Wheel Impact Test by Deep Learning:
Prediction of Location and Magnitude of Maximum Stress

Seungyeon Shin[1,a], Ah-hyeon Jin[1,a], Soyoung Yoo[1,3], Sunghee Lee[3],

ChangGon Kim[2], Sungpil Heo[2],

Namwoo Kang[1,3,*]

[1]Cho Chun Shik Graduate School of Mobility, KAIST, 34051, Daejeon, South Korea

[2]Hyundai Motor Company, 445706, Hwaseong-Si, Gyeonggi-Do, South Korea

[3]Narnia Labs, 34051, Daejeon, South Korea

[*]Corresponding author: nwkang@kaist.ac.kr

[a] Contributed equally to this work.

Abstract

The impact performance of the wheel during wheel development must be ensured through a wheel impact test for vehicle safety. However, manufacturing and testing a real wheel take a significant amount of time and money because developing an optimal wheel design requires numerous iterative processes of modifying the wheel design and verifying the safety performance. Accordingly, the actual wheel impact test has been replaced by computer simulations, such as Finite Element Analysis (FEA), but it still requires high computational costs for modeling and analysis. Moreover, FEA experts are needed. This study presents an aluminum road wheel impact performance prediction model based on deep learning that replaces the computationally expensive and time-consuming 3D FEA. For this purpose, 2D disk-view wheel image data, 3D wheel voxel data, and barrier mass value used for wheel impact test are utilized as the inputs to predict the magnitude of maximum von Mises stress, corresponding location, and the stress distribution of 2D disk-view. The wheel impact performance prediction model can replace the impact test in the early wheel development stage by predicting the impact performance in real time and can be used without domain knowledge. The time required for the wheel development process can be shortened through this mechanism.

1. Introduction

A vehicle wheel that is sufficiently durable to meet the safety requirement must be developed to ensure vehicle safety. Accordingly, a strict impact test must be performed during wheel development to test the impact damage. However, completing a wheel design involves inspecting the wheel safety through the wheel impact test, which takes a significant amount of time and cost due to the trial and error during the wheel development. Therefore, vehicle manufacturing companies need a solution to reduce time in wheel designing and manufacturing stages through a rapid impact analysis of the various design proposals.

Although the actual wheel impact test has been replaced by computer simulations, such as Finite Element Analysis (FEA) as Chang and Yang (2009) proposed, it is still time-consuming because the computationally expensive simulation process needs to be repeatedly executed, and FEA experts are needed to inspect the wheel performance. Accordingly, recent studies have suggested a process for predicting engineering performance through machine learning methodologies (Liang et al., 2018; Khadilkar et al., 2019; Madani et al., 2019; Nie et al., 2020; Yoo et al., 2021). These methods can greatly contribute to product development by accelerating the analysis process.

This study presents a 3D wheel impact performance prediction model based on deep learning that can replace the aluminum road wheel impact test. The purpose of this study is to replace the 3D FEA process for the wheel impact analysis, which requires high computational cost, to provide the impact performance of a wheel design in the conceptual designing stage, thereby reducing the time required for wheel development. Synthetic 3D wheel data were generated through the 3D wheel CAD automation process (Oh et al., 2019; Yoo et al., 2021; Jang et al., 2022) using the 2D disk-view images (spoke designs) and rim cross-sections, and the impact performance results were collected through FEA impact test simulation. Hence, we constructed a real time prediction model that predicts the magnitude of maximum von Mises stress, the corresponding location, and the overall stress distribution of 2D disk-view through this mechanism.

The impact performance of a wheel design can be checked in real time even in the conceptual designing stage by predicting the magnitude of maximum von Mises stress through the proposed model, and the location of the maximum stress can also be known, informing the parts to be supplemented in the wheel design. The overall von Mises stress distribution of 2D disk-view is also predicted, providing more information to the designer. Accordingly, this method can be easily utilized by general designers without engineering expertise, enabling rapid impact performance inspection of the various design proposals. The same process can be applied to any product that requires an impact test in addition to wheels.

The paper is organized as follows. Section 2 summarizes studies related to this study, and Section 3 presents the data collection and preprocessing step for training the model and the architecture of the proposed model. Section 4 discusses the prediction results. Finally, Section 5 presents the conclusions, limitations, and future work.

2. Related Work

Product development process requires repetitive iterations to find an optimal design. In particular, the design optimization for a product development is time-consuming and computationally costly because simulations, such as FEA and Computational Fluid Dynamics (CFD), which are essential in this process, are expensive. Therefore, studies to replace the simulation process by applying various machine learning methodologies have recently emerged (Deng et al., 2020; Lee et al., 2020; Qian & Ye, 2021; Zheng et al., 2021).

This study focuses on deep learning studies that predict the stress distribution of structures. Liang et al. (2018), which is an early study that applied deep learning to replace FEA, predicted the aortic wall stress distribution according to the shape of thoracic aorta. Shape encoding on the aorta shape was performed through principal component analysis (PCA) for this purpose, and the stress distribution of the aorta was predicted through a neural network. Madani et al. (2019) also proposed a study to predict the maximum von Mises stress value and the corresponding location for 2D arterial cross-sectional images to replace the finite element simulation using machine learning. Nie et al. (2020) conducted a study to predict the stress distribution of a 2D linear elastic cantilevered structure applied by static load to accelerate the structural analysis. In this study, two networks, namely, SCSNet and StressNet, were proposed, and the von Mises stress distribution was predicted by inputting various structures, external forces, and displacement boundary conditions. However, the aforementioned studies are predictions for 2D domain and have limits when applied in actual product development. In real-world problems, high-dimensional data, such as 3D data, must be dealt with. However, these high-dimensional data are difficult to train and require a large amount of training data. Therefore, an appropriate data representation method and training method for high-dimensional data must be devised to replace 3D simulation through deep learning.

A study that is similar to our study is Khadilkar et al. (2019), which proposed two CNN-based networks to predict the stress distribution by layer for the bottom-up SLA printing process among manufacturing methods. In particular, the 2-stream CNN network, which boasts higher performance among them, uses a binary image of the cross-section and a 3D model up to the previous layer as input, in the form of a point cloud, to predict the stress distribution of a layer cross-section. The 2D image passes through the convolutional layer, and the 3D point cloud enters the network by adding each feature vector that has passed through PointNet (Qi et al., 2017). This method has a similar input as ours. However, our proposed method predicts the maximum von Mises stress value and the corresponding location as well as the 2D stress distribution using voxel-based 3D data. Khadilkar et al. (2019) eventually predicted the stress distribution for 2D domains, while our proposed methodology can predict the 3D coordinates of the maximum von Mises stress location for 3D domain, enabling to replace the existing 3D FEA.

Two major issues need to be considered for deep learning in the 3D domain. First, a significant problem in the field of 3D deep learning is that it requires a large amount of training data. However, actually collecting a sufficient amount of 3D data in practice is difficult, so an idea is needed to construct a model with high accuracy even with limited data. Second, the representation of the 3D data is important when dealing with 3D data. In the field of 3D deep learning, representation methods, such as point cloud, mesh, and voxel, are commonly used.

The point cloud-based method represents the shape through a set of points distributed near the surface of a 3D shape (Bello et al., 2020). However, the point cloud method has a disadvantage in that it is difficult to express the details of a shape because it is sparse. The mesh-based method is a method for representing a 3D shape through a polygon-shaped face made of vertices. However, this method is sensitive to the quality of the input mesh, and the surface patch of the shape may not be stitched. The voxel-based method is a volumetric data, expressing 3D shape in a cube form. However, the voxel method takes up a large amount of memory storage because it expresses the occupied and the non-occupied parts (Ahmed et al., 2018). The resolution of the voxel needs to be increased to express the 3D shape in more detail, but the problem is that the higher the resolution, the more the parameters increase. Many studies have been conducted to solve the computational cost problem of the voxel method. For example, some studies have been performed to express 3D shapes with high resolution by proposing an Octree-based method (Häne et al., 2017; Riegler et al., 2017; Tatarchenko et al., 2017).

In this study, a prediction model based on a multi-modal autoencoder architecture that uses various types of data in parallel as inputs and outputs has been constructed to overcome the data shortage problem and reduce computational cost, utilizing a latent vector of the input to reduce the dimension of the high-dimensional data. To this end, we used voxel-based 3D wheel data to extract features of 3D CAD data using a 3D CNN-based convolutional variational autoencoder (cVAE). The training was carried out using the latent vector of 3D CAD data and 2D wheel image data through the pre-trained 3D cVAE model and the 2D convolutional autoencoder (cAE) model, and accurate results were derived even with 2,501 3D CAD data.

3. Deep Learning Framework for Wheel Impact Test

3.1. Overall Framework

The entire process of this study consists of four steps. In Stage 1, 3D road wheel CAD datasets were automatically generated. Spoke designs and rim cross-sections were selected, and data preprocessing was performed. In Stage 2, wheel impact analysis was performed with the generated 3D wheel data. Based on the analysis results, post-processing was performed to remove the outliers, and the magnitude of maximum von Mises stress and its location coordinates were extracted. Stage 3 is the stage of developing 3D cVAE and 2D cAE, which are dimensionality reduction models that will be used to improve the performance of the proposed model. These models were used to reduce the dimension of the input data. Finally, Stage 4 is the phase of developing a deep learning model that predicts the magnitude of maximum von Mises stress, the corresponding location coordinates (x, y, and z), and the overall von Mises stress distribution of 2D disk-view. The overall process proposed in this study is shown in Figure 1.

