Advances in Engineering Software 173 (2022) 103240

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Research paper

Deep learning paradigm for prediction of stress distribution in damaged structural components with stress concentrations

Hamed Bolandi [a,*], Xuyang Li [a], Talal Salem [a], Vishnu Naresh Boddeti [b], Nizar Lajnef [a]

[a] Department of Civil and Environmental Engineering, Michigan State University, East Lansing, MI 48824, United States of America
[b] Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, United States of America

ARTICLE INFO

Keywords:
Deep learning
Finite element analysis
Stress distribution
Structural engineering

ABSTRACT

Scientists and engineers agree that solving complex problems requires integrating traditional physics-based modeling techniques with state-of-the-art deep learning (DL) methods. This paper aims to integrate physics knowledge into a convolutional neural network (CNN) to boost learning within a feasible solution space in a specific domain. Our proposed method uses deep neural networks in the form of (CNNs) augmented with custom loss functions which uses physics rules to bypass the need for Finite Element Analysis and predict high-resolution stress distributions on damaged steel plates with variable loading and boundary conditions. We embedded physics constraints into the loss function to enforce the model training, precisely capturing stress concentrations around the tips of various structural damage configurations. The CNN was designed and trained to use the geometry, boundary conditions, and load as input and predict the stress contours. The proposed framework's performance is compared to Finite-Element simulations using partial differential equation (PDE) solver. The trained DL model can predict the stress distributions of damaged steel plates with a mean absolute error of 0.22% percent and an absolute peak error of 1.5% for the Von Mises stress distribution

1. Introduction

Accurately and efficiently predicting physical responses is essential for different real-world applications, such as the prediction of the remaining life of mechanical systems [1], earthquake alarms [2], and weather forecasting [3]. Although data-driven and physics-based solutions allow for solid predictions, both methods still suffer from several limitations. The drawbacks of data-driven methods are the requirement for large amounts of data, inability to produce physically consistent results, and weakness of generalize to out-of-sample scenarios. [4]. However, physics-based models, such as Finite Element Analysis (FEA), are computationally prohibitive. Therefore, to achieve fast analysis of mechanical systems and address deficiencies of data-driven models, we integrate conventional physics-based methods with state-of-the-art Deep Learning (DL) methods to predict stress distributions in damaged steel components.

Numerical analysis methods, such as FEA, are typically used to conduct stress analysis of various structures and systems for which it is impractical or hard to determine an analytical solution. Researchers commonly use FEA methods to evaluate the design, safety, and maintenance, of different structures in various fields, including aerospace, automotive, architecture, and civil structural systems. The current workflow for FEA applications includes: (i) Modeling the geometry and its components, which can be time-consuming based on the system complexity; (ii) Specifying material properties, boundary conditions, and loading ; (iii) Applying a meshing strategy for the geometry. The time-consuming and complexity of the current FEA workflow make it impractical in real-time or near real-time applications, such as in the aftermath of a disaster or during extreme disruptive events that require immediate corrections to avoid catastrophic failures.

Based on the steps of FEA described above, performing a complete stress analysis with conventional FEA has a high computational cost. We proposed DL methods [5,6] in our previous work [7] to resolve this issue, bypassing FEA once trained. This work added various types of cracks (damage) into the samples and enhanced our model with a physics equation to predict stress distributions in damaged steel plates.

Our approach in a sense could be viewed as a surrogate for FEA software and it avoids the computation bottlenecks in FEA. In particular, our model predicts the stress distributions and stress concentrations of the most common gusset plates used in infrastructures, such as bridges and buildings. The main idea here is to train a generalized model than can later be used in situations where real time estimations are needed, such as in the aftermath of extreme disruptive events. For example, focusing on critical structural components, there is a need

* Corresponding author.
E-mail address: bolandih@msu.edu (H. Bolandi).

https://doi.org/10.1016/j.advengsoft.2022.103240
Received 17 January 2022; Received in revised form 29 June 2022; Accepted 7 August 2022
Available online 19 August 2022

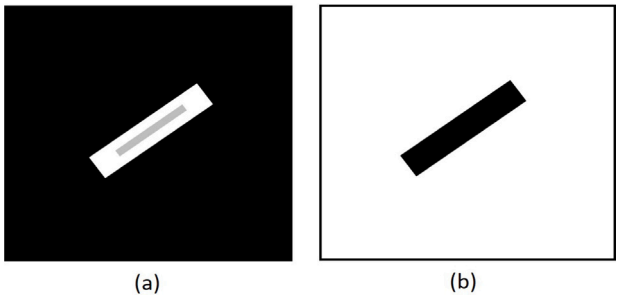

Fig. 1. Illustration of binary masks. (a) mask 1, (b) mask 2.

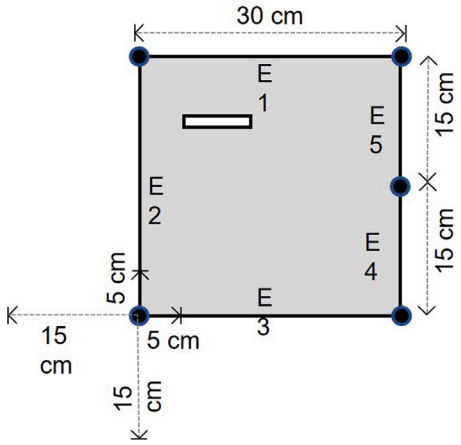

Fig. 2. Basic schematic topology for initializing the damaged steel plate geometries.

for immediate assessment following a disaster or during extremely disruptive events to guide corrective actions. Engineers could rely on the proposed computationally efficient algorithms in order to determine stress distributions over damaged gusset plates and apply the proper rehabilitation actions. It is important for them to be able to analyze damaged gusset plates quickly and accurately, which is exactly what our model can provide. To the best of our knowledge, this work is the first work to use the stress concentration factor equation to predict stress distribution and stress concentration in the specific domain of damaged steel plates.

2. Related works

The most recent works in data-driven applications have included design and topology optimization [8,9] data-driven approaches in fluid dynamics [10,11], molecular dynamics simulation [12,13], and material properties prediction [14–17]. Atalla et al. and Levin et al. [18,19] have used neural regression for FEA model updating. More recently, DL has shown promise in solving traditional mechanics problems. Some researchers used DL for structural damage detection, a promising alternative to conventional structural health monitoring methods [20,21]. Lee et al. [22] demonstrated the efficiency and accuracy of deep learning compared to the conventional neural network using a well-known ten-bar truss problem. Do et al. [23] presented DNN to replace FEA for buckling and free vibration analyses of the functionally graded plates. Lee et al. [24] proposed method can eliminate the step of finite element analysis and accelerate topology optimization processes.

Javadi et al. [25] used a typical neural network in FEA as a surrogate for the traditional constitutive material model. They simplified the geometry into a feature vector which approaches hard to generalize complicated cases. The numerical quadrature of the element stiffness matrix in the FEA on a per-element basis was optimized by Oishi et al. using deep learning [26]. Their approach helps to accelerate the calculation of the element stiffness matrix. Convolutional Neural Network (CNN) is a kind of neural network which has shown remarkable performance on several applications related to Computer Vision and Image Processing. The significant learning ability of CNN is mainly due to several feature extraction stages that can intrinsically learn representations from the feeding data. Recently, Madani et al. [27] developed a CNN architecture for stress prediction of arterial walls in atherosclerosis. Also, Liang et al. [28] proposed a CNN model for aortic wall stress prediction. It is expected that their method will allow real-time stress analysis of human organs for a wide range of clinical applications.

Gulgec et al. [29] proposed a CNN architecture to classify simulated damaged and intact samples and localize the damage in steel gusset plates. Modares et al. [30] conducted a study on composite materials to identify the presence and type of structural damage using convolutional neural networks. Also, for detecting concrete cracks without calculating the defect features, Cha et al. [31] proposed a vision-based method based on convolutional neural networks (CNNs). Do et al. [32] proposed a method for forecasting the crack propagation in risk assessment of engineering structures based “long short-term memory” and “multilayer neural network”. Truong et al. [33] proposed a deep forward neural network method to detect the location and severity of damaged elements in the bar planar truss and bar dome-like space truss using the noisy incomplete modal dat. Lieu et al. [34] presented a deep neural network-based adaptive surrogate model for structural reliability analysis. Zhuang et al. [35] developed a technique for bending, vibration, and buckling analysis of Kirchhoff plates based on deep autoencoders. Samaniego et al. [36] proposed a deep neural network to solve boundary value problems. They used relevant examples, from computational mechanics, using DNNs to build the approximation space. A deep feed forward artificial neural network has been developed by Berg et al. [37] to approximate partial differential equations with complex geometry. Truong et al. [38] proposed a method for the safety evaluation of steel trusses using the gradient tree boosting algorithm.

An approach for predicting stress distribution on all layers of non-uniform 3D parts was presented by Khadilkar et al. [39] More recently, Nie et al. [40] developed a CNN-based method to predict the low-resolution stress field in a 2D linear cantilever beam. Jiang et al. [41] developed a conditional generative adversarial network for low-resolution von Mises stress distribution prediction in solid structures. redTo the best of our knowledge, this work is the first to use a simple physics equation to penalize the custom loss function in the ‘DL-FEA’ approach, to perform a fast and accurate prediction of high-resolution stress distributions in the specific domain of damaged steel plates.

The algorithm takes the geometry, boundary conditions, and load as input and renders the Von Mises stress distribution as an output. We modeled the steel plates as gusset plates with loading applied at different edges, different boundary conditions, and varying complex geometries. Compared with our last work [7], the dataset is initialized with 61,440 samples, representing about 40 K fewer samples. Samples initialized with varying geometries, boundary conditions, and loads are used to train and evaluate the network.

