

Short-time Prediction Tool for Flow Fields around Side Mirrors Using Pix2Pix GAN

Takuro HASHIMOTO¹Koji OKABE¹Takeshi YAMATODA²Takafusa KIDO²

ABSTRACT

To examine the design direction of side mirrors in the early stages of development, a Pix2Pix GAN deep learning model was used to develop an easy-to-operate tool that enables prediction of flow fields with sufficient accuracy in a short period of time and output of the results. In addition to a shorter prediction time with a deep learning model, further time reduction was achieved by omitting the signed distance function for shape recognition that has a high computational load. The tool was also developed to maintain prediction accuracy even with a small amount of training data. Furthermore, output of fluid physical quantities required for verification on the tool operation screen was minimized, allowing performance of the process from input to output solely by button operations, making it easy to use for designers.

1. Introduction

Automobile side mirrors are parts used to understand the situation to the rear of the vehicle. As they are mounted on the exterior of the body, their part shape can disrupt the surrounding airflow at high speeds, potentially causing the mirror to vibrate and reduce rear visibility. Therefore, computational fluid dynamics (CFD) is used to predict performance and examine shapes to help minimize the potential impact of this flow. While the evolution of graphics processing units (GPU) continues to shorten CFD calculation times, it still requires around three days to calculate the flow velocity (three velocity components) around the side mirror, the surface streamlines, and the surface pressure distribution.

The side mirror shape affects the vibration of the mirror surface, and also impacts the styling. To achieve a good balance between functionality and styling, designers and styling designers should ideally be able to discuss matters on the spot. CFD calculations currently take time, however, making it necessary to take designs back and conduct performance predictions. Due to the inability to reconcile the optimal shape determined by designers as well as the design desired by styling designers during that period, inconsistencies in direction often lead to reworking, which becomes a bottleneck for shortening the development period.

To address this issue, it was thought that if there was a tool capable of performing prediction in a short time of around five minutes, designers and styling designers would

be able to repeat trial and error while viewing the prediction results, enabling determination of a shape with good balance during meetings.

Active research is underway on surrogate models based on deep learning as a method for performing prediction in a short time, and many reports have indicated its effectiveness in shortening prediction times⁽¹⁾.

Surrogate models using deep learning generally adopt an architecture based on CNN autoencoder (CNN-AE) that combines a CNN encoder and CNN decoder that use convolutional neural networks (CNN) as shown in Fig. 1⁽²⁾. However, CNN-AE is known to have the disadvantage that detailed information is lost as noise in the encoding (compression) process for feature extraction, resulting in blurred shape outlines. This is addressed by converting shape data to a signed distance function (SDF) for input to the CNN encoder. SDF is a function that provides geometric information about a shape and includes codes for calculating the distance from a certain point or determining which points are inside and which are outside the model

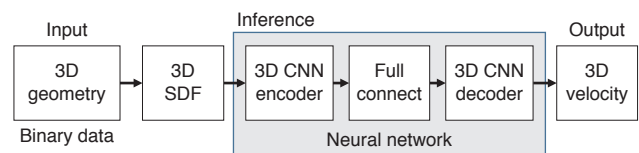


Fig. 1 Basic architecture of previous study

¹ Honda Motor Co., Ltd.

E-mail: takuro_hashimoto@jp.honda

² IDAJ Co., LTD.

surface. While this enhances accuracy, there is also the disadvantage that time is needed to calculate the distances for all points every time in both training and inference.

It is also generally known that data on the order of 10^3 is needed to provide sufficient accuracy in development of deep learning models, and Akasaka et al. conducted research with a similar condition⁽¹⁾. This presented a challenge for this study, as using CFD results to prepare this order of training data would take some five months.

An additional challenge is that the display method and operability need to be considered when designers use prediction tools. Since discussion time is limited, it is more important to be able to swiftly and easily display only the information needed for consideration, rather than confirming events from multiple perspectives.

To address these challenges, this study examined the following three points. First, to shorten the prediction time, the use of deep learning and the feasibility of omitting SDF were examined. Second, a deep learning model capable of providing prediction accuracy with a small amount of training data was investigated. Third, minimization of the outputs required for verification and an operation screen that enables operation solely by button operations were investigated to help prevent designer operations from becoming complicated.

