M * y + j)] \tag{3}$$

The dropout layer is a regularization technique used in artificial neural networks to prevent overfitting. Essentially, it zeros out random values in the input dataset and increases the efficiency of machine learning algorithms.

$$\begin{aligned} &w'_j = \{w_j, \text{ with } P(C), \text{otherwise } 0 \\ &P(c) - \text{probability conversation matrix} \\ &w_j - \text{initial matrix before dropout} \\ &w'_j - \text{matrix after dropout} \end{aligned} \tag{4}$$

Up-sampling layer is the process of increasing the width and length of the input matrix by N and M times, respectively. In experiments, this matrix was filled in according to the following formula, where N = 2, M = 2:

$$f'(x,y) = f\left(int\left(\frac{x}{N}\right), int\left(\frac{y}{M}\right)\right) \tag{5}$$

where $f'(x,y)$ – final matrix, $f(x,y)$ – initial matrix.

4. Architecture of experiment, training data and evaluation methods

4.1. First stage: Convolution

The architecture of the first stage corresponds to the usual architecture of a convolutional neural network. It consists of several alternating steps that conditionally indicate the depth of the model. The input data of the first step are based on two images: N-th iteration of the SIMP and the difference

between N-th and the previous image. Each step starts by applying two reconvolutions with 3×3 kernels with an increasing number of filters and a ReLU activation function, with the intermediate dropout layer of 10%. This is followed by the layer of reduction with a step of 2×2. In this experiment, the initial number of filters was taken as 16 and for each step, it was doubled. Also, the final result of each step is saved for transfer to the next stage.

4.2. Second stage: Deconvolution

The deconvolution stage (second stage) is the inverse of the first stage and differs in that as input data for each step, in addition to the received data, the data previously stored in the convolution stage is used. At each step, two 3×3 reconvolutions occur with a fitting number of filters and ReLU activation function, with the intermediate dropout layer of 10%. This is followed by a layer of magnification with a factor of two vertically and horizontally. After the 4th, final iteration, the result obtained passes through the last convolution layer with a single filter and a sigmoid activation function. After rounding the result, a binary image with the original resolution is obtained, where 0 corresponds to the void, and 1 to the material.

4.3. Training data

To implement the above model, images from iterations of the SIMP model along with the expected result are needed. Synthetic data created by Sosnovik and Oseledet [14], using an automatic 2D and 3D topology generator – an open-source SIMP Topy algorithm [33] – was used for training the CNN. In the end, 10000 imaginary samples were generated and used, and 100 SIMP iterations were run for each given boundary and initial conditions, along with the expected result. In Table 1, three different samples of input data, particularly, N-th iteration, the gradient (difference of N-th and (N-1)th iterations), and the final topology layout are shown. The grid size of the samples generated is 40×40 as can be noted in Table 1, and each topology sample is a tensor of 100×40×40, based on the grid size and the number of iterations, which is 100.

4.5. Evaluation method

This evaluation method checks the identity of each pixel in the image. Conventionally, it is denoted by β and given that each pixel of the image has a set of values consisting of two digits (zero to indicate emptiness and one to indicate material). Thus, binary accuracy is calculated according to the formula:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{6}$$

The coefficient of Jaccard or the intersection-to-union ratio is used to calculate the level of overlap between the predicted image and the true image.

$$\beta = \frac{\omega_{00} + \omega_{11}}{n_0 + n_1} \tag{7}$$

Due to the fact that there are only two classes of interest, the original formula can be modified to the following view:

$$\text{K} = 0.5 * \left[\frac{\omega_{00}}{n_0 + \omega_{10}} + \frac{\omega_{11}}{n_1 + \omega_{01}}\right] \tag{8}$$

In all of the above formulas, ω_{tp} is the number of instances of class t predicted as class p, and n_x this is the total number of instances of class x in the original image.

Mean absolute error (MAE) is also used to evaluate the accuracy of the model. MAE is a measure of the error difference between two observations that reflect the same phenomena.

$$MAE = \frac{1}{M}\sum_{i=1}^{M}|y^{(i)} - y'^{(i)}| = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{N}\sum_{j=1}^{N}|y_j^{(i)} - y'^{(i)}_j| \tag{9}$$

In the scope of this experiment, it will measure how much predicted image differs from the desired image.