3. Custom loss function

Learning biases can be established by proper choice of loss functions, constraints, and inference algorithms that can regulate the training step of the ML model to explicitly direct convergence towards solutions that adhere to the fundamental of physics [42]. The underlying physical laws can be satisfied by using and tuning such a penalty constraint. In this paper, we consider the stress concentration factor equation to improve the prediction of stress concentration around the crack tip in steel gusset plates. It can be demonstrated from mathematical analysis and experimental results that stress distributions occur

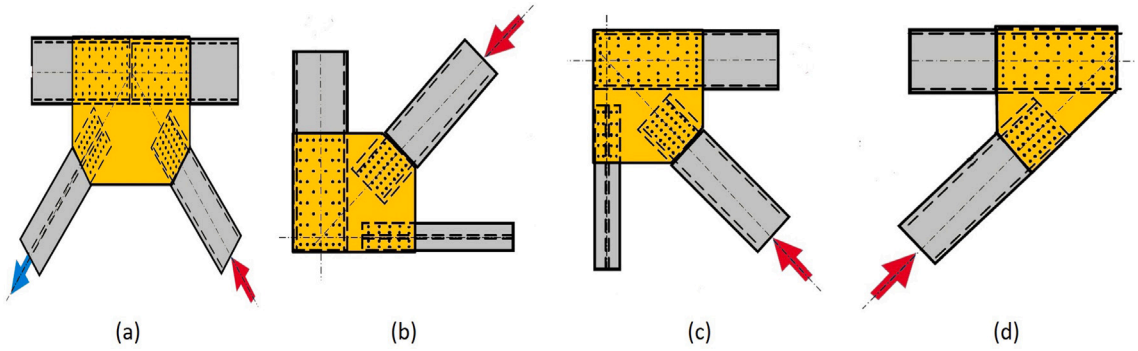

Fig. 3. Some of the most common gusset plates in practice.

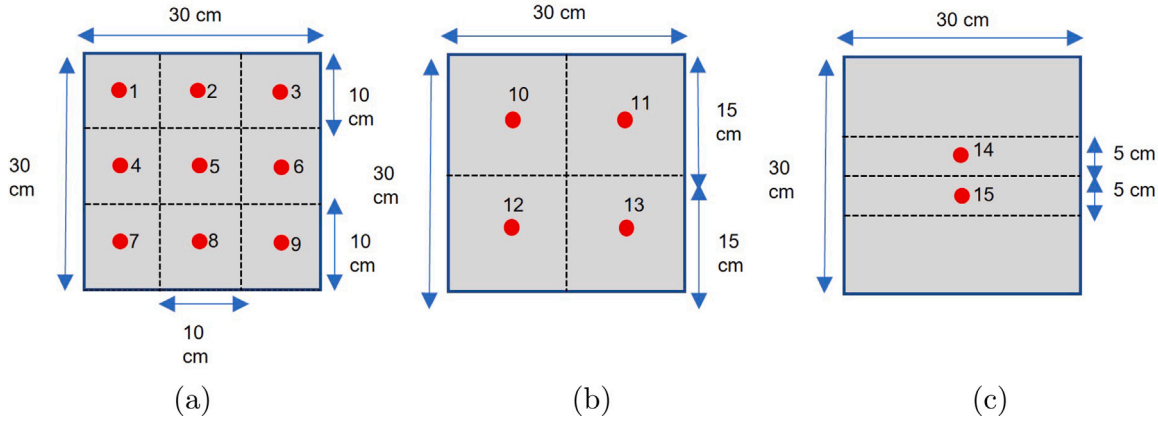

Fig. 4. Different location of damages on steel plates: (a) category 1, (b) category 2, (c) category.

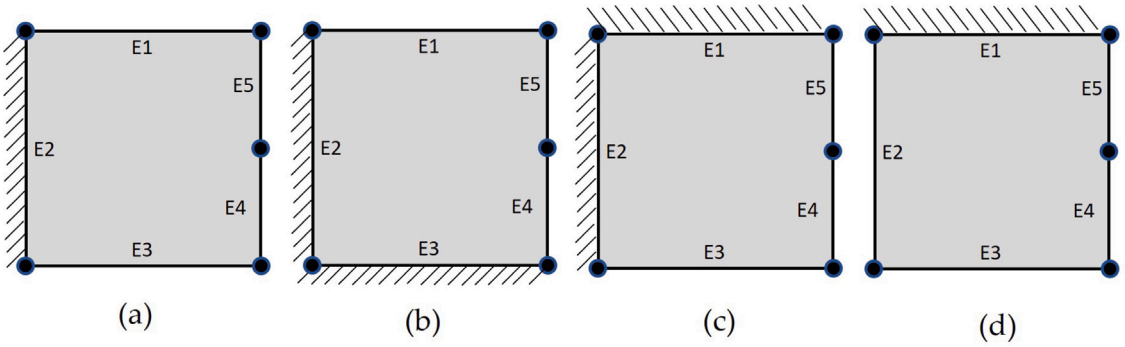

Fig. 5. Different types of boundary conditions for initializing population.

near changes in sections of a loaded structural component. It reaches greater magnitudes than the average stress in the section. It is called stress concentration when the peak stress increases near openings and other changes in the section. Eq. (1) defines stress concentration as the peak stress relative to the nominal stress that would exist if the stress distribution remained uniform [43].

$$K_t = \frac{\sigma_{max}}{\sigma_{nom}} \tag{1}$$

Where, k_t is stress concentration factor, σ_{max} and σ_{nom} are peak stress around the crack tip and nominal stress in the remainder of the section, respectively. We create binary masks to apply the stress concentration factor equation to the loss function. Fig. 1 illustrates one of the possible mask scenarios, in Fig. 1(a) crack (rectangle with gray color) is surrounded by mask 1 (white rectangle) which all pixel values of the mask are one (white), and all other pixel values are zero (black). Fig. 1(b) shows mask 2 which all the zero pixel values from mask 1 are replaced with value one, and all one pixel's values are replaced with value zero. The relation between the two masks is mask 2 = 1 - mask 1. Mask 1 represents the area where the stress concentration factor should be applied to capture the peak stress, and mask 2 represents the area with nominal stress distribution. Based on the above description, our custom loss function will be defined as below:

$$Loss = \frac{\lambda_{PHY}}{n}\sum_{n=1}^{n}(S(i) - S\hat{}(i))^2 \cdot M_1 + \frac{1}{m}\sum_{m=1}^{m}(S(i) - S\hat{}(i))^2 \cdot M_2 \tag{2}$$

Where, λ_{PHY} is stress concentration factor, n and m are the number of white pixels in mask 1 and mask 2, respectively. M_1 is mask 1 and M_2 is mask 2. S(i) is the stress value at a node 'i' computed by FEA as the ground truth and, s (i) is the corresponding predicted stress by the DL model and '·' is the symbol of the Hadamard product.

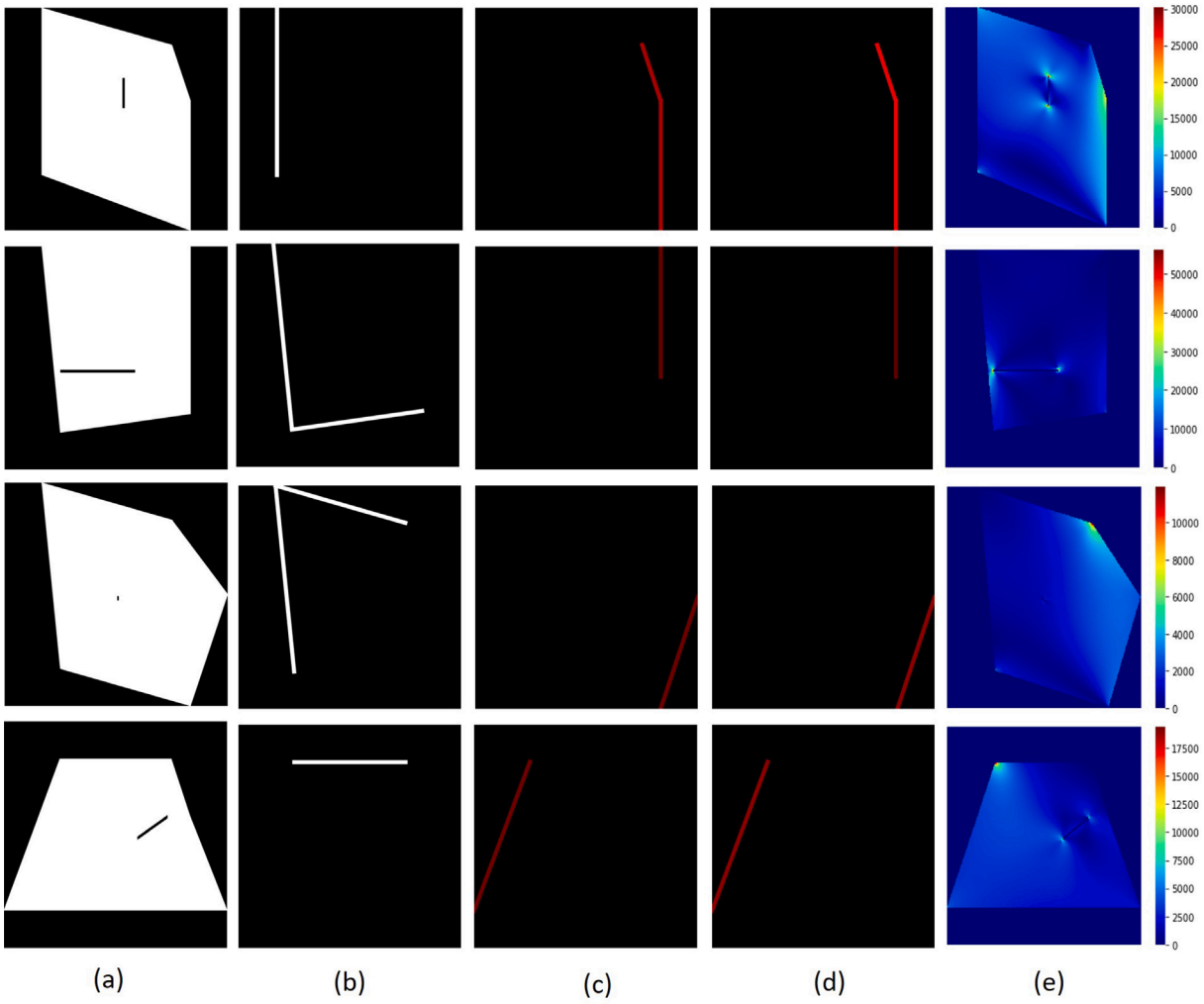

Fig. 6. Input and output representation for stress distribution prediction: (a) damaged geometry, (b) boundary condition, (c) horizontal load, (d) vertical load, (e) output.