2. Selection of Deep Learning Model and Development of Prediction Tool

To address the above challenges, the optimal deep learning model was selected and pre-checked, and a display screen that enables easy output by designers was investigated.

2.1. Shorter Prediction Time

To shorten the prediction time, this study also applied a surrogate model based on deep learning. Pix2Pix Generative Adversarial Networks (Pix2Pix GAN)⁽³⁾ was adopted as that deep learning model, and further speeding up by omitting SDF was also examined. Figure 2 shows the GAN that forms the basis of that algorithm. GAN is a generator training method proposed by Goodfellow et al.⁽⁴⁾, and consists of generator and discriminator neural networks. The generator generates data using random noise as the input, and the discriminator judges whether that data is real or fake. Training proceeds by having the generator and discriminator compete to enhance the quality of the generated data (adversarial training). As a result, GAN are characteristic in that training continues until sharp outputs are achieved, as opposed to CNN-AE, which may produce blurred shape contours.

An extension of GAN is the conditional generative adversarial network (conditional GAN: cGAN) shown in Fig. 3 proposed by Mirza et al.⁽⁵⁾ This enables specification of conditional inputs to the generator, making it possible to enhance the prediction accuracy by training a generator

with outputs corresponding to arbitrary inputs.

Pix2Pix GAN is a further extension of cGAN and is widely used as an image generation algorithm. By learning the relationship between a pair of images, it can generate another image that reflects that relationship based on a single image. 2D images are commonly used as conditional inputs, but to predict the spatial flow velocity (three velocity components) and pressure around a side mirror, the architecture is extended to three dimensions and STL-format 3D data is used instead of conditional images. While there is a case of using 3D expansion for brain tumor segmentation by Cirillo et al.⁽⁶⁾, there are no known cases of application to fluid prediction. Thus, a generator that inputs voxel data of shapes and outputs fluid physical quantities was used. Furthermore, a major feature of Pix2Pix GAN is the use of U-Net proposed by Ronneberger et al.⁽⁷⁾ as the generator. Figure 4 shows the architecture used in this study, which is an extension of the U-Net CNN layer to three dimensions. U-Net is an architecture consisting of an encoder and a decoder. Features extracted by the encoder

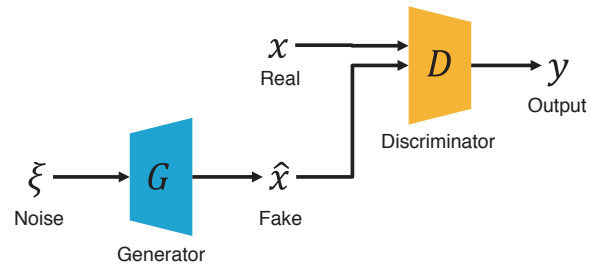


Fig. 2 GAN

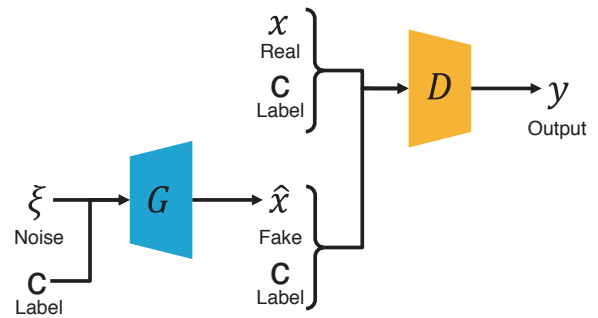


Fig. 3 Conditional GAN

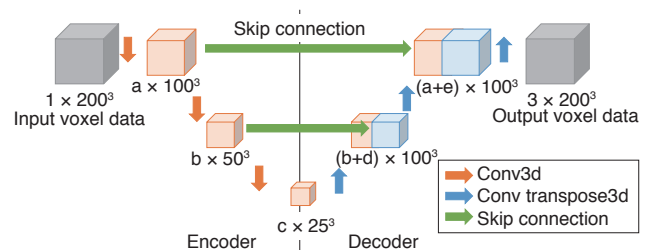


Fig. 4 Extension of U-Net to 3D

are restored to the original-size image by the decoder, and the skip connection between the encoder and decoder enables transmission of detailed information to the decoder layers of the same resolution without the loss of any data. With this architecture, spatial information and resolution details that tend to be lost by the encoder can be transmitted directly to the decoder, enabling retention of even fine features that might be lost in the CNN-AE encoding process. It was examined whether use of these GAN and U-Net would enable shortening of prediction times while maintaining prediction accuracy, even without use of SDF.