5. Results and discussions

At the initial stage of the study, two experiments are carried out to determine the effect of depth on the CNN model. Here, depth means the number of layers in the neural network. To fully characterize

Table 1
Input data: samples obtained SIMP for Nth iteration, gradient and final topology for given set of boundary and initial conditions

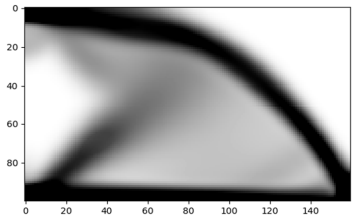
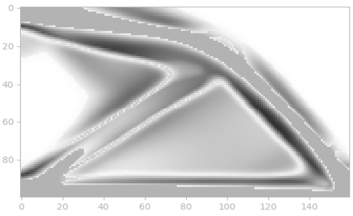
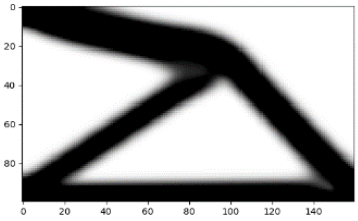
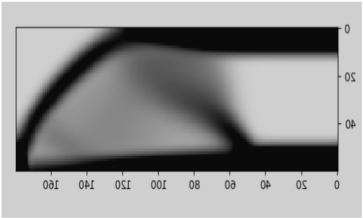
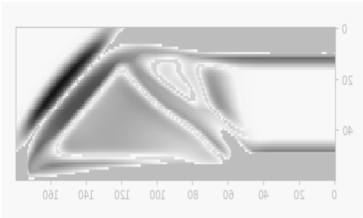
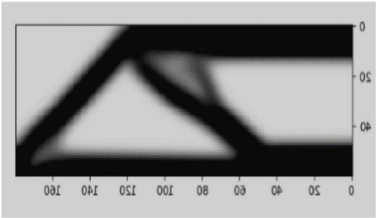
	N-th iteration of SIMP	Gradient	Final topology
Sample 1			
Sample 2			

Table 2
Initial parameters of the model

No.	Distribution	Depth	Optimizer	Number of filters
1	Uniform [1-100]	3	Adam optimization algorithm	16, 32, 64, 64, 32, 16
2	P (λ = 5)	3		16, 32, 64, 64, 32, 16
3	P (λ = 10)	3		16, 32, 64, 64, 32, 16
4	P (λ = 30)	3		16, 32, 64, 64, 32, 16
5	Uniform [1-100]	4		16, 32, 64, 128, 128, 64, 32, 16
6	P (λ = 5)	4		16, 32, 64, 128, 128, 64, 32, 16
7	P (λ = 10)	4		16, 32, 64, 128, 128, 64, 32, 16
8	P (λ = 30)	4		16, 32, 64, 128, 128, 64, 32, 16

the problem of TO under consideration, two different depths with different sizes are studied: in the first experiment, the depth is taken to be 3, while in the second one, a maximum depth is set to 4. The maximum depth is limited by the image resolution and it is one unit less than the minimum logarithm of the resolution to the base of two. For this case (40×40), this value is equal to four.

Also, each experiment has four sections, where the effectiveness of the choice of initial data is also investigated. To run the experiment, along with a uniform distribution, the Poisson distribution is also used. It should be noted that the Poisson distribution is taken with three different coefficients P (λ = 5), P (λ = 10) and P (λ = 30), and experiments are carried out for each coefficient. Table 2 gives the main initial parameters of all models. As can be seen from Table 2, 8 experiments are carried out, 4 experiments with a depth of 3, and another 4 experiments with a depth of 4. Adam optimization algorithm is used in all experiments to update the learning rate of each network weight individually. The number of filters is also provided in Table 2, and as explained above they are doubled at the convolution stage and halved at the sweep stage.

Thus, the initial parameters for all experiments, as well as which data are used, and how the model is built are presented in Table 2. As a result, 8 different experiment results from 8 different CNN are received. Each model is evaluated by comparing it with different SIMP iterations. Thus, in this experiment, SIMP iterations from 5th to 80th, with

Table 3
Results of the experiment with the depth of three layers. P – distribution, PP – uniform distribution, P(x) – Poisson distribution with coefficient x