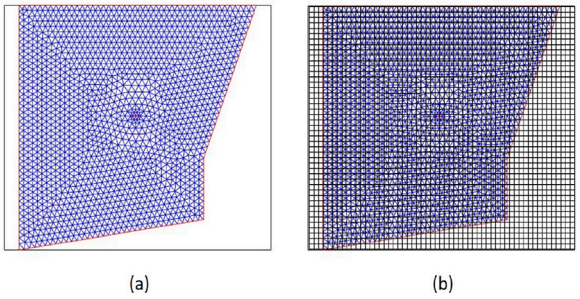

Fig. 7. A sample of mesh generation: (a) unstructured triangular mesh, (b) structured grid surface.

4. Methods

4.1. Data generation

Two-dimensional steel plate structures with five edges, E1 to E5 denoting edges 1 to 5 as shown in Fig. 2, are considered homogeneous and isotropic linear elastic material. Various geometries are generated by changing the position of each node in horizontal and vertical directions, as shown in Fig. 2, which led to 1024 unique pentagons. The material properties remain unchanged and isotropic for all samples. 1024 crack scenarios with various widths, lengths, angles, and locations were also created on the steel plates (Fig. 2). Since the number of geometries is the same as the number of crack scenarios, each geometry has a unique crack.

The 2D steel plates approach the geometry of gusset plates. The boundary conditions and loading angles are considered to simulate similar conditions in common gusset plate structures under external loading. Some most common gusset plates in practice are shown in Fig. 3. Adding different loading and boundary conditions extended the population into 61,440 unique samples. All input variables used to initialize the population are shown in Table 1. Gusset plates are used for connecting beams and columns to braces in steel structures. The behavior and analysis of these components are particularly important since various reports have observed failures of gusset plates subject to lateral loads [44–47].

For crack initiation, we divided the steel plate into 15 different regions to create cracks with different lengths and ensure that the crack length would not violate the edges of the steel plate. Fig. 4 shows all damage locations in 3 categories. Every red point represents the center of single damage in the steel plate. Categories 1 to 3 each have 9, 4, and 2 subcategories, respectively. Each plate has just one crack, and other red points just represent the location of the cracks. Details of crack initiation are shown in Table 2.

The distributed static loads, applied to the gusset plates in this study, ranged from 1 to 5 kN with intervals of 1 kN. Moreover, loads were applied with 3 different angles, including $\frac{\pi}{6}$, $\frac{\pi}{4}$ and $\frac{\pi}{3}$ on either

Table 1
Input variables.

Geometry	Boundary conditions	Load position	Load angle (degree)	Load Magnitude (kN)
pentagon	E2	E4E5	30,45,60	1,2,3,4,5
pentagon	E2E3	E5	30,45,60	1,2,3,4,5
pentagon	E1E2	E4	30,45,60	1,2,3,4,5
pentagon	E1	E2	30,45,60	1,2,3,4,5

Table 2
Detail of damages in steel plates.

Crack number	Width mm	Length of category1 mm	Length of category2 mm	Length of category3 mm	Angle degree
1	1	10,40,80	120	160,200	0,30,45,90
2	2	10,40,80	120	160,200	0,30,45,90
3	3	10,40,80	120	160,200	0,30,45,90
4	4	10,40,80	120	160,200	0,30,45,90
5	5	10,40,80	120	160,200	0,30,45,90
6	6	10,40,80	120	160,200	0,30,45,90
7	7	10,40,80	120	160,200	0,30,45,90

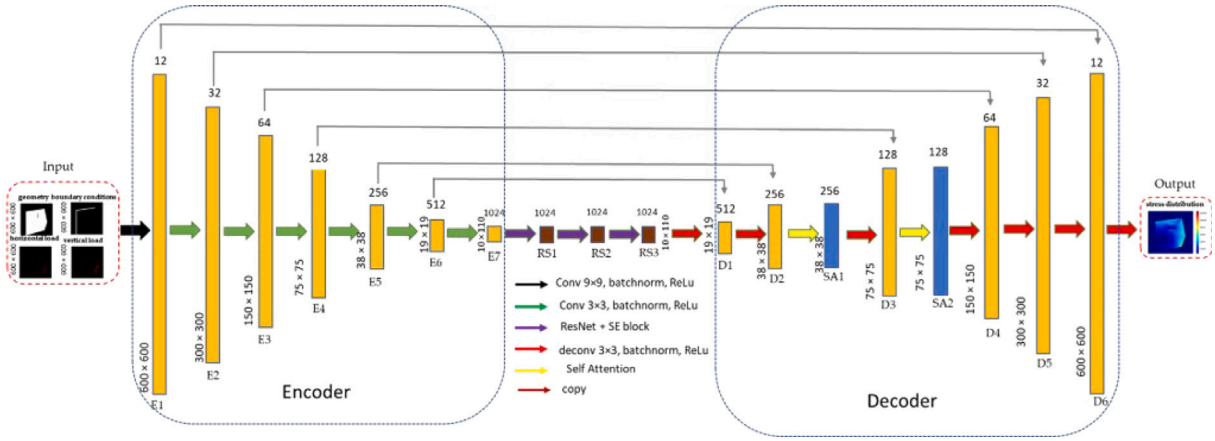

Fig. 8. Proposed CNN architecture.

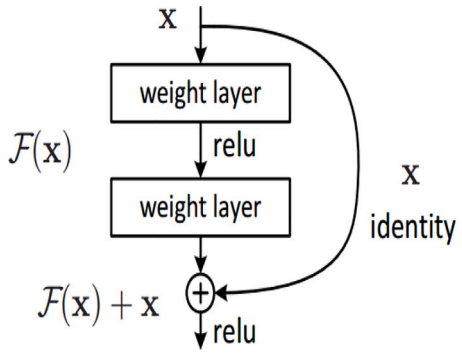

Fig. 9. The building block of residual learning [48].

one or two edges of the plate. The load is decomposed to its horizontal and vertical direction components. Also, four types of boundary conditions are considered, as shown in Fig. 5, similar to real gusset plates' boundary conditions. All the translational and rotational displacements were fixed at the boundary conditions. The minimum and maximum range for the width and height of the plate are from 30 cm to 60 cm.

4.2. Input data

The geometry is encoded into a 600 × 600 matrix as a single channel binary image. 0 (black) and 1 (white) denote the outside and inside of the geometry, as shown in Fig. 6(a). The boundary condition is also represented by another 600 × 600 pixel binary image, where the constrained edges are defined by 1 (white) (Fig. 6(b)). Moreover, each horizontal and vertical component of the load is encoded as one 600 × 600-pixel single-channel colored image, as shown in Fig. 6(c) and (d). Each row of Fig. 6 represents one of the simulated boundary conditions and its load positions as described in Table 1. The magnitude of the horizontal and vertical components of the loads, after decomposition, varies between 0.5 kN and 4.33 kN. These loads are normalized between (100,0,0) and (255,0,0) as RGB colors to create a color image where the colored part represents the location and magnitude of the load (Fig. 6(c) and (d)).

4.3. Output data

FEA was performed using the Partial Differential Equation (PDE) solver in the MATLAB toolbox to obtain the stress distributions of each sample. We did not use other FE approaches such as a Carrera unified formulation [51] due to computational cost. Carrera Unified Formulation allows FE matrices and vectors to be derived in terms of fundamental nuclei. The MATLAB PDE toolbox mesh generator only generates unstructured triangulated meshes, which are not compatible with CNN. The minimum and maximum triangulated mesh size is 5 mm and 10 mm; respectively. Since each element should be represented by one pixel in an image, we develop a 600 × 600 grid surface equal to the dimensions of the largest possible geometry. Fig. 7(a) and (b) show the unstructured mesh and 600 by 600 grid surface on top of the one random sample, respectively. The stress values are then interpolated between the triangular elements and grids to determine a

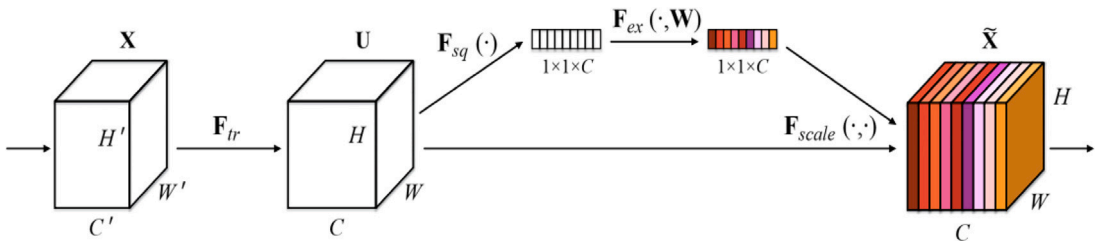

Fig. 10. The building block of Squeeze-and-Excitation [49].

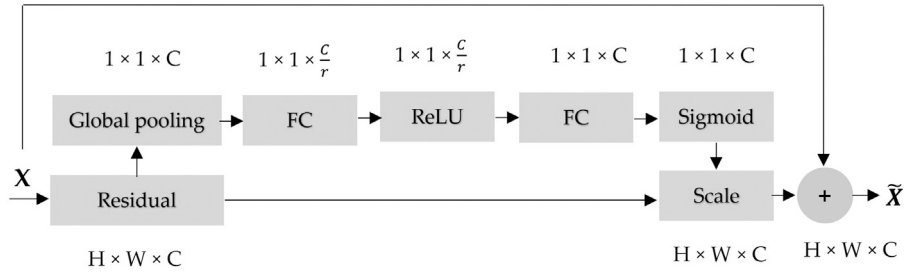

Fig. 11. SE-ResNet module.