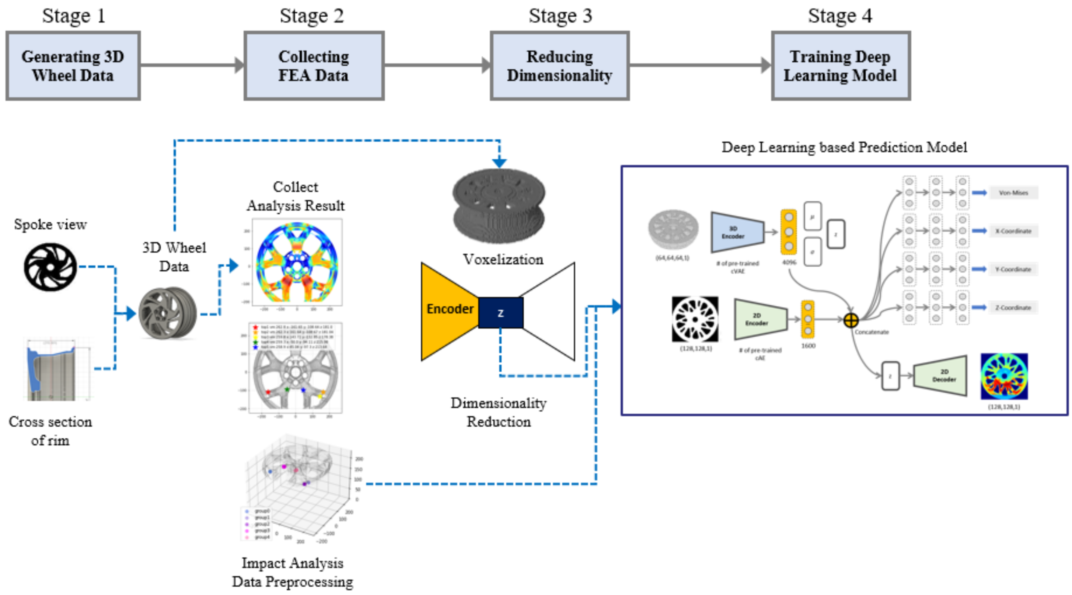


Figure 1. Overall framework of the proposed method

3.2. Stage 1: Generating 3D Wheel Data

Many detailed 3D wheel models used in reality are needed to construct an accurate wheel impact performance prediction model. However, a large amount of synthetic concept wheel data was generated and used for training because sufficient detailed 3D wheel data are difficult to obtain. In this study, the 3D CAD automation framework proposed by Yoo et al. (2021) was used. The framework consists of a stage that handles 2D spoke designs (disk-view images) and rim cross-sections and a stage that creates these into 3D CAD. This process automatically generates a large amount of 3D roadwheel CAD. First, 2D work dealing with spoke designs and rim cross-section images are proceeded. Accordingly, spoke designs and rim-cross section images were collected for this purpose, which is explained in Sections 3.2.1 and 3.2.2.

3.2.1 Disk-view spoke design data collection and preprocessing

First, 2D disk-view spoke design images for 3D CAD generation were collected in various ways. The three main collection methods were as follows: first, 603 binary wheel images available on the Internet were collected, and topology optimization using the collected images as the reference design was performed to collect 177 generative design piece wheels. The topology optimization was performed on wheel pieces, as shown in Figure 2, and they are rotated to make a complete wheel, solving the conventional problem of generative design that the symmetry of the generation result is not guaranteed. Here, 10 types of equal wheel pieces from 4 pieces to 13 pieces were used, and generative design was performed by diversifying the similarity condition, load ratio condition, and volume ratio conditions among the topology optimization conditions. In addition, 93 2D wheel images and 142 3D wheels from Hyundai Motors were converted into binary images. Thus, 1,015 2D disk-view spoke design images of 128×128 pixel were collected.

The collected spoke designs are post-processed through a series of processes, as shown in Figure 2. First, the edges of the anti-aliased 128×128 image are detected and saved using the Sobel operator. Among the edge values, the hub hole, which is the center hole of a wheel, and the edge of the outermost wheel are deleted, and only the edge values forming the spoke design are left. The coordinates of the remaining edge points are saved in a .csv file. A set of points forming a single loop are grouped together to use the randomly aligned points for CAD modeling. When the Euclidean distance between points is closer than a certain threshold, it is clustered as the same group. The points were evenly deleted to lower the density of the points in a sketch to draw a smooth spline. The pre-processed spoke design images are shown in Figure 3.

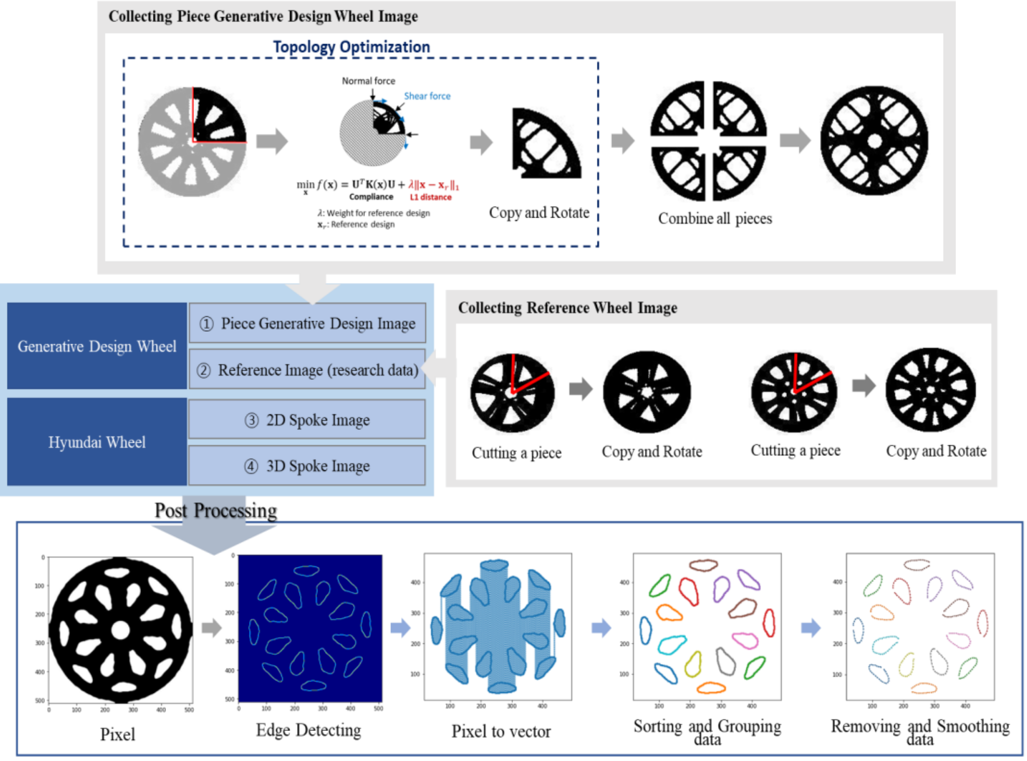

Collecting Piece Generative Design Wheel Image
Topology Optimization
Normal force
Shear force
$\min_{\mathbf{x}} f(\mathbf{x}) = \mathbf{U}^T\mathbf{K}(\mathbf{x})\mathbf{U} + \lambda\|\mathbf{x} - \mathbf{x}_r\|_1$
Compliance
L1 distance
λ: Weight for reference design
$\mathbf{x}_r$: Reference design
Copy and Rotate
Combine all pieces
Generative Design Wheel
① Piece Generative Design Image
② Reference Image (research data)
Hyundai Wheel
③ 2D Spoke Image
④ 3D Spoke Image
Collecting Reference Wheel Image
Cutting a piece
Copy and Rotate
Cutting a piece
Copy and Rotate
Post Processing
Pixel
Edge Detecting
Pixel to vector
Sorting and Grouping data
Removing and Smoothing data

Figure 2. Process of collecting spoke design data

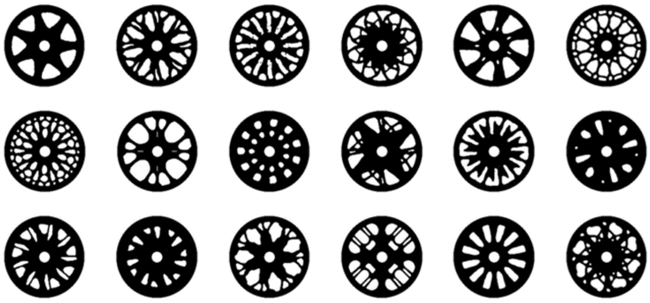
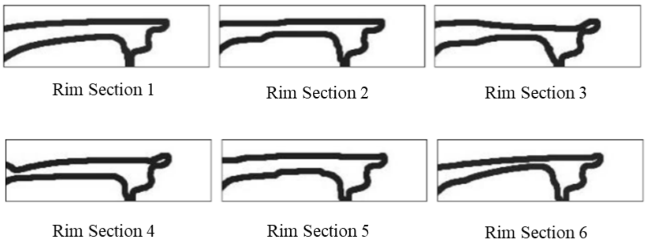


Figure 3. Samples of the 2D disk-view spoke design data

3.2.2 Rim cross-sectional data collection and preprocessing

In contrast with the spoke design images, several representative cross-sections were selected for the rim cross-section and combined with the spoke design images to create 3D wheels. The following section is about the dimension reduction for selecting the representative rim cross-sections and preprocessing the images. First, the rim cross-sections of the wheels were collected by various angles to select representative rim cross-sections. The 2D rim cross-sectional images were cropped to split the lower part of the rim, similar to that in Figure 4 for CAD modeling. Thereafter, 70×235 pixel data were trained with a cAE to reduce the features of the rim cross-sections into 64 dimensions, and they were clustered into six groups using K-means clustering and elbow method. Six representative rim cross-sections as in Figure 4 are selected, which are located at the center point of each cluster. Accordingly, we simplified CAD automation while maintaining the diversity of the rim cross-sections. The representative rim cross-sections selected through the process above goes through the same preprocessing steps as the spoke design images.



Rim Section 1

Rim Section 2

Rim Section 3

Rim Section 4

Rim Section 5

Rim Section 6

Figure 4. Rim cross-sectional data

In addition, the coordinates of the spoke part and the lower part of the rim were separately stored to ensure that the bodies could be individually created when modeling.