Figure 5 shows the architecture examined in this study. The inference part used a Pix2Pix GAN. A method that omits the SDF that was required before inference in CNN-AE and inputs the voxel representation of the shape data directly to the generator was examined while verifying the prediction accuracy.

2.2. Provision of Prediction Accuracy with a Small Amount of Training Data

Typically used CNN-AEs are trained with small data units called mini-batches, but as they are unable to consider the entire data distribution, a large amount of training data is needed to obtain high accuracy.

In contrast, Pix2Pix GAN is an effective deep learning model that can achieve high prediction accuracy even with a small amount of training data. The reasons given for this are that adversarial training enables the generative model to grasp the entire data distribution⁽⁴⁾, and the U-Net employed in the generator enables high-accuracy prediction. The combination of these factors enables training of the generator to achieve high prediction accuracy using less data than for CNN-AE.

This study set target values for prediction accuracy and the number of training data in order to verify the prediction accuracy with a small amount of training data. As the accuracy of CFD using Reynolds-averaged Navier-Stokes (RANS) is about $\pm 10\%$, the target prediction accuracy was set to a discrepancy of $\pm 10\%$ or less. The pressure coefficient C_p of the side mirror is in the range of ± 0.3 , so this target was set to ± 0.03 or less, and the target relative to the mainstream velocity of 38.8 m/s was set to ± 3.8 m/s (± 13.6 km/h). The target amount of training data was set to

Table 1 Deep learning conditions

	CNN-AE	Pix2Pix GAN
Number of training data	32,500	2,000
Using SDF	Yes	No

100 or less.

Accuracy was verified using the mean squared error (MSE) given by Eq. (1).

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (1)$$

Here, Y_i and $\hat{Y}_i$ respectively indicate the CFD results and the results estimated by the prediction tool, and n indicates the number of voxel points.

From this equation, the accuracy targets are an MSE of 750 or less for pressure, and 14.4 or less for velocity.

2.3. Verification of Accuracy Using 2D Images

In developing this tool, the validity of the Pix2Pix GAN prediction accuracy was verified in advance by comparing Pix2Pix GAN with the CNN-AE developed by Guo et al. 2D velocity distribution images calculated for various shapes using the general-purpose CFD software ANSYS Fluent by Synopsys Inc. were used as the training data. Table 1 shows the learning conditions. CNN-AE used 32,500 images as training data, and also converted them to SDF. Pix2Pix GAN used 2,000 images selected randomly from among those 32,500 images as training data, and omitted SDF. Figure 6 shows scooter-shaped binary data as an example of input in prediction, and the SDF calculated from that data. Figure 7 compares the prediction results of the flow velocity U_x component (horizontal direction in the figure) under these conditions. Both CNN-AE and Pix2Pix GAN produced images of the scooter shape and the velocity distribution