No	R	5	10	15	20	30	40	50	60	70	80
1	RR	93.6	95.5	96.2	96.7	97.2	97.7	97.7	97.9	98.1	98.2
2	$P\ (\lambda = 5)$	94.0	95.7	96.2	96.6	96.9	97.3	97.5	97.5	97.6	97.7
3	$P\ (\lambda = 10)$	94.1	95.5	96.1	96.5	96.8	97.2	97.4	97.5	97.5	97.6
4	P $(\lambda = 30)$	93.9	95.9	96.9	97.2	97.6	97.8	98.0	98.2	98.3	98.4

Table 4
Results of the experiment with a depth of four layers. P – distribution, PP – uniform distribution, P(x) – Poisson distribution with coefficient x

No	R	5	10	15	20	30	40	50	60	70	80
1	RR	93.8	95.4	96.3	96.7	97.2	97.5	97.7	97.9	97.9	98.0
2	$P\ (\lambda = 5)$	94.1	95.6	95.9	96.4	96.9	97.1	97.3	97.5	97.6	97.5
3	$P\ (\lambda = 10)$	93.9	95.4	96.0	96.4	96.9	97.1	97.4	97.5	97.5	97.6
4	$P\ (\lambda = 30)$	93.5	95.7	96.5	96.7	97.4	97.7	97.9	98.1	98.2	98.2

an interval of 5 iterations, are selected as shown in Tables 3 and 4. To obtain a satisfactory average accuracy of 94%, 5/10 iterations are sufficient. Summing up the intermediate results, it can be noted that each of these iterations is tested using the presented CNN model, particularly U-net model.

Thus, the results of the first 4 experiments are listed in Table 3, and the next four in Table 4 below. Based on these tables, it can be concluded that the accuracy of the presented U-net model is high and it can be used to speed up SIMP method. It can also be noted that the minimum accuracy of the model equal to 94% was observed when using the fifth iteration. Compared to the previous studies by [15], our results are significantly accurate even using 10.000 datasets. Wang et al. [15] achieved an accuracy of 94% when the data set increased to 20.000. Therefore, our current U-net model is simpler in terms of neural network complexity, which requires less time to obtain the topology, is more accurate, and can performance better accuracy with less training data set. Moreover, the maximum accuracy of 99% for material distribution is achieved using 80th iteration. At the same time, it should be emphasized that in this case, changing the depth of the model had a minimal effect on the results. This can be explained by the fact that the image has a small size, and increasing the depth from 3 to 4 does not make a big difference, but increases the computational complexity and runtime of the

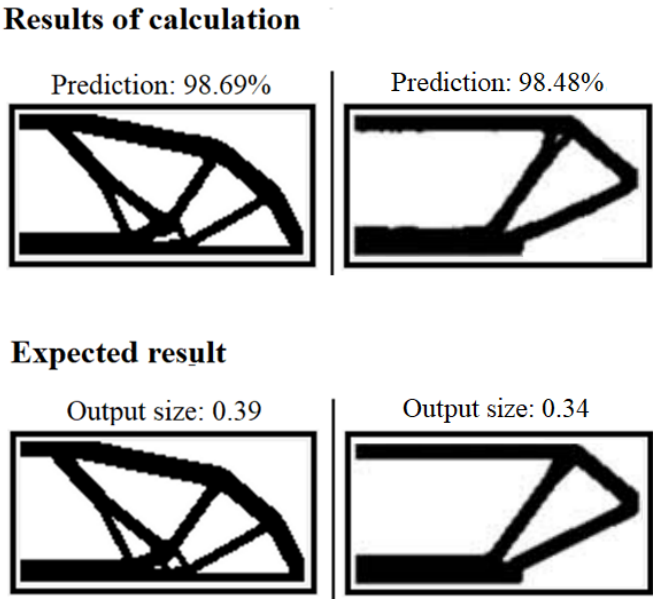


Fig. 5. Visualization of results.

CNN model. Therefore, U-net with depths of 3 (i.e. 3 neural network layers) is better to be selected for faster and more accurate results. It should be noted that the computational time of the trained model here is less than 1 sec to get the final topology, and it is comparable to a single iteration of SIMP method.

The binary and MAE of the studies that are obtained using (7)-(9) are 0.97 and 0.0312 respectively.

Moreover, the accuracy results of our studies using U-net based CNN are compared with previous results from the literature as shown in Table 5. The comparative analysis is based on the same number of training data samples, but different CNN depths that result in the accuracy. Thus, as can be seen from Table 5, our CNN has less number of layers, which directly correlates with faster computational time. In terms of accuracy, our result is better than Wang et al. [15] and comparable with the accuracy of Sosnovik and Oseledet [14].