3.2.3 3D CAD automation

The following is a 3D CAD modeling process using the points of the 2D disk-view spoke design images and the rim cross-sections prepared above. Autodesk's Fusion 360 Application Programming Interface tool (Autodesk, 2022) was used for 3D wheel CAD automation. 3D modeling was automated by drawing lines and splines using a .csv file containing coordinates of the images. The 3D CAD generation procedure is shown in Figure 5. First, after sketching the coordinates corresponding to the spoke part of the rim, the spoke body was created through revolving the sketch. Next, the coordinates for the disk-view spoke design were sketched and extruded, and the predefined lug holes were also sketched and extruded through the spoke body. Finally, after sketching the coordinates for the lower part of the rim section, the rim body was created through revolving the sketch, and it was combined with the spoke body to create the final wheel body. The CAD was then extracted. This process was automatically generated until the desired number of wheels is created.

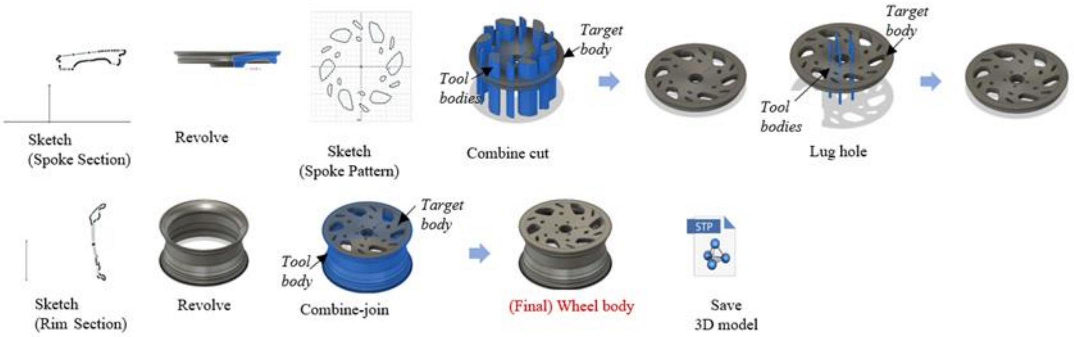


Figure 5. 3D CAD modeling automation process

3.3 Stage 2: Collecting FEA Data

3.3.1 Analysis Result Collection and Pre-processing

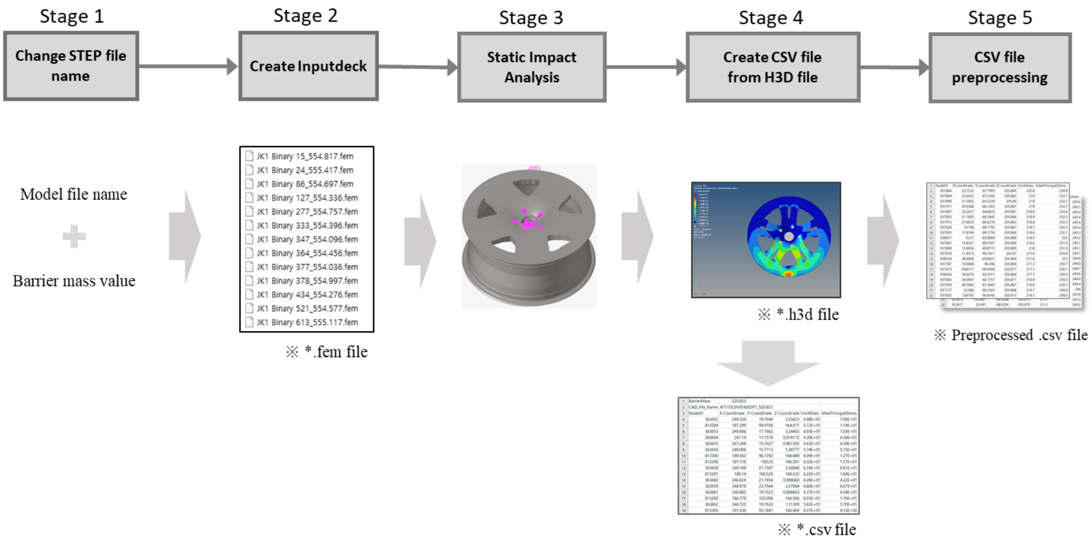
The 3D wheel model created by the 3D CAD automation process is subjected to wheel impact analysis to collect the von Mises stress response and preprocessed to use as training data. The collection of analysis result and preprocessing are conducted in five steps, as shown in Figure 6.

First, a barrier mass value, which means the impact load used for impact analysis, is assigned to the CAD file name of the generated CAD model. The barrier mass applied in the impact analysis is given a force with a load unit (kg) at the same location. In this study, loads from 498.0 kg to 558.0 kg were divided into 1,000 equal intervals and randomly applied to ensure that the loads are not biased.

An inputdeck file containing information on the analysis condition, including the impact location and the constraint conditions, is created for each CAD files by using an automation program. The impact location, where the barrier mass is applied, was designated at the air hole of a wheel, which is located at the center of the widest hole among the wheel spoke holes. Wheel Impact Analysis, which is an impact analysis automation program provided by Altair based on HyperWorks (Altair HyperWorks, 2020), was used to automatically generate the inputdeck files. As a condition applied to the impact analysis automation program, the result types were set to element stresses (2D and 3D) and von Mises stress, and the hub and bolt holes were constrained. In addition, the area near the air hole, which is the impact location, was set as the 3D elastic area.

Impact analysis was performed using Altair Optistruct (Altair Optistruct, 2020). The analysis was automated using a python-based code to continue it for 2,501 files. After the analysis was completed, each analysis result was extracted as a .csv file to be used as label data for training. This process was performed through Wheel Impact Analysis, which was previously mentioned. In the extracted .csv file, information about all the node IDs, xyz coordinates, von Mises stress values, and maximum principal stress values are recorded.

Before the location of the maximum von Mises stress was extracted from the analysis result file, the nodes with values that do not meet our criterion were first removed. The criterion are as follows: First, the nodes near the impact location were excluded, and the nodes at the lower part of the rim were also excluded. In addition, the nodes with negative maximum principal stress value were excluded. Accordingly, the stress received in the compressive direction can be excluded. The remaining nodes were arranged in a descending order based on the von Mises stress value. This process was performed using MATLAB (MATLAB, 2020).



JK1 Binary 15_554.817.fem
JK1 Binary 24_555.417.fem
JK1 Binary 86_554.697.fem
JK1 Binary 127_554.336.fem
JK1 Binary 277_554.757.fem
JK1 Binary 333_554.396.fem
JK1 Binary 347_554.096.fem
JK1 Binary 364_554.456.fem
JK1 Binary 377_554.036.fem
JK1 Binary 378_554.997.fem
JK1 Binary 434_554.276.fem
JK1 Binary 521_554.577.fem
JK1 Binary 613_555.117.fem

Figure 6. Process of preprocessing wheel impact analysis result data

3.3.2 Label Data Preprocessing

The following is the label data preprocessing to be used for training. Based on the analysis results obtained in Section 3.3.1, the maximum von Mises stress value, the corresponding coordinates, and the overall stress distribution in the 2D disk view are used as label. The maximum stress values were obtained after removing the outliers to predict accurate maximum von Mises stress value and its location. The maximum stress point can be found through three steps, as shown in Figure 7.

First, according to the preprocessed analysis result, the top 50 points are obtained based on the von Mises stress values. Next, the nodes with a 3D L2 distance of less than 10 mm from the location of maximum stress were grouped into one to cluster high stress concentrated areas with the top 50 nodes. After excluding nodes that have already been grouped, a cluster with a high stress concentration can be found by repeating the same process for the remaining nodes. Following the selection of the maximum stress point from the top cluster, the coordinates of the node and the von Mises stress value are stored and used as a labels for training.

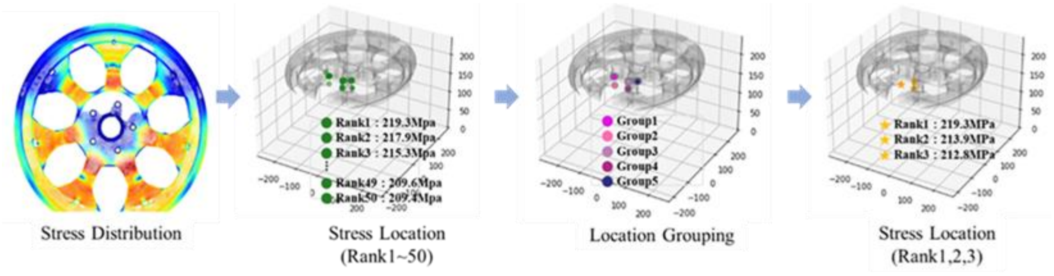
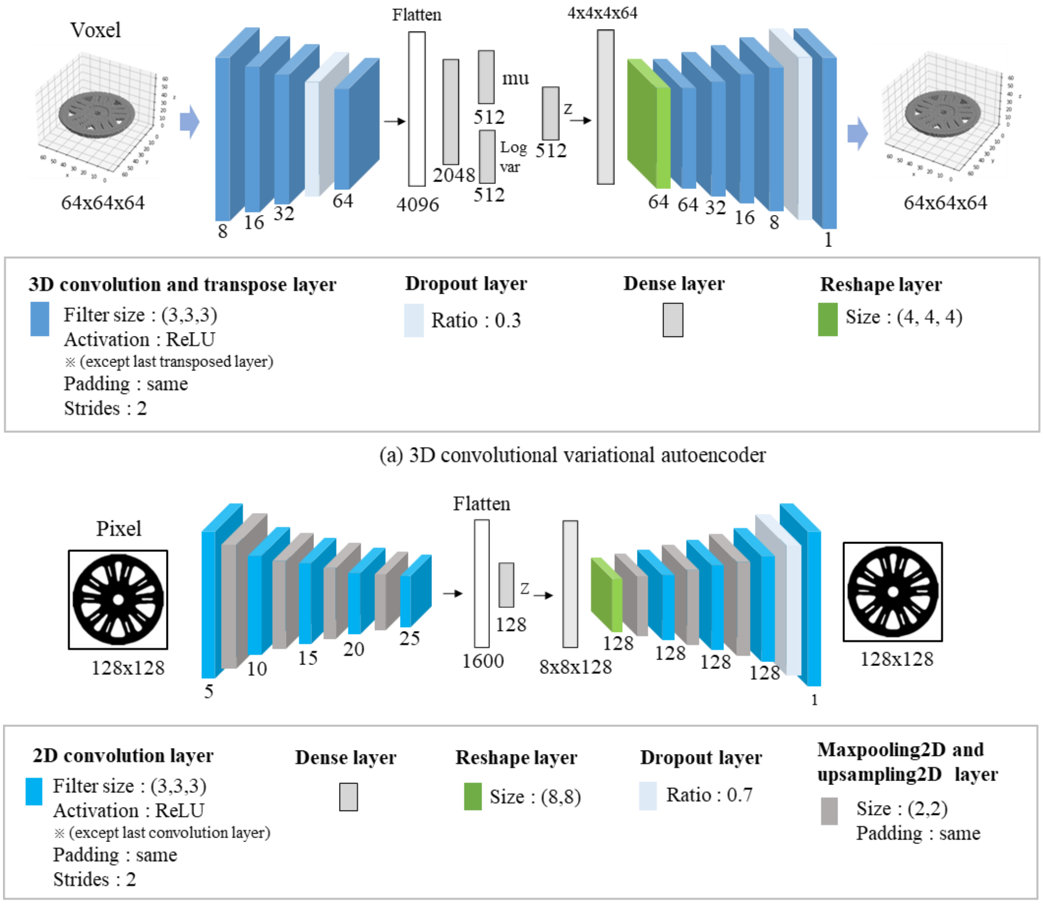