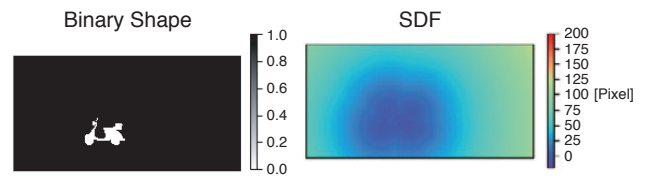


Fig. 6 2D input image

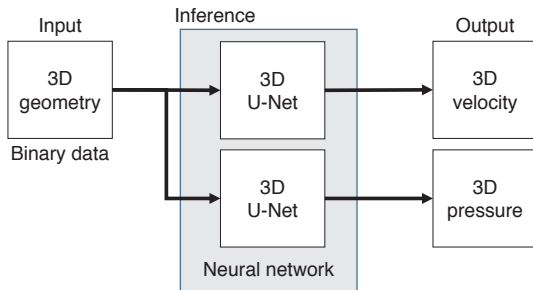


Fig. 5 Proposed architecture

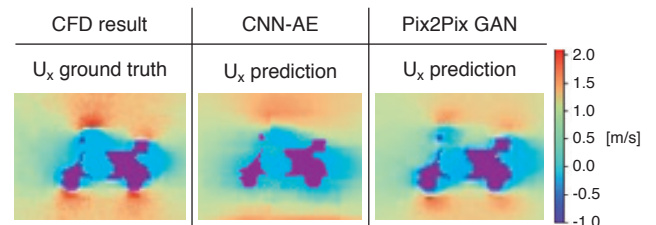


Fig. 7 Comparison between CNN-AE and Pix2Pix GAN

around the scooter with little blur similar to CFD. Both were also able to qualitatively reproduce the relative velocities and corresponding areas. This result indicated that by adopting Pix2Pix GAN instead of CNN-AE, the amount of training data can be reduced and the flow velocity near the body within the flow field can be predicted even without SDF.

2.4. Achievement of an Operation Screen Enabling Easy Output

Since the operation screen needs to be easy to operate for designers, it was designed so that all operations can be performed using only buttons. The buttons needed to perform all operations from loading the input STL data to output of the prediction results are arranged in order, enabling performance of all operations on a single screen. The outputs were narrowed down to the cross-sectional velocity distribution and the fluid information (surface pressure distribution, flow velocity distribution, and streamlines) for the side mirror surface shown in Fig. 8, making it easy to select and display the desired data using only button operations.

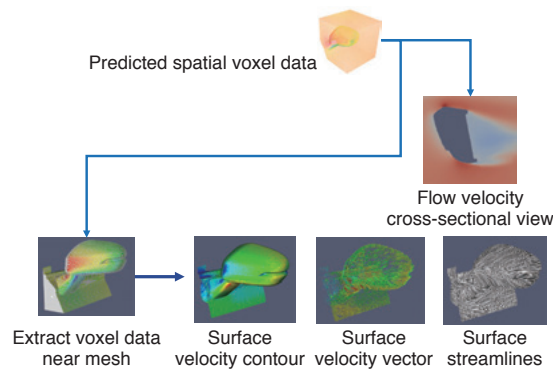


Fig. 8 Mapping from spatial distributions to surfaces

3. Training for Side Mirror Prediction

Training to predict the pressure and flow velocity distributions around the side mirror was conducted using the tool described in Chapter 2.

3.1. Training Process

Figure 9 shows the training process for spatial pressure distribution in the proposed Pix2Pix GAN model used in this study. First, the voxel representation of the side mirror shape was applied to the generator as conditional input data. Here, a pair consisting of the spatial pressure distribution predicted by that generator and the side mirror shape, and a pair consisting of the CFD result (true values) and the side mirror shape were generated. Next, the discriminator performed real/fake judgment using those two pairs as input, and then adversarial training was conducted based on

that judgment. Training was also performed similarly for the 3D spatial velocity distribution (X, Y, and Z directions). For reference, this model is capable of predicting spatial pressure and spatial velocity using only the generator when given the STL data of an unknown side mirror shape.

3.2. Generation of Training Data by CFD

The training data used CFD results for a vehicle equipped with side mirrors. Since application to various mirror sizes is assumed, the body with the largest mirror size was selected in order to encompass all mirror sizes. Eleven shape parameters representing the shape, position, and size features of the side mirror were set as shown in Fig. 10. Three levels were set for each shape parameter, and 101 cases of combinations were generated based on randomized design of experiments. Parametric data generation used the commercially available morphing tool DEP MeshWorks by Detroit Engineered Products, and CFD used ANSYS Fluent.