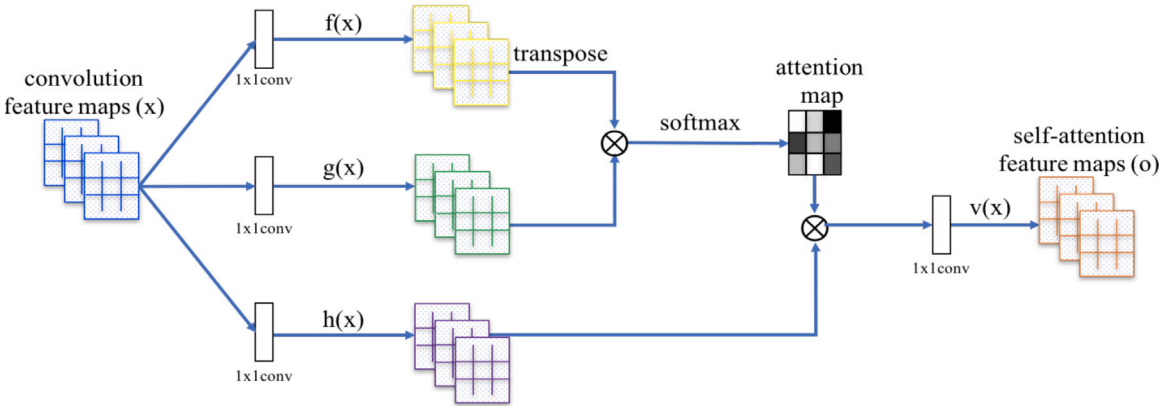

Fig. 12. The building block of self-attention module for the SAGAN [50].

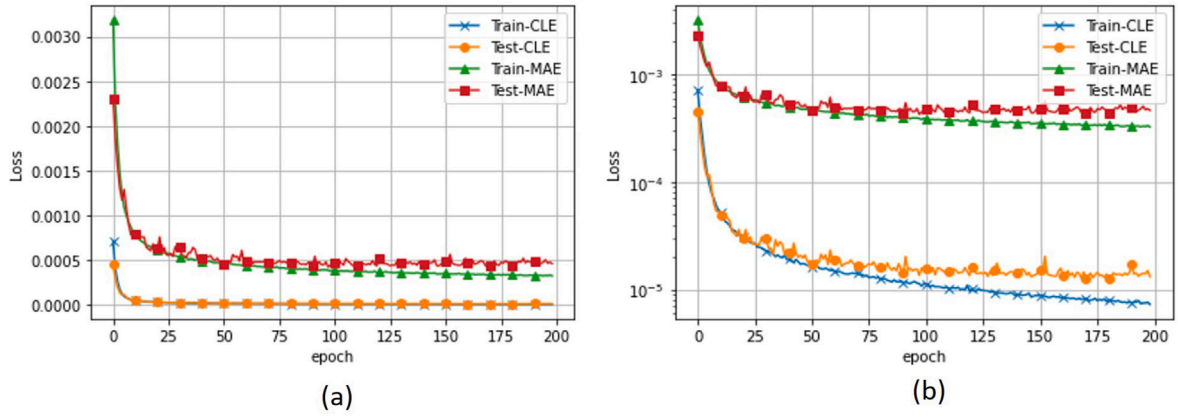

Fig. 13. CLE and MAE curves on training and testing data with two scales: (a) linear scale, (b) logarithmic scale.

stress distribution compatible with our CNN network. The stress values of all the elements outside of the material geometry are assigned to zero, as shown in Fig. 6(e). The dimensions of the largest sample are 600 × 600 mm, and the smallest are 300 × 300 mm. Therefore, the dimension of each element is 1 × 1 mm, which means that each image has 360000 pixels. All the cracks are initialized in the smallest dimension (300 × 300 mm) to keep all lengths of cracks inside the geometry. This high-resolution dataset led to achieving significant accuracy. The maximum and minimum von Mises stress values for elements among the entire dataset are 362,687 MPa and −138.35 MPa, respectively.

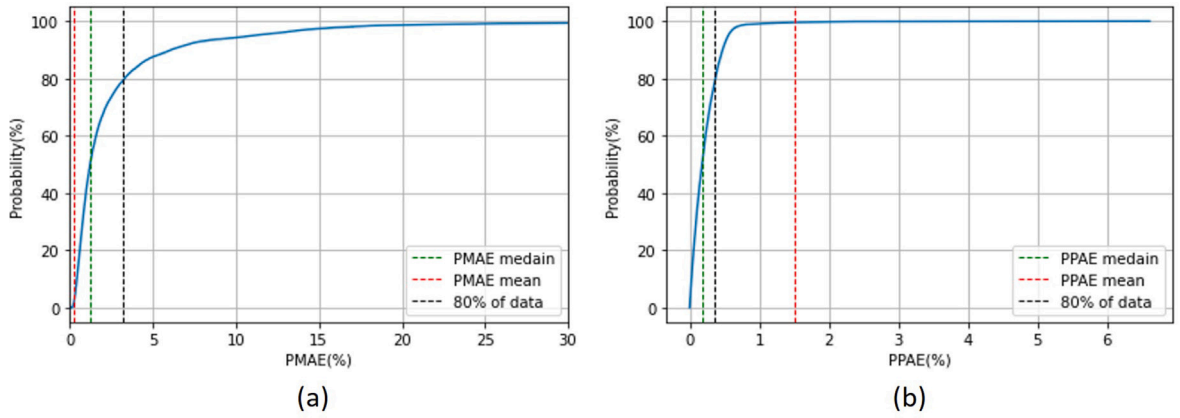

Fig. 14. Cumulative distribution of PMAE and PPAE: (a) PMAE of samples less than mean, median and, 80% of data on the test dataset, (b) PPAE of samples less than mean, median and, 80% of data on the test dataset.

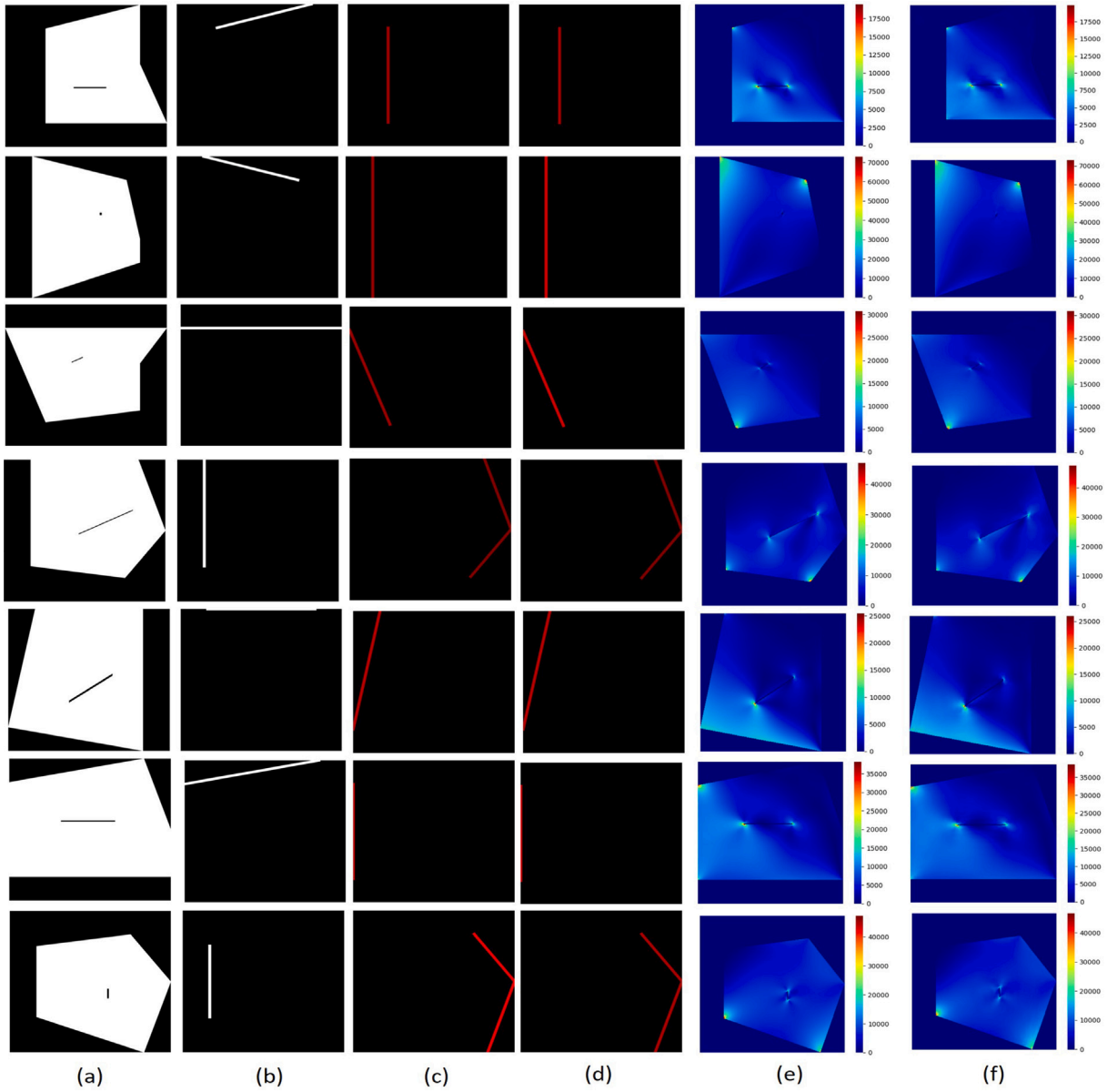

Fig. 15. Predicted stress distribution and corresponding inputs with different loads and boundary conditions scenarios. Columns (a) to (d) represent damaged geometry, boundary conditions, load in the horizontal and vertical direction, respectively. Columns (e) and (f) represent ground truth and predicted stress distribution, respectively. (Units = mm-MPa-N).