Figure 7. Process of finding the maximum von Mises stress location

3.4 Stage 3: Reducing Dimensionality

To learn the features of the wheels, two pretrained encoders were used to predict the impact performance of the wheels. These two encoders will be described in this section (Figure 8). The features of the 2D disk-view spoke images were compressed through a 2D cAE. cAE is a network that makes unsupervised learning into a form of supervised learning by using the input as label data, which is trained to ensure that the difference between the input and the output passed through the decoder is small. Consequently, the latent vector from the encoder compresses the features of the training data. The input pixel value of 2D cAE is 128×128, and only the encoder from the trained cAEs is used afterwards. In addition, the 3D convolutional variational autoencoder (cVAE) is a network that learns to represent the inputs as probability values based on probability distributions. This tool is mainly trained for the purpose of using the decoder. Nonetheless, the encoder is used in this study. The input voxel data have the dimension of 64×64×64, and the latent vector that compresses the features of the input is used to construct the proposed prediction model, which will be explained in Section 3.6.



(a) 3D convolutional variational autoencoder

(b) 2D convolutional autoencoder

Figure 8. Dimensionality reduction

(a) 3D convolutional variational autoencoder; (b) 2D convolutional autoencoder

3.5 Stage 4: Training Deep Learning Model

3.5.1 Data

The input data for the proposed model consists of 2D spoke design data, 3D voxel data, and the barrier mass value used for the impact analysis. The label data consist of maximum von Mises stress value, its coordinates, and a heatmap of the 2D disk-view stress distribution. A total of 2,501 3D road wheel CAD data were used, of which 1,753 (70%) wheel data were used as the training set, 374 (15%) wheel data as the validation set, and 374 (15%) wheel data as the test set.

First, the 2D spoke design data and 3D voxel data used as input will be explained. The 3D CAD data are high-dimensional data, and a deep learning model is difficult to train due to the nature of the high-dimensional data. The computational cost is also expensive. Accordingly, the lower part of the rim body was excluded, and only the spoke body was used to reduce the dimension of the CAD data and concentrate on the spoke body of the wheel. This approach was possible because all the wheels have the same shape of lower rim body. The 3D CAD data with only the spoke body of a wheel were converted to 64×64×64 voxel, covering approximately 7–8 mm per voxel. The 2D spoke design images corresponding to each 3D CAD data are represented by 128×128 pixel. The two types of wheel data are shown in Figure 9.

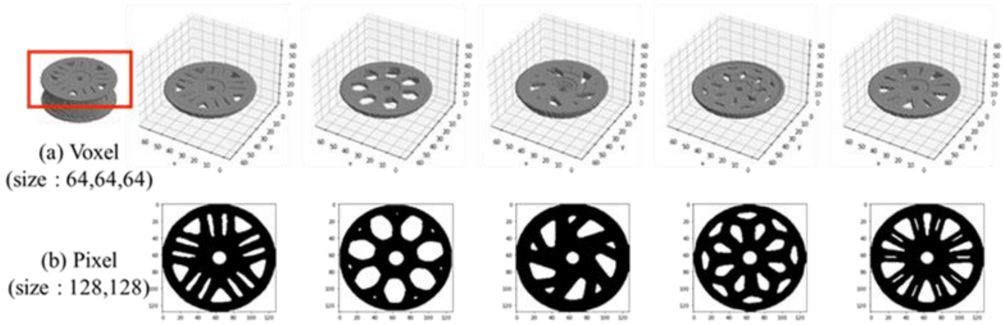
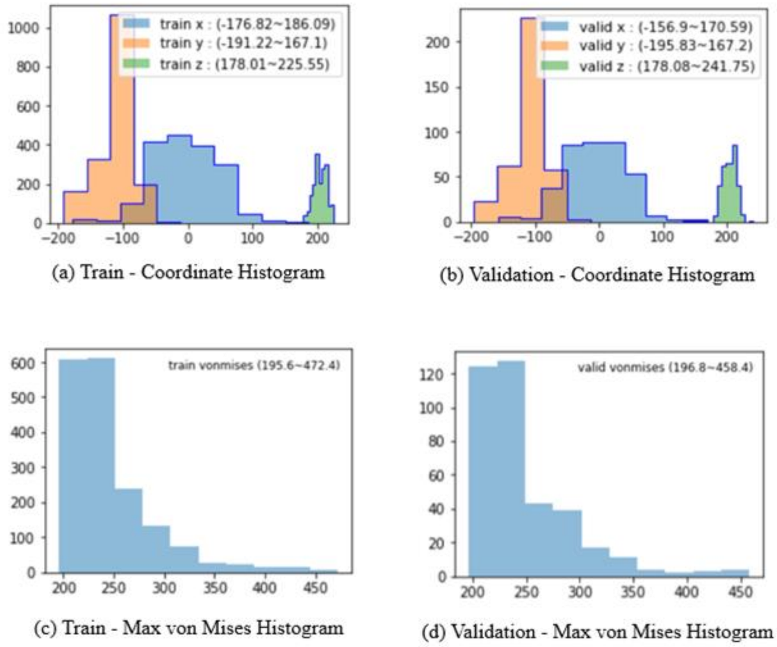


Figure 9. Input wheel data

(a) 3D voxel wheel data (64×64×64); (b) 2D pixel wheel image data (128×128)

Next, the maximum von Mises stress values, the corresponding coordinates, and the 2D disk-view stress distribution heatmap, which are used as the labels, will be explained. First, the histogram of the coordinates of the wheel data and the maximum von Mises stress values is shown in Figure 10. Given that the coordinates and stress values have different distributions and ranges, minmax scaling was applied for both values, allowing more stable training.



(a) Train - Coordinate Histogram

(b) Validation - Coordinate Histogram

(c) Train - Max von Mises Histogram

(d) Validation - Max von Mises Histogram

Figure 10. Histogram of the coordinates and magnitude of maximum von Mises stress

(a) Train dataset coordinate histogram; (b) validation dataset coordinate histogram; (c) train dataset magnitude of the maximum von Mises stress histogram; (d) validation dataset magnitude of maximum von Mises stress histogram

2D disk-view stress distribution heatmap was also used as the label. This approach allows the proposed model to learn not only the maximum von Mises stress value and its location but also the overall von Mises stress distribution of the 2D disk-view. The von Mises stress values were mapped to 128×128 matrix based on the analysis results obtained in Section 2.2 to create the corresponding heatmap data. The visualized image of the heatmap is shown in Figure 11. The red color indicates the part with a high von Mises stress value, and the blue color denotes the portion with a low von Mises stress value.

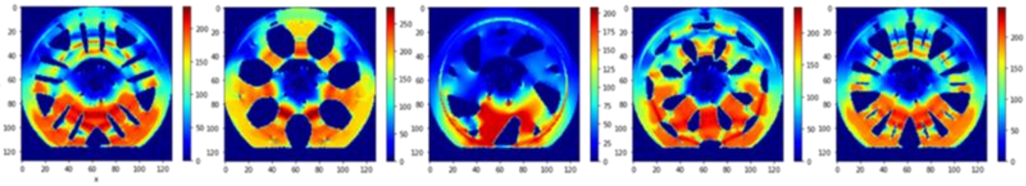


Figure 11. Stress distribution heatmap of 2D disk-view

3.5.2 Model Architecture

This section is about the architecture of the proposed model. Similar to Figure 12, the entire network is configured to predict the maximum von Mises stress value, the location, and the heatmap by inputting a 3D voxel spoke body, a 2D spoke design image, and a barrier mass value. The model was built using Tensorflow and Keras.

To improve the training performance, the 2D wheel image and the high-dimensional 3D voxel data received as inputs were used by extracting features through the pretrained 2D cAE and 3D cVAE described in Section 3.4. The latent vectors of the 2D wheel image and 3D voxel are combined with the barrier mass value and inputted into each regression model. The proposed network consists of five models to predict the maximum von Mises stress value, the corresponding xyz coordinates, and the stress distribution in 2D disk-view. First, a regression model composed of fully connected layers was used to predict the maximum von Mises stress coordinates and the stress value. Rectified linear unit (ReLU) was used for the activation function except for the last layer. Meanwhile, a linear activation function was used for the last layer. A decoder composed of 2D convolutional layers and 2D upsampling was used to predict the heatmap. A ReLU activation function was used except for the last layer, and a linear activation function was used for the last layer. The architecture for each model is shown in detail in Figure 12. The loss function based on mean squared error (MSE) used for the training is as shown in Equation (1):

$$Loss_{total} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 + \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \frac{1}{n}\sum_{i=1}^{n}(z_i - \hat{z}_i)^2 + \frac{1}{n}\sum_{i=1}^{n}(s - \hat{s}_i)^2 + \frac{1}{n}\sum_{i=1}^{n}(I_i - \hat{I}_i)^2, \quad (1)$$

where n is the number of input data, i is the i-th input data, x_i is the ith ground truth x-coordinate, $\hat{x}_i$ is the i-th predicted x-coordinate, y_i is the i-th ground truth y-coordinate, $\hat{y}_i$ is the i-th predicted y-coordinate, z_i is the i-th ground truth z-coordinate, $\hat{z}_i$ is the ith predicted z-coordinate, s_i is the i-th ground truth von Mises stress value, $\hat{s}_i$ is the i-th predicted von Mises stress value, I_i is the i-th ground truth heatmap, and $\hat{I}_i$ is the i-th predicted heatmap. Adam Optimizer was used for training, using a learning rate of 0.0001, batch size of 32, and 1,000 epochs. Bayesian Optimization-based hyperparameter search and random search-based hyperparameter search were tested to increase the accuracy of the model. In conclusion, the hyperparameters, such as learning rate, batch size, and epoch, were selected through random search.