Furthermore, Fig. 5 shows the expected results (bottom row) and calculation results (top row) of the presented model for three different cases for the material distribution using TO. It should be noted that the accuracy in all cases shown is greater than 98%. Furthermore, such accurate results obtained using neural networks are also 3D printable and allow for acceleration of the development process of 3D printable structures.

The TO is highly reliable and helps to reduce the material cost for 3D printing while keeping the mechanical or other properties of the structures at the same level as the original one. In the examples presented in Fig. 5, the volume fraction of the TO is set to 50% and it allows to reduce the material usage by 50%. Such an approach with the faster method of obtaining topologically optimized structures using CNN, can give many advantages to engineers who use 3D printing technologies for the design and construction of advanced structures.

As a future work, the current CNN model will be extended and improved to develop structures in 3D domain, that are highly required for the engineering of advanced structures. In future work, the main goal will be to keep the computational time as low as possible, while increasing the accuracy and robustness of CNN models, including the applicability of such approaches in 3D printing.

Table 5
Comparison of the current work with previous literature with the same number of training data

Model type	Number of samples	Depth	Accuracy
Our model	10.000	4	~94 (5th iteration)
CNN by Wang et al. [15]	10.000	8	~89
CNN by Sosnovik and Oseledet [14]	10.000	6	~ 92-95 (5th iteration)

6. Conclusions

This work is devoted to the development of a method for accelerating the topology optimization (TO) of 3D printable structures. TO is a reliable mathematical method that allows for reducing material usage while developing advanced structures that are 3D printable. Although, the computational time of TO becomes costly with the increasing complexity of the boundary and initial conditions of the structural analysis. Here, a machine learning method based on a convolutional neural network (CNN) is proposed to accelerate the computational time.

Thus, as an example, MBB beam is considered in the work. The model based on the U-Net architecture is presented to speed up the computation of the TO under consideration, which in turn, reduces the design and development time of the structure. The results of the study are checked for identity using the binary identity by comparing it with the conventional TO method SIMP. Hence, the results showed that CNN model could give 95% of accuracy for 10th iteration of SIMP. Furthermore, when they are compared at 80th iterations, the accuracy of CNN showed much more accurate results and it is about 98%. In addition, it is noticed that changing the depth from level 3 to level 4 does not improve the accuracy of the experiment, although it increases the computational complexity and task execution time. Therefore, it is concluded that by keeping the depth as low as possible, still highly accurate results can be achieved, but within a shorter time and it is very important in reducing the computational time. As shown in the results, the proposed method is highly accurate and can be used for the generation of optimized material distribution to various structures with a given boundary and initial conditions. Such structures are 3D printable, and highly efficient and less material costly compared to structures developed using subtractive manufacturing. Using TO method developed here, the amount of material to develop structures can be significantly reduced while maintaining its strength.

Acknowledgment

This work has been supported by the research programme of Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan (Grant No. AP08856141).

References

[1]. M. Shams, Z. Mansurov, C. Daulbayev, B. Bakbolat, *Eurasian Chem.-Technol. J.* 23 (2021) 257–266. DOI: 10.18321/ectj1129

[2]. J.N. Levesque, A Shah, S Ekhtiari, J.R. Yan, et al., *EFORT Open Reviews* 5 (2020) 430–441. DOI: 10.1302/2058-5241.5.190024.

[3]. O. Sigmund, K. Maute, *Struct. Multidis. Optim.* 48 (2013) 1031–1055. DOI: 10.1007/s00158-013-0978-6

[4]. J. Martínez-Frutos, D. Herrero-Pérez, *Comput. Methods Appl. Mech. Eng.* 311 (2016) 393–414. DOI: 10.1016/j.cma.2016.08.016

[5]. T. Borrvall, J. Petersson, *Comput. Methods Appl. Mech. Eng.* 190 (2001) 6201–6229. DOI: 10.1016/S0045-7825(01)00216-X

[6]. A. Mahdavi, R. Balaji, M. Frecker, E.M. Mockensturm, *Struct. Multidisc. Optim.* 32 (2006) 121–132. DOI: 10.1007/S00158-006-0006-1

[7]. K. Vemaganti, W.E. Lawrence, *Comput. Methods Appl. Mech. Eng.* 194 (2005) 3637–3667. DOI: 10.1016/j.cma.2004.08.008