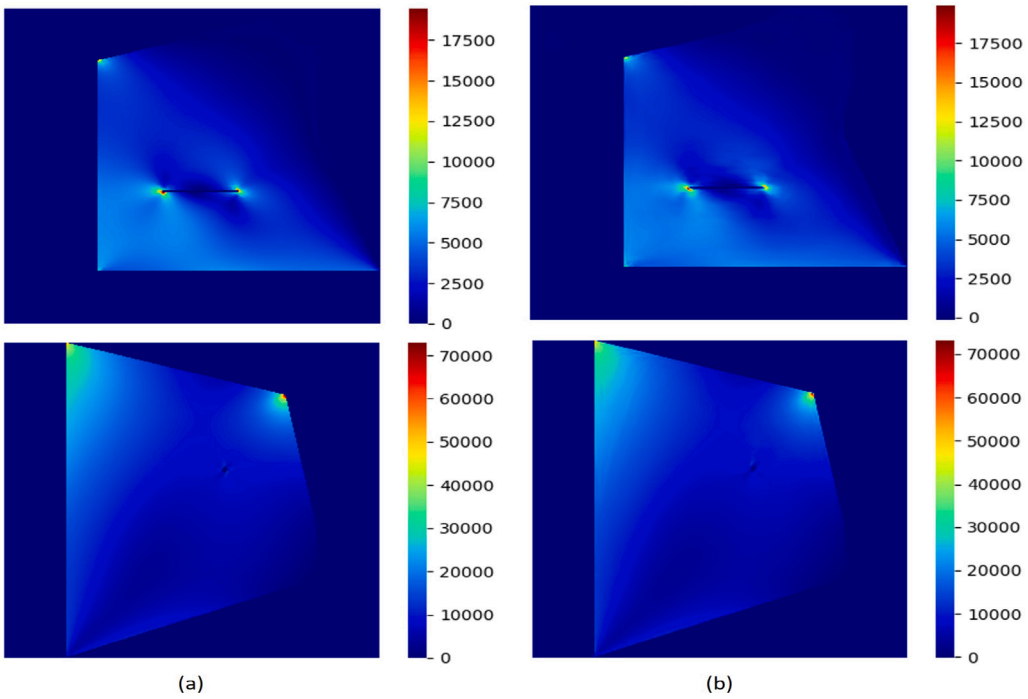

Fig. 16. Larger representative from Fig. 15. (a) Ground truth (b) predicted. (Units = mm-MPa-N).

Table 3
Size of network layers.

Downsampling layers	E1	E2	E3	E4	E5	E6	E7	
Height× Width × Channel	600 × 600 × 12	600 × 600 × 32	600 × 600 × 64	600 × 600 × 128	600 × 600 × 256	600 × 600 × 512	600 × 600 × 1024	
SE-ResNet Layers	RS1	RS2	RS3					
Height× Width × Channel	10 × 10 × 1024	10 × 10 × 1024	10 × 10 × 1024					
Upsampling layers	D1	D2	SA1	D3	SA2	D4	D5	D6
Height× Width × Channel	19 × 19 × 512	38 × 38 × 256	38 × 38 × 256	75 × 758 × 128	75 × 758 × 128	150 × 150 × 64	300 × 300 × 32	600 × 600 × 12

We normalized all the output data between 0 and 1 to ensure faster convergence and encoded it to 600 × 600 matrices.

5. CNN architecture

The CNN can be built in different ways using a sequence of convolutional layers. The convolutional layers learn to encode the input in a set of simple signals and then reconstruct the input [52]. Our CNN architecture consists of 4 types of layers: The first stage is downsampling layers which consist of seven convolutional layers (E1, E2, E3, E4, E5, E6, E7), the second stage are 3 layers (RS1, RS2, and RS3) of Squeeze-Excitation and Residual blocks (SE-ResNet). The last stage is upsampling layers which consist of six deconvolutional layers (D1, D2, D3, D4, D5, D6) and 2 self-attention layers (SA1, SA2), as illustrated in Fig. 8. The size of layers can be also seen in Table 3.

5.1. Residuals blocks

We used residual blocks to address the vanishing gradient problem. In addition, SE blocks are computationally lightweight and result in only very small increases in model complexity. As illustrated in Fig. 9, the formulation of F(x)+x can be realized by feedforward neural networks with shortcut connections. The shortcut connection simply performs identity mapping, and its output is added to the output of the stacked layers [48].

5.2. Squeeze-and-excitation blocks

As depicted in Fig. 10, Squeeze-and-Excitation blocks improve the representational capacity of the network, enabling dynamic channel-wise feature recalibration. A SE-block can be implemented with five steps. First, we feed the input x as a convolutional block and the current number of channels to the SE function, where F_{tr} in Fig. 10 is the convolutional operator for the transformation of X to U. Then, at the second phase, Each channel is squeezed into a single numeric value by using average pooling. Additionally, in the third phase, a fully connected layer is followed by a ReLu function, which applies a nonlinearity and reduces the output channel complexity. Then at the fourth phase, SE blocks can be used directly with residual networks. Fig. 11 depicts a SE-ResNet module which the SE block transformation. F_{tr} is regarded as the non-identity branch of a residual module. Before summation of the identity branch, both squeeze and excitation act. Using both SE and ResNet in the network outperforms using ResNet [49].

5.3. Self-attention blocks

In DL, the attention mechanism is inspired by human vision. Our brain transmits a signal via neurons after we receive visual information from the outside. Humans benefit from this process as it helps them focus on the right areas, and it reduces the weight of unrelated areas in their attention. As part of the feature extraction process of the input image, attention increases the weight of the area of interest and reduces the weight of unrelated regions. In the current paper, we use Self-Attention GAN (SAGAN) [50]to improve the prediction's results. Convolution processes information in a local neighborhood; therefore,

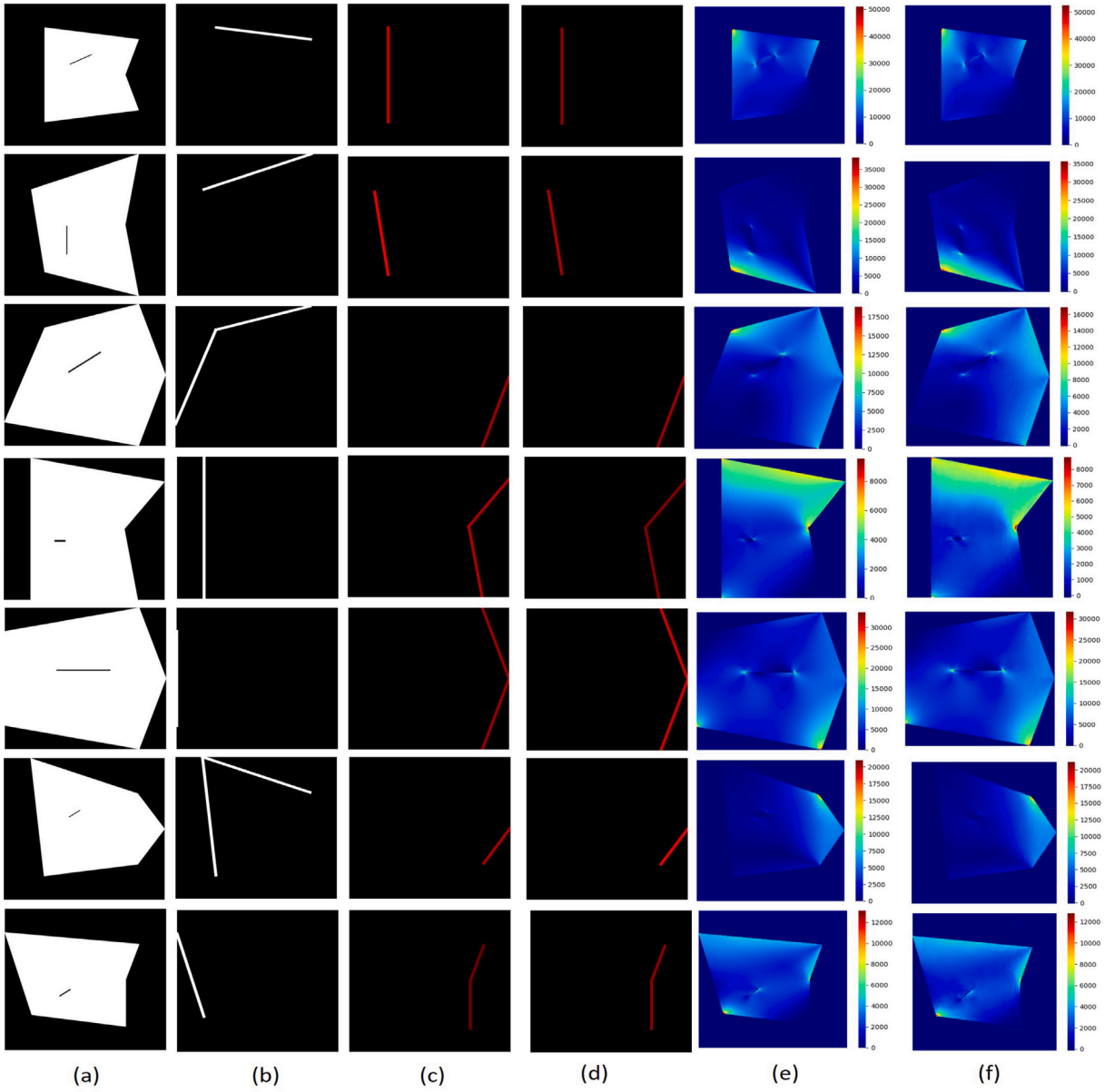

Fig. 17. Inaccurate predicted stress distribution and corresponding inputs with different loads and boundary conditions scenarios. Columns (a) to (d) represent geometry, boundary conditions, load in the horizontal and vertical direction, respectively. Columns (e) and (f) represent ground truth and predicted stress distribution, respectively. (Units = mm-MPa-N).

using a single convolutional layer is computationally inefficient for modeling long-range dependency in images. SAGAN helps efficiently model relationships between widely separated spatial regions, even areas far apart; it can simply capture global dependencies.