The MSE loss for the epoch during the training process is shown in Figure 13. The training result converges well for the training and validation sets.

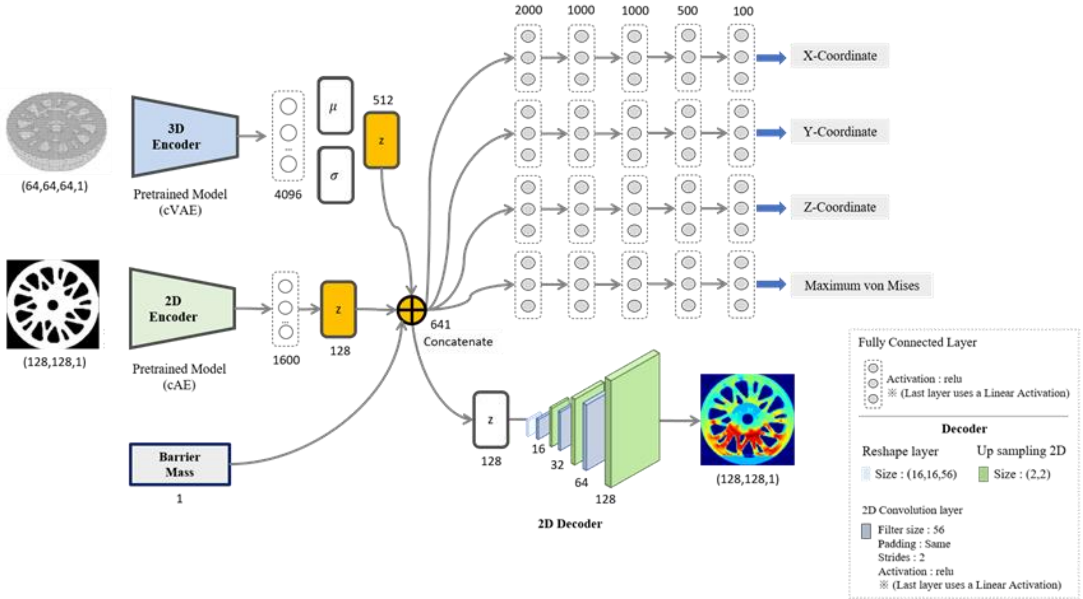


Figure 12. Proposed model architecture

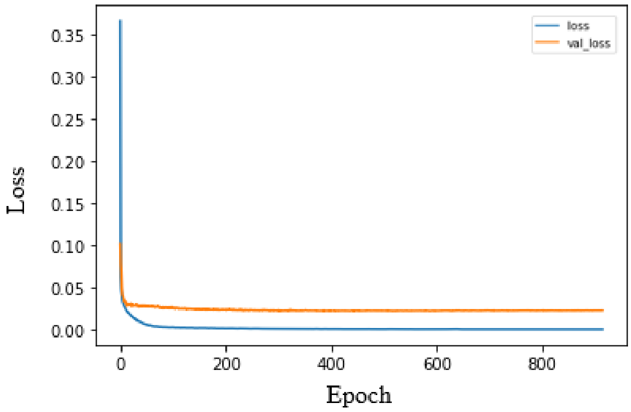


Figure 13. MSE loss for each epoch in the training process

4 Results and Discussion

4.1 Proposed Model Prediction Result

Various metrics were used to evaluate the prediction results of the model. The mean and median 3D Euclidean distance errors were used for the coordinates. Meanwhile, the mean absolute percentage error (MAPE) and Pearson correlation R value were checked for von Mises stress values. The accuracy of the heatmap was evaluated by root mean square error (RMSE). The equations for the 3D Euclidean distance error, MAPE, and RMSE are as follows:

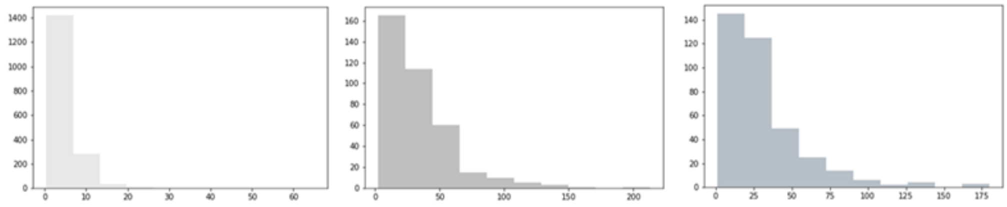
$$Euclidean\ Distance\ Error\ _{3D} = \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + \ (z_i - \hat{z}_i)^2}\ , \tag{2}$$

$$MAPE\ _{von\ Mises} = \ 100 \times \ \frac{1}{n}\ \sum_{i=1}^{n} \left|\frac{s_i - \hat{s}_i}{s_i}\right|, \tag{3}$$

$$RMSE\ _{heatmap}\ = \ \sqrt{\frac{1}{n}\ \sum_{i=1}^{n} (I_i - \hat{I}_i)^2}\ . \tag{4}$$

The test results for the proposed model with the corresponding metric show that the mean 3D Euclidean distance error is 31.49 mm, and the median 3D Euclidean distance error is 24.73 mm. Considering that the size of one voxel of the corresponding 3D CAD data is 7–8 mm, a meaningful prediction result has been obtained. Figure 14 shows a histogram of the overall distribution of 3D Euclidean error with respect to the predicted coordinates. The figure shows that the validation and test results have an error of 50 mm or less. In addition, the relative 3D Euclidean distance error was examined to compare the Euclidean distance error relative to the wheel size. The Euclidean distance error was normalized based on the length of the wheel diameter (about 483 mm). The relative mean 3D Euclidean distance error was 6.52%, and the relative median 3D Euclidean distance error was 5.12%. The MAPE of the maximum von Mises stress value was 2.99%, and the R value was 0.96, indicating that accurate prediction was possible for the test set.

The scatter plots of the predicted maximum von Mises stress values for the train, validation, and test sets are shown in Figure 15. The scatter plots show that the prediction accuracy is slightly low for wheels with a large maximum von Mises stress value. This result is obtained because the maximum von Mises stress value of most training datasets is less than 300 MPa, similar to that in Figure 10(c). If we additionally collect wheel data with large maximum von Mises stress value to solve this data imbalance problem, then better training results can be obtained. Finally, the RMSE for each pixel of the restored heatmap was 0.0490, showing that the heatmap was meaningfully restored. The test results of the proposed model and the training and validation sets are organized in Table 1.

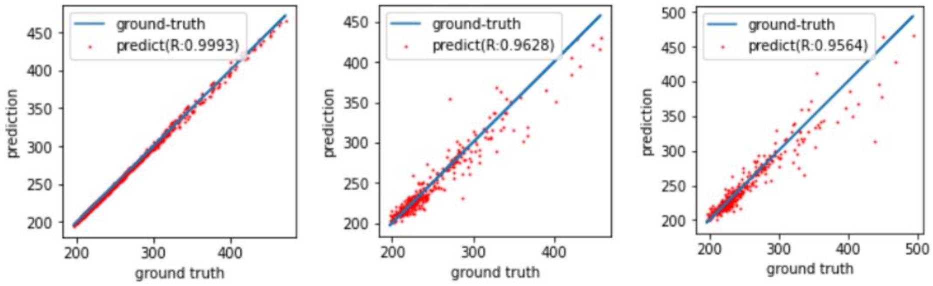


(a) Train (b) Validation (c) Test

Figure 14. Histogram of the coordinate 3D Euclidean distance error

Table 1. Proposed model prediction result

Dataset	3D Euclidean Distance Error (Mean) (mm)	3D Euclidean Distance Error (Median) (mm)	Relative 3D Distance Error (Mean) (%)	Relative 3D Distance Error (Median) (%)	Von Mises MAPE (%)	Von Mises (R)	Heatmap RMSE (MPa)
Train	5.40	4.79	1.12	0.99	0.44	0.99	0.0332
Validation	34.23	26.97	7.09	5.58	2.87	0.96	0.0480
Test	31.49	24.73	6.52	5.12	2.99	0.96	0.0490



(a) Train (b) Validation (c) Test

Figure 15. Scatter plot of the model's prediction and the ground truth

(a) Scatter plot of the train dataset; (b) scatter plot of the validation dataset; (c) scatter plot of the test dataset

As discussed above, the prediction accuracy of the maximum stress value and the stress distribution heatmap was high. Accordingly, a more detailed analysis on the predicted coordinates was conducted. The 1D Euclidean distance errors for the x-coordinate, y-coordinate, and z-coordinate were examined. The result is same as the above-mentioned 3D Euclidean distance error equation, with only a difference that it is expressed in one dimension. In Table 3, the relative mean 1D Euclidean distance errors of the test set are 5.20%, 2.36%, and 1.32% for the x-coordinate, y-coordinate, and z-coordinate, respectively. Meanwhile, the relative median 1D Euclidean distance errors of the test set are 3.72%, 1.36%, and 1.03% for the x-coordinate, y-coordinate, and z-coordinate, respectively. The Euclidean distance error results for each coordinate for the results of the training, validation, and test sets are organized in Table 2. The result shows that the prediction rate for the z-coordinate is relatively high. Given that the z-axis corresponds to the height direction of a wheel, it is expected to be relatively easy to predict. However, the prediction rate for the x-coordinate is the lowest, which is considered to be the most difficult coordinate because the wheel has a symmetrical design with respect to the impact location.