[8]. S. Schmidt, V. Schulz, *Comput. Visial Sci.* 14 (2011) 249–256. DOI: 10.1007/s00791-012-0180-1

[9]. K. Suresh, *Struct. Multidisc. Optim.* 42 (2010) 665–679. DOI: 10.1007/s00158-010-0534-6

[10]. T. Zegard, G.H. Paulino, *Struct. Multidisc. Optim.* 48 (2013) 473–485. DOI: 10.1007/s00158-013-0920-y

[11]. K. Paraskevoudis, P. Karayannis, E.P. Koumoulos, *Processes* 8 (2020) 1464. DOI: 10.3390/pr8111464

[12]. N.S. Johnson, P.S. Vulimiri, A.C. To, X. Zhang, et al., *Addit. Manuf.* 36 (2020) 101641. DOI: 10.1016/j.addma.2020.101641

[13]. K. Guo, Z. Yang, C.-H. Yu, M.J. Buehler, *Mater. Horiz.* 4 (2021). DOI: 10.1039/d0mh01451f

[14]. I. Sosnovik, I. Oseledets, *Russ. J. Numer. Anal. Math. Model.* 54 (2019) 215–223. DOI: 10.1515/rnam-2019-0018

[15]. D. Wang, C. Xiang, Y. Pan, A. Chen, et al., *Eng. Optim.* 54 (2022) 973–988. DOI: 10.1080/0305215X.2021.1902998

[16]. Z. Nie, T. Lin, H. Jiang, L.B. Kara, *J. Mech. Des.* 143 (2021) 031715. DOI: 10.1115/1.4049533

[17]. M.P. Bendsøe, S*truct. Optim.* 1 (1989) 193–202. DOI: 10.1007/BF01650949

[18]. G.I.N. Rozvany, *Struct. Multidisc. Optim.* 37 (2009) 217–237. DOI: 10.1007/s00158-007-0217-0

[19]. T.P. Ribeiro, L.F.A. Bernardo, J.M.A. Andrade, *Appl. Sci.* 11 (2021) 2112. DOI: 10.3390/app11052112

[20]. D.J. Munk, G.A. Vio, G.P. Steven, *Struct. Multidisc. Optim.* 52 (2015) 613–631. DOI: 10.1007/s00158-015-1261-9

[21]. J.D. Deaton, R.V. Grandhi, *Struct. Multidisc. Optim.* 49 (2014) 1–38. DOI: 10.1007/s00158-013-0956-z

[22]. O. Sigmund, *Struct. Multidisc. Optim.* 21 (2001) 120–127. DOI: 10.1007/s001580050176

[23]. K. Svanberg, *Int. J. Numer. Methods. Eng.* 24 (1987) 359–373. DOI: 10.1002/nme.1620240207

[24]. C. Lundgaard, J. Alexandersen, M. Zhou, C Schousboe, et al., *Struct. Multidisc. Optim.* 58 (2018) 969–95. DOI: 10.1007/s00158-018-1940-4

[25]. T. Dbouk, *Appl. Therm. Eng.* 112 (2017) 841–854. DOI: 10.1016/j.applthermaleng.2016.10.134

[26]. E. Wadbro, M. Berggren, *SIAM Review* 51 (2009). DOI: 10.1137/070699822

[27]. A. Neofytou, F. Yu, L. Zhang, H.A. Kim. Level Set Topology Optimization for Fluid-Structure Interactions. AIAA 2021-1686. Session: Shape and Topology Optimization I, 2021. DOI: 10.2514/6.2021-1686

[28]. T. Gao, P. Xu, W. Zhang, *Comput. Struct.* 173 (2016) 150–160. DOI: 10.1016/j.compstruc.2016.06.002

[29]. F.J. Ramírez-Gil, E.C.N. Silva, W. Montealegre-Rubio, *Comput. Methods Appl. Mech. Eng.* 302 (2016) 44–69. DOI: 10.1016/j.cma.2015.12.021

[30]. F.J. Ramírez-Gil, C.M. Pérez-Madrid, E.C.N. Silva, W. Montealegre-Rubio, *Sustain. Comput. Informatics Syst.* 30 (2021) 100481. DOI: 10.1016/j.suscom.2020.100481

[31]. D. Jiang, R. Hoglund, D.E. Smith, *Fibers* 7 (2019). DOI: 10.3390/FIB7020014

[32]. W. Hunter, et al., (2017), Topy - topology optimization with python, https://github.com/williamhunter/topy