In the self-attentions mechanism, the convolutional image feature maps are broadening into three copies, corresponding to the key, value, and query concepts in the transformer [53]. key, value, and query are Key: $f(x) = W_{fx}$, Query: $g(x) = W_{gx}$, and Value: $h(x) = W_{hx}$. The image features from the previously hidden layer are first transformed into two feature spaces f(x) and g(x), to calculate the attention and then apply the dot-product attention to output the self-attention feature maps using Eqs. (3) and (4). The entire process of the self-attention mechanism in SAGAN is depicted in Fig. 12.

$$a_{i,j} = Softmax(f(x_i)^T g(x_i)) \tag{3}$$

$$o_j = w_v(\sum_{i=1}^{n} \alpha_{i,j} h(x_i)) \tag{4}$$

6. Loss function and performance metrics

We used the custom loss function for the training loss as defined in Eq. (2). Therefore, its error is defined as CLE (Custom Loss Error). We also used MAE (Mean Absolute Error), PMAE (Percentage Mean Absolute Error), PAE (Peak Absolute Error), and PPAE (Percentage Peak Absolute Error) to evaluate the overall quality of predicted stress distribution. These metrics are defined in Eqs. (5) (6), (7), (8), respectively.

$$MSE = \frac{1}{n} |\sum_{i=1}^{n} (S(i) - S^{\wedge}(i))^2| \tag{5}$$

Where $S(i)$ is the stress value at a node 'i' computed by FEA as the ground truth and, $S^{\wedge}(i)$ is the corresponding predicted stress by the DL model, and n is the total number of elements at each sample which is 360000 in our work. Symbol || denotes the absolute value. Our model's prediction and ground truth are displayed as 600 × 600 resolution images.

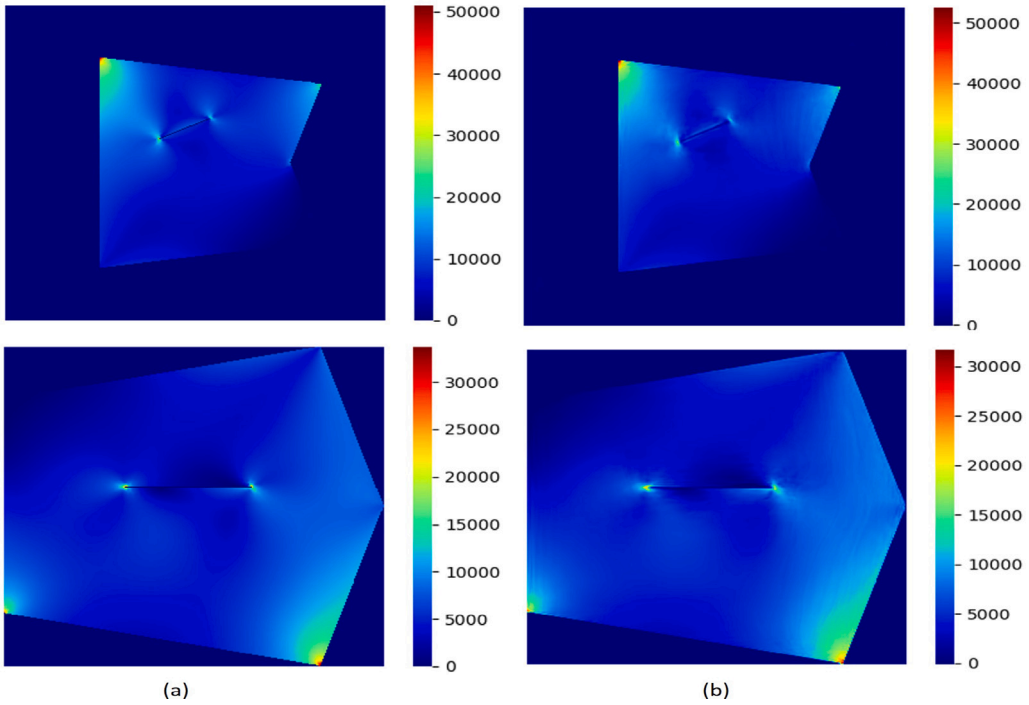

Fig. 18. Larger representative from Fig. 17. (a) Ground truth (b) predicted. (Units = mm-MPa-N).

The percentage mean absolute error is defined as:

$$PMAE = \frac{MAE}{max[S(i)] - min[S(i)]} \times 100 \tag{6}$$

Where max $[S(i)]$ is the maximum value in a set of ground truth stress values and min $[S(i)]$ is the minimum value.

PAE and PPAE measure the accuracy of maximum stress which are one of the main important critical load values in the predicted stress distribution. The importance of maximum stress matters in design phase, since maximum stress should be less than yield strength to avoid permanent deformation. PAE and PPAE are defined as:

$$PAE = [S(i)] - [S\hat{}(i)] \tag{7}$$

$$PPAE = \frac{PAE}{[S(i)]} \times 100 \tag{8}$$

7. Implementation details

All codes are written in PyTorch Lightning and run on two NVIDIA TITAN RTX 24G GPUs. AdamW optimizer [54] was used to speed up the convergence of our models using a learning rate of 1e−5. The batch size is set to 8, leading to the best accuracy compared to other batch sizes. The value of stress concentration factor, which is applied as a λ_{PHY} in the custom loss function is 15, leading to the best results compared to the other values.

8. Results and discussions

We train and evaluate our model on custom loss using the entire dataset for 200 epochs, the other metrics are plotted as independent metrics. The training data size is 49,152, and the separate testing data size is 12,288 which is randomly divided with a train/test ratio of 80%–20%. Fig. 13 shows Custom Loss Function (CLE) and MAE losses as a function of epochs. Fig. 13(a) is in linear scales, and Fig. 13(b) is in logarithmic scales. Fig. 13(a) shows that the curves of both CLE and MAE rapidly declined after a few epochs. However, Fig. 13(b) gives a more precise representation of the model's behavior. From Fig. 13(b), it can be seen that CLE is less than MAE, which is due to penalization of the loss with the stress concentration factor.

Table 4
Error metrics at epoch 181 (Units: mm-MPa-N).

Metrics	CLE	MAE	PMAE (%)	PAE	PPAE (%)
Testing	4.5	22.01	0.22	10.8	1.5

We saved the best checkpoint during training, epoch 181, and all error metrics are based on this checkpoint. The evaluation results of the network are shown in Table 4. As it can be seen, PMAE and PPAE for the testing dataset are just 0.22% and 1.50%, respectively. We consider these results satisfactory for stress distribution predictions of damaged structural components, specifically the PPAE, which is the most critical load value for stress distribution and stress concentration in engineering do main applications.

Fig. 14 illustrates the cumulative distribution of PMAE and PPAE in the test dataset. Fig. 14(a) shows that the probability of mean in PMAE is about 2%, which means that about 2% of predicted samples have a PMAE of less than 0.22; however, 80% of predicted samples have PMAE less than 3, and 50% of predicted samples have a PMAE of less than 1.21, which is the median. Fig. 14(b) shows that about 99% of predicted samples have a PPAE of less than 1.5, 80% of predicted samples have PMAE less than 0.35 and, 50% of the predicted samples have a PPAE of less than 0.17.

The prediction results of some randomly selected samples from the test dataset are visualized in Fig. 15. Each row represents a sample. Columns (a) to (d) represent geometry, boundary conditions, and load in horizontal and vertical directions, respectively. Columns (e) and (f) represent the ground truth and predicted stress distributions, respectively. Comparing columns (e) and (f) shows that predicted stress distributions are pretty similar to the ground truths. Fig. 15 also demonstrates that the network can localize and quantify different sizes of damages, even tiny cracks. The second row of Fig. 15 is an example of rectangle damage with a size of 10 × 1 mm, which shows proper damage localization and stress distribution prediction of the remainder of the section.

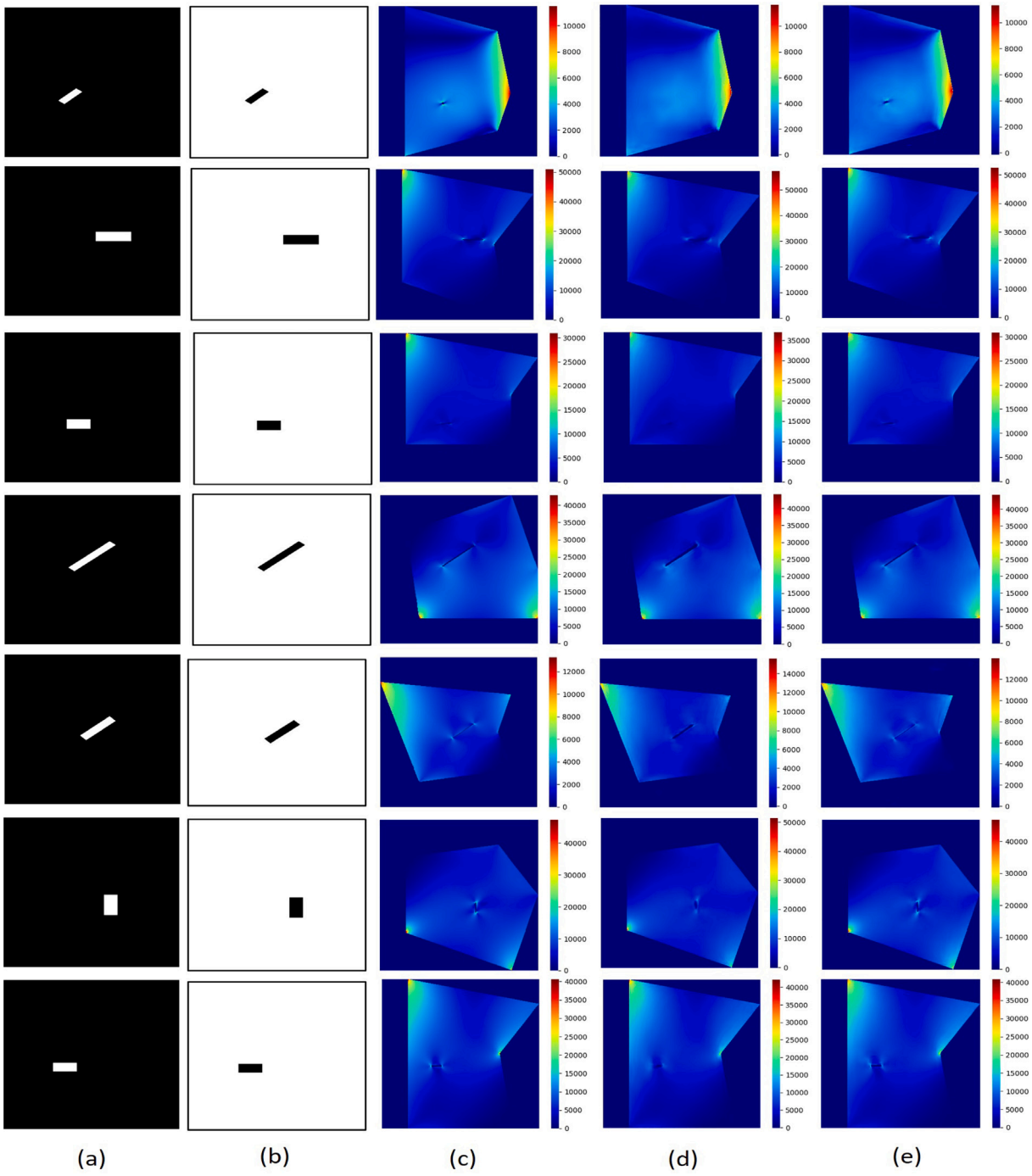

Fig. 19. Comparison of predicted stress distribution in MSE and custom loss model and their corresponding masks. Columns (a) and (b) represent the mask for the damaged area and the remainder of the section, respectively. Columns (c) to (e) represent ground truth, predicted stress distribution in the MSE and custom loss models, respectively. (Units = mm-MPa-N).