Table 2. Proposed model 1D coordinate prediction result

Dataset	X Distance Error (Mean) (mm)	X Distance Error (Median) (mm)	Y Distance Error (Mean) (mm)	Y Distance Error (Median) (mm)	Z Distance Error (Mean) (mm)	Z Distance Error (Median) (mm)
Train	3.43	2.60	2.14	1.75	2.34	2.06
Validation	27.32	20.64	12.28	6.78	7.05	5.48
Test	25.14	17.98	11.38	6.55	6.38	4.97

Table 3. Proposed model 1D coordinate prediction result (relative error)

Dataset	X Relative Distance Error (Mean)	X Relative Distance Error (Median)	Y Relative Distance Error (Mean)	Y Relative Distance Error (Median)	Z Relative Distance Error (Mean)	Z Relative Distance Error (Median)
Train	0.71	0.54	0.44	0.36	0.48	0.43
Validation	5.66	4.27	2.54	1.40	1.46	1.14
Test	5.20	3.72	2.36	1.36	1.32	1.03

Finally, we visualized the prediction results of the proposed model. The visualization of the ground truth maximum von Mises stress location and the predicted location is shown in Figure 16. Figure 17 shows a rendered wheel data with the ground truth and the prediction location.

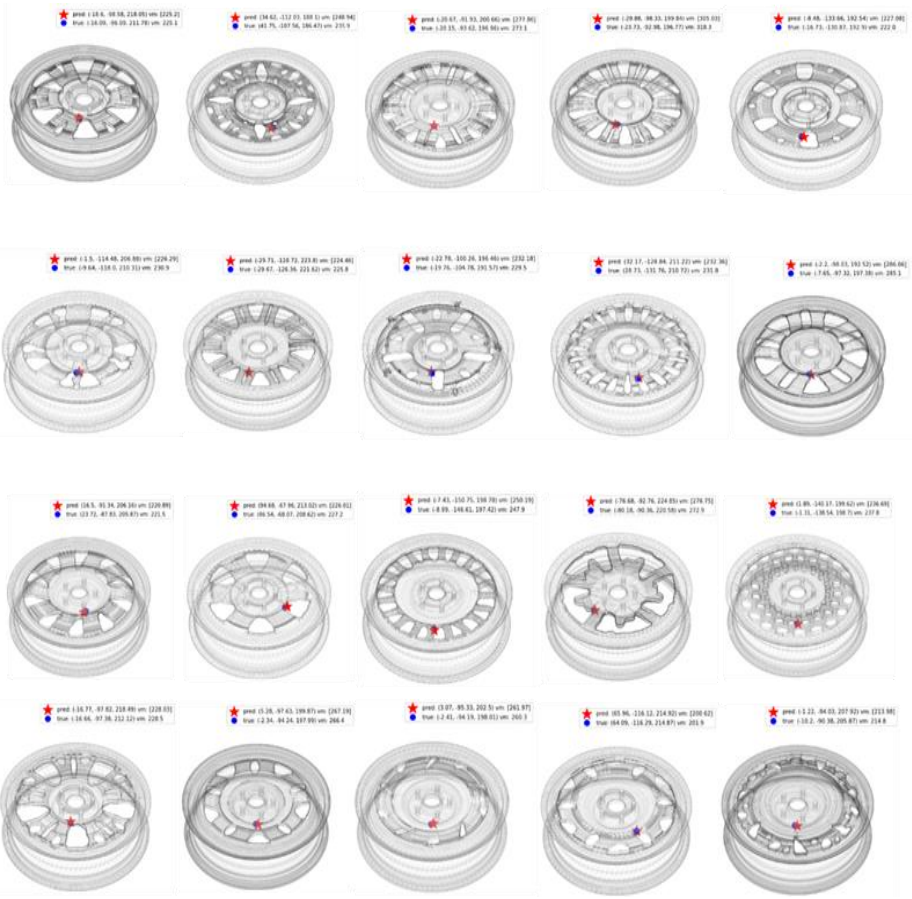


Figure 16. Visualization of the prediction location (red star) and the ground truth location (blue dot)

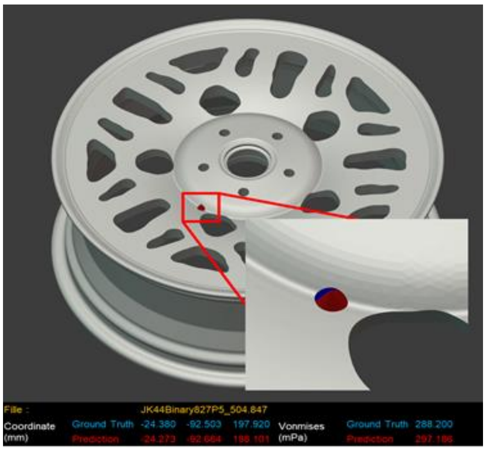


Figure 17. Visualization of the rendered wheel and the prediction (ground truth [blue point] and prediction [red point])

In addition, the comparison between the ground truth stress distribution heatmap and the prediction stress distribution heatmap is shown in Figure 18. The figure shows that the heatmap is meaningfully reconstructed.

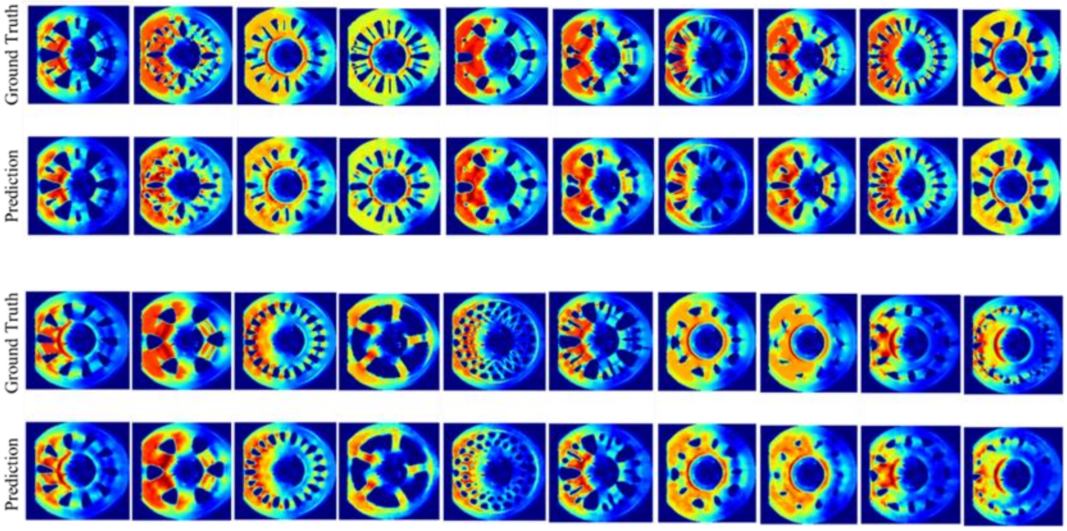


Figure 18. Comparison between the ground truth and the prediction of the stress distribution heatmap

Figure 19 shows the result for a case that is poorly predicted. Figure 19(a) is a visualization of the predicted maximum stress location and the actual location. Figure 19(b) shows the ground truth and the prediction heatmap corresponding to the wheel in Figure 19(a). The results show that most poorly predicted cases were predicted to a location where a high stress of a similar size occurred near the actual maximum stress location. Given that many locations have high stress of similar size, users should determine the vulnerable area by not only looking at the location of the maximum stress and its value but also the overall stress distribution to make an accurate judgement.

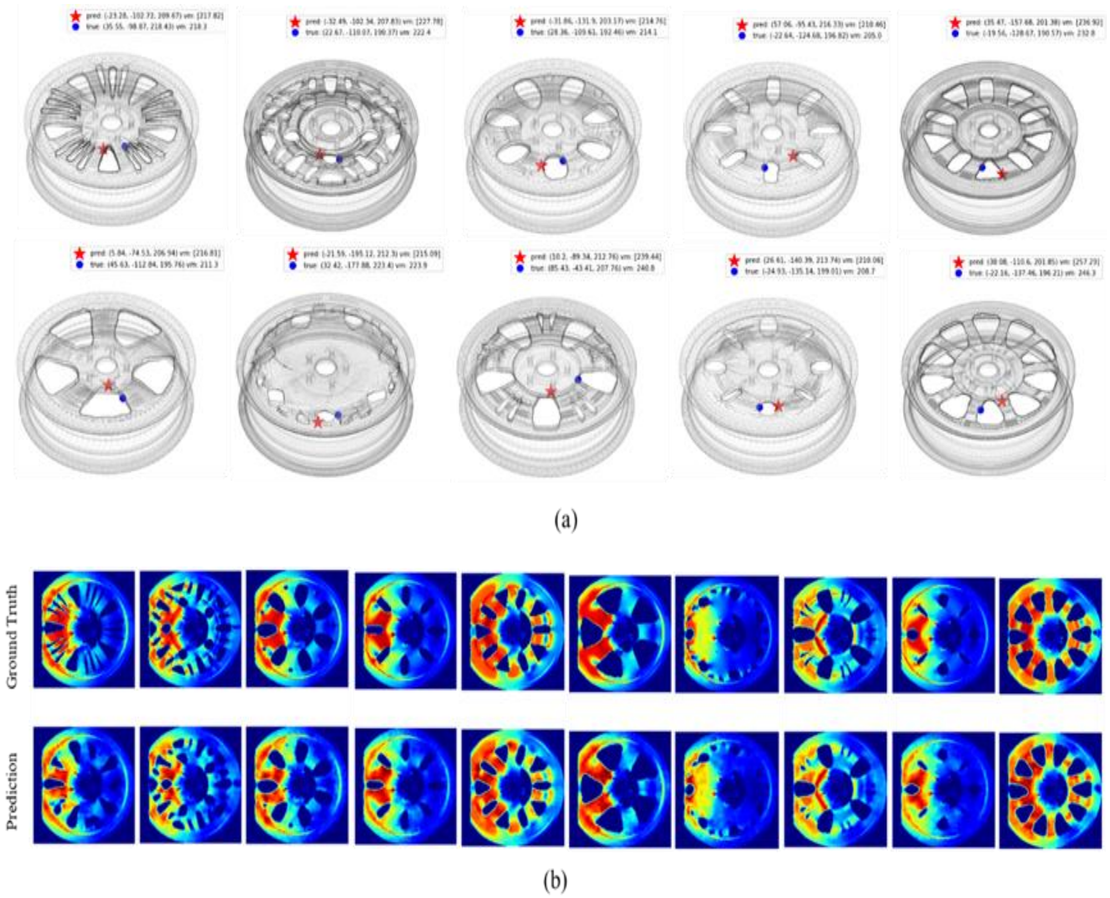


Figure 19. Visualization of the bad prediction cases

(a) Ground truth and prediction location; (b) ground truth and prediction of the stress distribution heatmap

4.2 Architecture Comparison

As shown in Section 3.5, the model used various inputs in parallel, and the maximum von Mises stress value, the coordinates, and the overall stress distribution of 2D disk-view can be predicted with high accuracy through this mechanism.

The prediction accuracy may decrease with the increase in the number of variables to be predicted because the training weight differs for each variable to be predicted during the training process. We examined how additionally predicting the stress distribution heatmap of 2D disk-view affects the model performance by comparing the prediction performance of Model B, which does not predict the stress distribution heatmap and the proposed model. Model B has the same architecture and hyperparameters as the proposed model except for the decoder, and the results are as follows. In the case of Model B, for the test set results, the 3D mean relative Euclidean distance error was 6.92%, the 3D median relative Euclidean distance error was 5.47%, and the MAPE for the maximum von Mises stress value was 3.07%, as compared in Table 5. However, the 3D mean relative Euclidean distance error is 6.52%, the 3D median relative Euclidean distance error is 5.12%, and the MAPE for the maximum von Mises stress value is 2.99% in the case of our proposed model. Therefore, the proposed model can predict 2D disk-view stress distribution while maintaining high prediction performance for the maximum stress value and the coordinates.

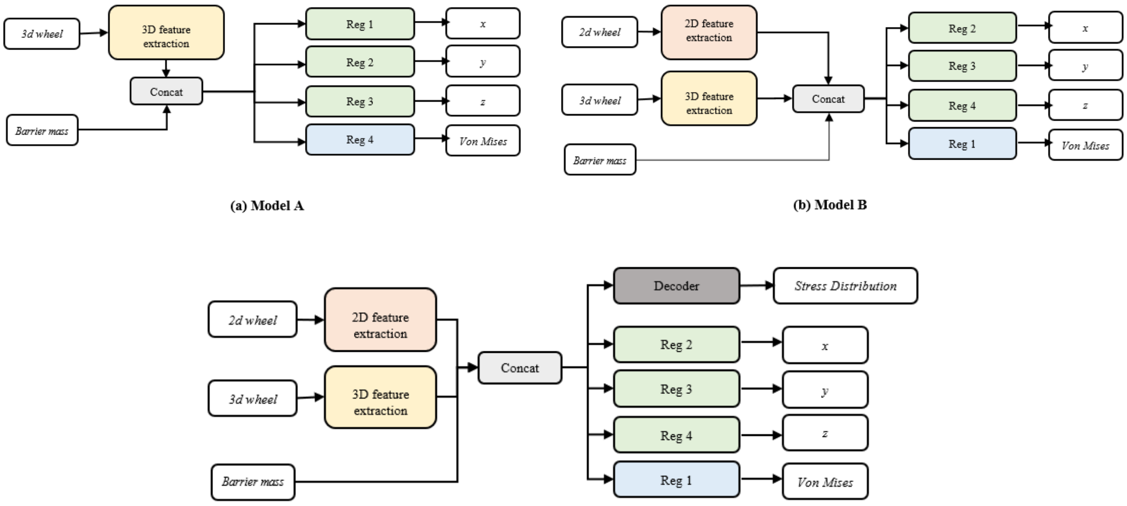
Furthermore, we compared the prediction results of Model A, which does not use 2D wheel image as the input, and Model B under the same condition that the stress distribution heatmap of 2D disk-view is not restored to examine the performance change of the model according to the input type. Model A was also trained with the same architecture and hyperparameters as Model B except for the input type. The comparison in Tables 4 and 5 confirmed that the method using 3D wheel data and 2D wheel image together is more effective in predicting the maximum von Mises stress value. A figure comparing the structures of Model A, Model B, and the proposed model is shown in Figure 20.

Table 4. Different models' validation dataset prediction result

Model	3D Euclidean Distance Error (Mean) (mm)	3D Euclidean Distance Error (Median) (mm)	Relative 3D Distance Error (Mean) (%)	Relative 3D Distance Error (Median) (%)	Von Mises MAPE (%)	Von Mises (R)
Model A	34.64	27.08	7.17	5.61	3.69	0.93
Model B	34.21	27.93	7.08	5.78	2.81	0.97
Proposed Model	34.23	26.97	7.09	5.58	2.87	0.96

Table 5. Different models' test dataset prediction result

Model	3D Euclidean Distance Error (Mean) (mm)	3D Euclidean Distance Error (Median) (mm)	Relative 3D Distance Error (Mean) (%)	Relative 3D Distance Error (Median) (%)	Von Mises MAPE (%)	Von Mises (R)
Model A	33.53	25.08	6.94	5.19	3.88	0.93
Model B	33.42	26.42	6.92	5.47	3.07	0.95
Proposed Model	31.49	24.73	6.52	5.12	2.99	0.96



(a) Model A

(b) Model B

(c) Proposed Model

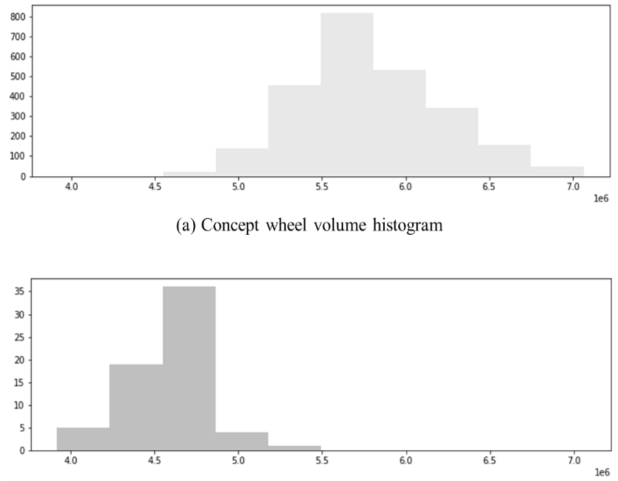
Figure 20. Comparison between the proposed model and other models

(a) Architecture of Model A; (b) architecture of Model B; (c) architecture of the proposed model

4.3 Different Domain Data

We checked the prediction performance of the proposed model with 63 detailed wheel data (which are used in real vehicles) provided by Hyundai Motors to examine the applicability of the proposed method in actual product development, which is the ultimate goal of our research. Table 6 shows that the relative mean 3D Euclidean distance error was 9.13%, the relative median 3D Euclidean distance error was 6.91%, the MAPE of the maximum von Mises stress value was 9.04%, and the RMSE for the restored heatmap was 0.1212. Figure 22 shows the distribution of the 3D Euclidean distance error and the scatter plot for the maximum von Mises stress values.

The result in Figure 23(a) and the relative 3D Euclidean distance error confirmed that the vulnerable location of a wheel was well predicted by looking at the wheel features even though the data domains were different. However, the prediction result for the maximum stress value is inferior. Given that the detailed wheel data have a detailed design, such as intaglio (curved surface), it is quite different from that of training data, and the distributions of the x- and y-coordinate of the wheel data are distinct from that of the training dataset, the concept wheel data. In particular, the scatter plot of the prediction result shows that most of the predicted maximum von Mises stress values were predicted to be lower than the actual stress values, and the predicted heatmap in Figure 23(b) was also predicted as a lower value overall. This problem is due to the maximum von Mises stress value of the detailed wheel that is inevitably higher than that of the concept wheel for the similar spoke design due to the detailed design. Similar to that in Figure 21, which shows the volume distribution of the two domains, the average volume of the detailed wheel is 4.62e6 mm^3, and the average volume of the concept wheel is 5.78e6 mm^3. This finding shows that the volume of the detailed wheel is relatively small. Thus, the maximum von Mises stress value of the detailed wheel is bound to be larger. Therefore, additional research is needed to predict the maximum von Mises stress value in the actual wheel data domain due to this problem. However, the maximum stress location was well predicted.