Fig. 16 shows two cases of Fig. 15 with larger scales to show details of ground truth and predicted stress distribution. Moreover, comparing the last two columns of Fig. 15 shows the efficacy of our novel custom loss function, which can accurately capture stress concentration around crack tips. This means the learned algorithm can capture the underlying knowledge of physics behind the stress concentration.

Some inaccurate predictions are also shown in Fig. 17. As it can be seen, these predictions can still capture damage locations and stress concentration around crack tips; however, mean and peak stress distribution in some parts of the ground truth and predictions slightly vary. Fig. 18 shows two cases of Fig. 17 with larger scales to show details of ground truth and predicted stress distribution.

9. Study on the effect of the custom loss function

We have also trained our model using torch.nn.MSELoss function compares its results with our custom loss model to evaluate how efficient our proposed custom loss function is. We investigated the performance of custom loss model with different stress concentration factor and compared with MSE model. Table 5 demonstrates that the performance of custom loss model with stress concentration factor equal to 1 is almost same as MSE model, which is expected based on Eq. (2).

However, with applying higher and lower values than 15 the PPAE increased in the custom loss model. It seems lower stress concentration

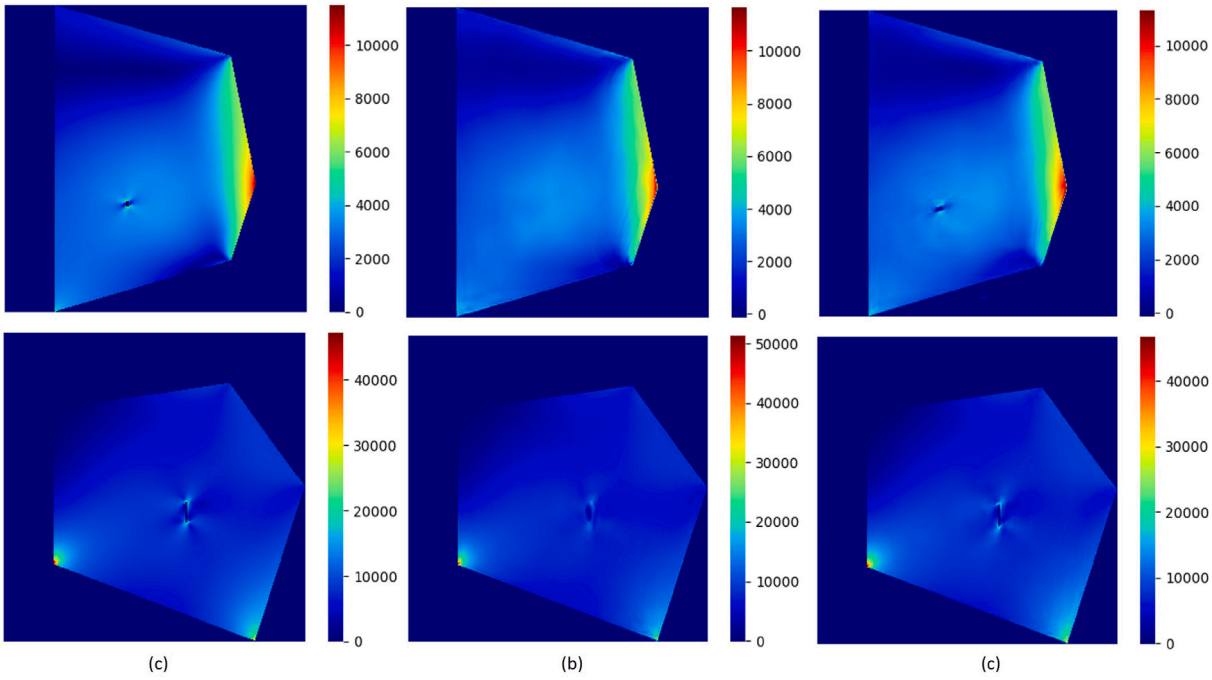

Fig. 20. Larger representative from Fig. 19. (a) ground truth (b) MSE loss model (c) custom loss model. (Units = mm-MPa-N).

Table 5
Error metrics with different stress concentration factors (Units: mm-MPa-N).

Model	Stress concentration factor	PMAE (%)	PPAE (%)
Custom loss	1	0.05	2.52
Custom loss	8	0.12	2
Custom loss	15	0.22	1.5
Custom loss	20	0.27	2.3
Custom loss	30	0.48	3.7
MSE loss	None	0.053	2.5

Table 6
Error metrics in MSE and custom loss model (Units: mm-MPa-N).

Metrics	CLE	MAE	PMAE (%)	PAE	PPAE (%)
Custom loss	4.5	22.01	0.22	10.8	1.5
MSE		5.07	0.053	109.03	2.5

factors do not trigger the pixel values within the damaged area and higher stress concentration factors are over triggering the pixel values within the damaged area. Therefor, we trained our custom loss model with stress factor 15 to obtain the best results.

The error metric in the MSE loss model and the custom loss model with stress factor 15 are presented in Table 6. As it can be seen, the custom loss model's performance is slightly better than MSE model in terms of PPAE, which is expected since we penalize the damaged areas 15 times more than the remainder of the section. However, MSE model shows better performance in terms of MAE metric than custom loss model. This means custom loss model is better in damage localization and capturing stress concentration around the crack tips and MSE model is better in general stress prediction. Although PPAE in the custom loss model is 1% less than the MSE model, the custom loss has a significant advantage in damage localization and capturing stress concentration around crack tips.

Fig. 19 compares predicted stress distribution between MSE and custom loss model and the masks used for custom loss function. In Fig. 19 Columns (a) represents the mask used for penalizing the damaged area, and column (b) represents the mask used for the remainder of the section, which has no penalization. Columns (c), (d), and (e) represent the ground truth, MSE model prediction and, custom loss model prediction. As it can be seen in the first row of Fig. 19, the MSE model cannot localize the crack; however, the custom loss model has completely localized the tiny crack and its stress concentration around the crack tip. Fig. 19 also shows that the MSE model can localize larger cracks but still is not as good as the custom loss model in capturing stress concentration around crack tips. Fig. 20 shows two cases of Fig. 19 with larger scales to show details of ground truth and predicted stress distribution with MSE loss model and custom loss model.

10. Conclusion

This work develops a convolutional neural network (CNN) augmented with the custom loss function which is inspired from stress concentration physics equation to predict high-resolution von Mises stress distribution in the specific domain of damaged steel plates. The proposed network learns to predict the stress distribution given the damaged geometries, load, and boundary conditions as input and high-resolution stress contours as the output. The dataset is composed of 61,440 unique and complex cases of various geometries, boundary conditions, and loads. The PDE toolbox of MATLAB was used to generate the output data for training as FEA samples. We also build a CNN model using torch.nn.MSELoss function to see how much our proposed custom loss function is efficient. The CNN network achieves high accuracy in both custom loss and MSE models, under multiple metrics, in the evaluations of stress distribution datasets. The custom loss model outperforms the MSE model in terms of peak stress value predictions, in addition to accurately localizing damages and capturing stress concentration around crack tips, which is not possible with other ML methods. The custom loss trained DL model which trained with 49152 FEA samples can be used for future predictions of stress distributions of damaged steel plates of 12888 FEA samples. The custom loss trained DL model has a mean absolute error of 0.22% and a maximum stress error of 1.5% in the Von Mises stress distribution.

Declaration of competing interest

One or more of the authors of this paper have disclosed potential or pertinent conflicts of interest, which may include receipt of payment,

either direct or indirect, institutional support, or association with an entity in the biomedical field which may be perceived to have potential conflict of interest with this work. For full disclosure statements refer to https://doi.org/10.1016/j.advengsoft.2022.103240. Nizar Lajnef reports financial support was provided by National Science Foundation.

Acknowledgment

This research was funded in part by National Science Foundation, United States of America, grant number CNS 1645783. The corresponding author states that there is no conflict of interest on behalf of all authors.