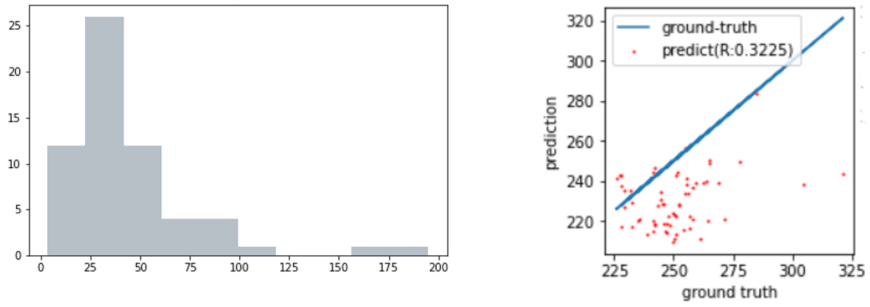
(a) Concept wheel volume histogram

(b) Detailed wheel volume histogram

Figure 21. Comparison between the concept wheel data volume and the detailed wheel data volume

Table 6. Proposed model detailed wheel dataset prediction result

Dataset	3D Euclidean Distance Error (Mean) (mm)	3D Euclidean Distance Error (Median) (mm)	Relative 3D Distance Error (Mean) (%)	Relative 3D Distance Error (Median) (%))	Von Mises MAPE (%)	Von Mises (R)	Heatmap RMSE (MPa)
Detailed Wheel	44.10	33.40	9.13	6.91	9.04	0.32	0.1212

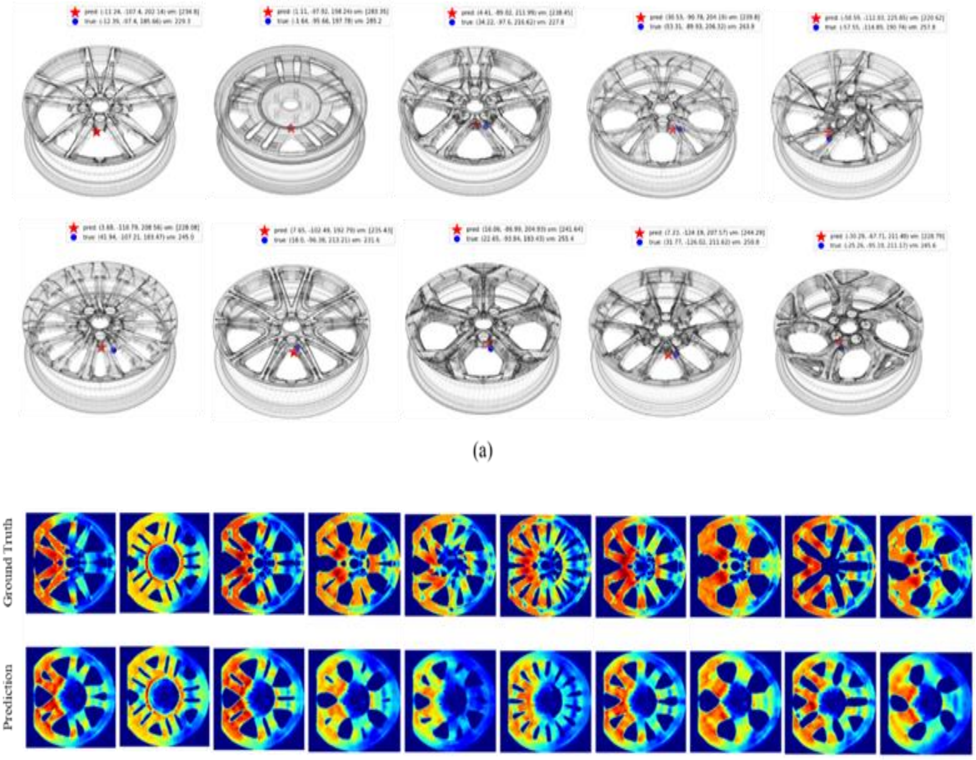


(a) Euclidean Distance Error

(b) Scatter Plot

Figure 22. Detailed wheel dataset prediction result

(a) 3D Euclidean distance error histogram; (b) scatter plot between the proposed model's prediction and the ground truth



(a)

Figure 23. Visualization of the prediction for a detailed wheel dataset

(a) Ground truth and prediction location; (b) ground truth and prediction of the stress distribution heatmap

The proposed model took about 27.41 min to train with one GPU (Geforce RTX 3090). Meanwhile, prediction of the impact performance of a wheel data with the trained model took only about 0.001 s. This finding shows that this method is highly effective compared with the conventional impact analysis, such as FEA, which requires an average of 1h or more (CPU 64 CORE) to analyze wheel data. Therefore, we proved that our methodology is suitable for accelerating the 3D FEA process and reducing the computational cost.

5 Conclusion

This study proposes a real-time wheel impact performance prediction model based on deep learning to replace the computationally expensive and time-consuming 3D FEA process. The proposed model can play a role in helping designers to quickly derive an optimal design without engineering knowledge by providing quick prediction about the impact performance, even with a conceptual design at the design phase. In particular, this study shows the applicability to actual product development because we applied this method to 3D synthetic road wheel data. Considering the characteristics of 3D CAD data, which are difficult to collect a large amount of data and are high-dimensional data, we derived an accurate prediction model by using latent vectors containing the features of input data through pretrained autoencoder-based models. Consequently, a prediction model with high accuracy was constructed with only a total of about 2,501 3D CAD data. The model constructed in this way is meaningful because it replaces the 3D FEA and provides real-time impact analysis results of an unseen wheel design because not only the maximum von Mises stress value but also the location of the maximum stress occurrence and the stress distribution of 2D disk-view, can be predicted.

The contribution of this study is summarized as follows. First, to our knowledge, this is the first study to apply 3D deep learning to vehicle system impact analysis. A 3D deep learning model capable of predicting the maximum stress value, the coordinates, and the stress distribution is presented. Accordingly, impact analysis results for a new design can be obtained in real time through this mechanism.

Second, we propose a multi-modal autoencoder architecture that uses various types of data in parallel as input and output. This architecture has a structure in which a 3D voxel, a 2D image, and a scalar value are inputted in parallel. Meanwhile, 2D images are restored in parallel with a vector value as output. The proposed architecture was verified through an ablation study.

Third, the data shortage problem was overcome through transfer learning using a 3D convolutional variational autoencoder (cVAE). After extracting features by reducing the dimension through 3D VAE for input data, the encoder was used during transfer learning for supervised learning because the 3D data require a large amount of data to directly perform supervised learning.

However, this study has some limitations. First, direct application of this model to the detailed design used in the actual product has a limitation because it is trained with the concept design data of road wheel, as mentioned in Section 4.3. Accordingly, we intend to perform further research about this model. Considering that sufficient amount of detailed design data is difficult to obtain, we would like to conduct a future study to expand it to detailed roadwheel design by applying the domain adaptation of Ganin and Lempitsky (2015) based on the proposed model trained from the concept wheel design. In addition, the accuracy of the prediction model will increase if more 3D wheel data are collected. Given that a large amount of 3D CAD data is difficult to collect, we would like to use physics-informed neural network (Raisi et al., 2019; Raissi et al., 2020), which is based on physical information, to solve the data shortage problem and reduce unnecessary computational costs in the data collection process. This method is expected to increase the accuracy of the prediction model because the training is proceeded by utilizing physics informed loss function rather than simply predicting by black box.

Acknowledgements

This work was supported by Hyundai Motor Company, the National Research Foundation of Korea (2018R1A5A7025409), and the Ministry of Science and ICT of Korea (No.2022-0-00969).

Reference

Ahmed, E., Saint, A., Shabayek, A. E. R., Cherenkova, K., Das, R., Gusev, G., ... & Ottersten, B. (2018). A survey on deep learning advances on different 3D data representations. arXiv preprint arXiv:1808.01462.

Autodesk. (2022). Retrieved from https://www.autodesk.com/products/fusion-360/

Bello, S. A., Yu, S., Wang, C., Adam, J. M., & Li, J. (2020). Deep learning on 3D point clouds. Remote Sensing, 12(11), 1729.

Chang, C. L., & Yang, S. H. (2009). Simulation of wheel impact test using finite element method. Engineering Failure Analysis, 16(5), 1711-1719.

Deng, C., Qin, C., & Lu, W. (2020). Deep-Learning-Enabled Simulated Annealing for Topology Optimization. arXiv preprint arXiv:2002.01927.

Ganin, Y., & Lempitsky, V. (2015, June). Unsupervised domain adaptation by backpropagation. In International conference on machine learning (pp. 1180-1189). PMLR.

Häne, C., Tulsiani, S., & Malik, J. (2017, October). Hierarchical surface prediction for 3d object reconstruction. In 2017 International Conference on 3D Vision (3DV) (pp. 412-420). IEEE.

Jang, S., Yoo, S., & Kang, N. (2022). Generative design by reinforcement learning: enhancing the diversity of topology optimization designs. Computer-Aided Design, 146, 103225.

Khadilkar, A., Wang, J., & Rai, R. (2019). Deep learning–based stress prediction for bottom-up SLA 3D printing process. The International Journal of Advanced Manufacturing Technology, 102(5), 2555-2569.

Lee, S., Kim, H., Lieu, Q. X., & Lee, J. (2020). CNN-based image recognition for topology optimization. Knowledge-Based Systems, 105887.

Liang, L., Liu, M., Martin, C., & Sun, W. (2018). A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis. Journal of The Royal Society Interface, 15(138), 20170844.

Madani, A., Bakhaty, A., Kim, J., Mubarak, Y., & Mofrad, M. R. (2019). Bridging finite element and machine learning modeling: stress prediction of arterial walls in atherosclerosis. Journal of biomechanical engineering, 141(8).

Nie, Z., Jiang, H., & Kara, L. B. (2020). Stress field prediction in cantilevered structures using convolutional neural networks. Journal of Computing and Information Science in Engineering, 20(1), 011002.

Oh, S., Jung, Y., Kim, S., Lee, I., & Kang, N. (2019). Deep generative design: Integration of topology optimization and generative models. Journal of Mechanical Design, 141(11).

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 652-660).

Qian, C., & Ye, W. (2021). Accelerating gradient-based topology optimization design with dual-model artificial neural networks. Structural and Multidisciplinary Optimization, 63(4), 1687-1707..

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational physics, 378, 686-707.

Raissi, M., Yazdani, A., & Karniadakis, G. E. (2020). Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. Science, 367(6481), 1026-1030.

Riegler, G., Osman Ulusoy, A., & Geiger, A. (2017). Octnet: Learning deep 3d representations at high resolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3577-3586).

Tatarchenko, M., Dosovitskiy, A., & Brox, T. (2017). Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In Proceedings of the IEEE international conference on computer vision (pp. 2088-2096).

Yoo, S., Lee, S., Kim, S., Hwang, K. H., Park, J. H., & Kang, N. (2021). Integrating deep learning into CAD/CAE system: generative design and evaluation of 3D conceptual wheel. Structural and Multidisciplinary Optimization, 64(4), 2725-2747.

Zheng, L., Kumar, S., & Kochmann, D. M. (2021). Data-driven topology optimization of spinodoid metamaterials with seamlessly tunable anisotropy. Computer Methods in Applied Mechanics and Engineering, 383, 113894.