References

[1] Liu Yongming, Stratman Brant, Mahadevan Sankaran. Fatigue crack initiation life prediction of railroad wheels. Int J Fatigue 2006;28(7):747–56.
[2] Dutta NC. Geopressure prediction using seismic data: Current status and the road ahead. Geophysics 2002;67(6):2012–41.
[3] Maqsood Imran, Khan Muhammad Riaz, Abraham Ajith. An ensemble of neural networks for weather forecasting. Neural Comput Appl 2004;13(2):112–22.
[4] Karpatne Anuj, Atluri Gowtham, Faghmous James H, Steinbach Michael, Banerjee Arindam, Ganguly Auroop, et al. Theory-guided data science: A new paradigm for scientific discovery from data. IEEE Trans Knowl Data Eng 2017;29(10):2318–31.
[5] LeCun Yann, Bengio Yoshua, Hinton Geoffrey. Deep learning. Nature 2015;521(7553):436–44.
[6] Schmidhuber Jürgen. Deep learning in neural networks: An overview. Neural Netw 2015;61:85–117.
[7] Bolandi Hamed, Li Xuyang, Salem Talal, Boddeti Vishnu, Lajnef Nizar. Bridging finite element and deep learning: B high-resolution stress distribution prediction in structural components. Front. Struct. Civ. Eng. 2022;17.
[8] Umetani Nobuyuki. Exploring generative 3D shapes using autoencoder networks. In: SIGGRAPH Asia 2017 technical briefs. 2017, p. 1–4.
[9] Yu Yonggyun, Hur Taeil, Jung Jaeho, Jang In Gwun. Deep learning for determining a near-optimal topological design without any iteration. Struct Multidiscip Optim 2019;59(3):787–99.
[10] Farimani Amir Barati, Gomes Joseph, Pande Vijay S. Deep learning the physics of transport phenomena. 2017, arXiv preprint arXiv:1709.02432.
[11] Kim Byungsoo, Azevedo Vinicius C, Thuerey Nils, Kim Theodore, Gross Markus, Solenthaler Barbara. Deep fluids: A generative network for parameterized fluid simulations. In: Comput Graph Forum. 38, (2):Wiley Online Library; 2019, p. 59–70.
[12] Goh Garrett B, Hodas Nathan O, Vishnu Abhinav. Deep learning for computational chemistry. J Comput Chem 2017;38(16):1291–307.
[13] Mardt Andreas, Pasquali Luca, Wu Hao, Noé Frank. VAMPnets for deep learning of molecular kinetics. Nature Commun 2018;9(1):1–11.
[14] Mohammadi Bayazidi Alireza, Wang Gai-Ge, Bolandi Hamed, Alavi Amir H, Gandomi Amir H. Multigene genetic programming for estimation of elastic modulus of concrete. Math Probl Eng 2014;2014.
[15] Sarveghadi Masoud, Gandomi Amir H, Bolandi Hamed, Alavi Amir H. Development of prediction models for shear strength of SFRCB using a machine learning approach. Neural Comput Appl 2019;31(7):2085–94.
[16] Mousavi Seyyed Mohammad, Aminian Pejman, Gandomi Amir Hossein, Alavi Amir Hossein, Bolandi Hamed. A new predictive model for compressive strength of HPC using gene expression programming. Adv Eng Softw 2012;45(1):105–14.
[17] Bolandi Hamed, Banzhaf Wolfgang, Lajnef Nizar, Barri Kaveh, Alavi Amir H. An intelligent model for the prediction of bond strength of FRP bars in concrete: A soft computing approach. Technologies 2019;7(2):42.
[18] Atalla Mauro J, Inman Daniel J. On model updating using neural networks. Mech Syst Signal Process 1998;12(1):135–61.
[19] Levin Robert Ian, Lieven NAJ. Dynamic finite element model updating using neural networks. J Sound Vib 1998;210(5):593–607.
[20] Fan Zhun, Wu Yuming, Lu Jiewei, Li Wenji. Automatic pavement crack detection based on structured prediction with the convolutional neural network. 2018, arXiv preprint arXiv:1802.02208.
[21] Dung Cao Vu, et al. Autonomous concrete crack detection using deep fully convolutional neural network. Autom Constr 2019;99:52–8.
[22] Lee Seunghye, Ha Jingwan, Zokhirova Mehriniso, Moon Hyeonjoon, Lee Jaehong. Background information of deep learning for structural engineering. Arch Comput Methods Eng 2018;25(1):121–9.
[23] Do Dieu TT, Lee Dongkyu, Lee Jaehong. Material optimization of functionally graded plates using deep neural network and modified symbiotic organisms search for eigenvalue problems. Composites B 2019;159:300–26.
[24] Lee Seunghye, Kim Hyunjoo, Lieu Qui X, Lee Jaehong. CNN-based image recognition for topology optimization. Knowl-Based Syst 2020;198:105887.
[25] Javadi AA, Tan TP, Zhang M. Neural network for constitutive modelling in finite element analysis. Comput Assist Mech Eng Sci 2003;10(4):523–30.
[26] Oishi Atsuya, Yagawa Genki. Computational mechanics enhanced by deep learning. Comput Methods Appl Mech Engrg 2017;327:327–51.
[27] Madani Ali, Bakhaty Ahmed, Kim Jiwon, Mubarak Yara, Mofrad Mohammad RK. Bridging finite element and machine learning modeling: stress prediction of arterial walls in atherosclerosis. J Biomech Eng 2019;141(8).
[28] Liang Liang, Liu Minliang, Martin Caitlin, Sun Wei. A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis. J R Soc Interface 2018;15(138):20170844.
[29] Gulgec Nur Sila, Takáč Martin, Pakzad Shamim N. Convolutional neural network approach for robust structural damage detection and localization. J Comput Civ Eng 2019;33(3):04019005.
[30] Modarres Ceena, Astorga Nicolas, Droguett Enrique Lopez, Meruane Viviana. Convolutional neural networks for automated damage recognition and damage type identification. Struct Control Health Monit 2018;25(10):e2230.
[31] Cha Young-Jin, Choi Wooram, Büyüköztürk Oral. Deep learning-based crack damage detection using convolutional neural networks. Comput-Aided Civ Infrastruct Eng 2017;32(5):361–78.
[32] Do Dieu TT, Lee Jaehong, Nguyen-Xuan H. Fast evaluation of crack growth path using time series forecasting. Eng Fract Mech 2019;218:106567.
[33] Truong Tam T, Dinh-Cong D, Lee Jaehong, Nguyen-Thoi T. An effective deep feedforward neural networks (DFNN) method for damage identification of truss structures using noisy incomplete modal data. J Build Eng 2020;30:101244.
[34] Lieu Qui X, Nguyen Khoa T, Dang Khanh D, Lee Seunghye, Kang Joowon, Lee Jaehong. An adaptive surrogate model to structural reliability analysis using deep neural network. Expert Syst Appl 2022;189:116104.
[35] Zhuang Xiaoying, Guo Hongwei, Alajlan Naif, Zhu Hehua, Rabczuk Timon. Deep autoencoder based energy method for the bending, vibration, and buckling analysis of Kirchhoff plates with transfer learning. Eur J Mech A Solids 2021;87:104225.
[36] Samaniego Esteban, Anitescu Cosmin, Goswami Somdatta, Nguyen-Thanh Vien Minh, Guo Hongwei, Hamdia Khader, et al. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. Comput Methods Appl Mech Engrg 2020;362:112790.
[37] Berg Jens, Nyström Kaj. A unified deep artificial neural network approach to partial differential equations in complex geometries. Neurocomputing 2018;317:28–41.
[38] Truong Viet-Hung, Vu Quang-Viet, Thai Huu-Tai, Ha Manh-Hung. A robust method for safety evaluation of steel trusses using gradient tree boosting algorithm. Adv Eng Softw 2020;147:102825.
[39] Khadilkar Aditya, Wang Jun, Rai Rahul. Deep learning–based stress prediction for bottom-up SLA 3D printing process. Int J Adv Manuf Technol 2019;102(5):2555–69.
[40] Nie Zhenguo, Jiang Haoliang, Kara Levent Burak. Stress field prediction in cantilevered structures using convolutional neural networks. J Comput Inf Sci Eng 2020;20(1):011002.
[41] Jiang Haoliang, Nie Zhenguo, Yeo Roselyn, Farimani Amir Barati, Kara Levent Burak. Stressgan: A generative deep learning model for two-dimensional stress distribution prediction. J Appl Mech 2021;88(5).
[42] Karniadakis George Em, Kevrekidis Ioannis G, Lu Lu, Perdikaris Paris, Wang Sifan, Yang Liu. Physics-informed machine learning. Nat Rev Phys 2021;3(6):422–40.
[43] Pilkey Walter D, Pilkey Walter D. Formulas for stress, strain, and structural matrices, Vol. 107. John Wiley & Sons New Jersey; 2005.
[44] Zahraei Seyed Mahdi, Heidarzadeh M. Destructive effects of the 2003 bam earthquake on structures. Asian J Civ Eng (Buil Housing) 2007.
[45] Zahrai Seyed Mehdi, Bolandi Hamed. Towards lateral performance of CBF with unwanted eccentric connection: A finite element modeling approach. KSCE J Civ Eng 2014;18(5):1421–8.
[46] Zahrai SM, Bolandi H. Numerical study on the impact of out-of-plane eccentricity on lateral behavior of concentrically braced frames. Int J Steel Struct 2019;19(2):341–50.
[47] Bolandi H, Zahrai SM. Influence of in-plane eccentricity in connection of bracing members to columns and beams on performance of steel frames. J Civ Eng (J School Eng) 2013.
[48] He Kaiming, Zhang Xiangyu, Ren Shaoqing, Sun Jian. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, p. 770–8.
[49] Hu Jie, Shen Li, Sun Gang. Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 7132–41.
[50] Zhang Han, Goodfellow Ian, Metaxas Dimitris, Odena Augustus. Self-attention generative adversarial networks. In: International conference on machine learning. PMLR; 2019, p. 7354–63.

[51] Carrera Erasmo, Cinefra Maria, Petrolo Marco, Zappino Enrico. Finite element analysis of structures through unified formulation. Wiley Online Library; 2014.
[52] Masci Jonathan, Meier Ueli, Cireşan Dan, Schmidhuber Jürgen. Stacked convolutional auto-encoders for hierarchical feature extraction. In: International conference on artificial neural networks. Springer; 2011, p. 52–9.
[53] Vaswani Ashish, Shazeer Noam, Parmar Niki, Uszkoreit Jakob, Jones Llion, Gomez Aidan N, et al. Attention is all you need. Adv Neural Inf Process Syst 2017;30.
[54] Loshchilov Ilya, Hutter Frank. Decoupled weight decay regularization. 2017, arXiv preprint arXiv:1711.05101.