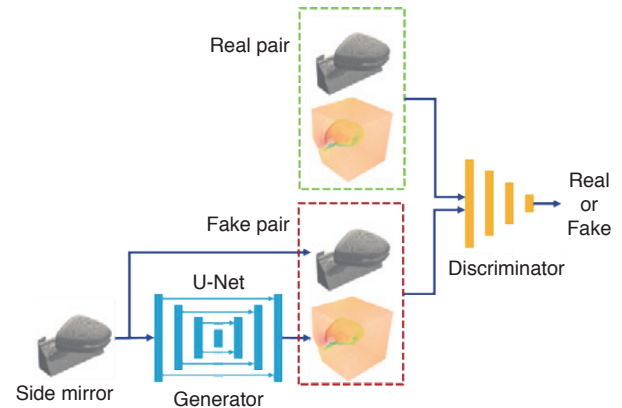


Fig. 9 Deep learning model for pressure distribution

No.	Parameters
1	Peak position (Horizontal direction)
2	Peak position (Vertical direction)
3	Inner plane angle
4	Outer plane angle
5	Leading edge angle
6	Mirror size
7	Length in mainstream direction
8	Mounting position (Rotation)
9	Mounting position (Mainstream direction)
10	Mounting position (Vertical direction)
11	Mounting position (Horizontal direction)

Fig. 10 CFD model creation for training data

Table 2 Mesh conditions

Number of mesh	50 million
Vehicle surface mesh size	4 mm
Circumference	8-16 mm
Boundary layer mesh	2 mm x 1 layer

Table 3 Analysis conditions

Turbulence model	Realizable k- ϵ
Speed	140 km/h (38.8 m/s)
Density	1.205 kg/m ³
Viscosity coefficient	1.8075e-5 Pa-s

Table 2 shows the CFD mesh conditions, and Fig. 11 shows the mesh placement. A hexahedral mesh was used as the overall base, and a single-layer boundary-layer mesh was positioned to reproduce only the surface shape. Table 3 shows the CFD analysis conditions. The turbulence model used RANS. Figure 12 shows an example of the CFD results.

3.3. Extraction of Training Data

Compared to the CFD results for the overall vehicle, the data volume was reduced, and speedup was achieved while enhancing the shape resolution of the side mirror model by narrowing down the prediction space that uses deep learning to the area around the side mirror. The specific prediction space that uses deep learning was a 400 mm cube encompassing the side mirror as shown in Fig. 13, and

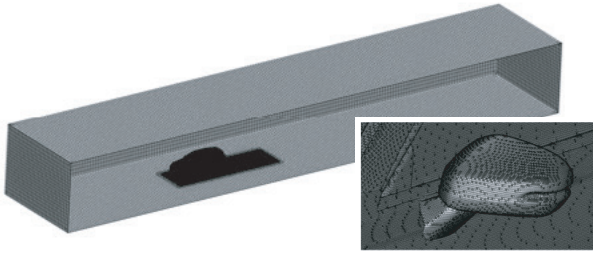


Fig. 11 Mesh placement

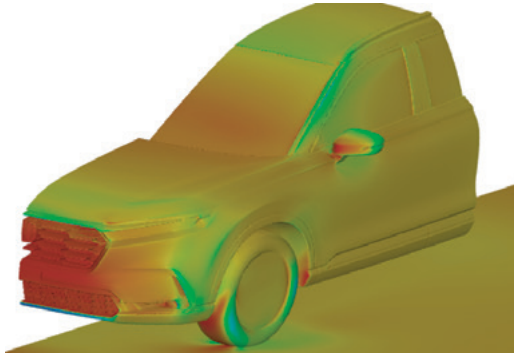


Fig. 12 Example of pressure distribution results using CFD

Table 4 Machine learning conditions

	Velocity	Pressure
Optimizer	Adams	Adams
Epoch	1000	1000
Batch size	1	1
Learning rate	2×10^{-4}	2×10^{-5}

voxels were arranged at 2-mm intervals within that cube. Conversion to the voxel data for the mirror shape shown in Fig. 14 used the vtkSelectEnclosedPoints algorithm from the open-source The Visualization Toolkit (VTK) library.

Figure 15 shows an example of a spatial pressure field used as training data. Conversion from CFD mesh to voxels was performed by substituting the average value of the 10 closest CFD mesh points to the center of each voxel.

3.4. Learning Conditions and Learning Time

Table 4 shows the learning conditions, with 80 of the prepared 101 cases used as training data. A single NVIDIA

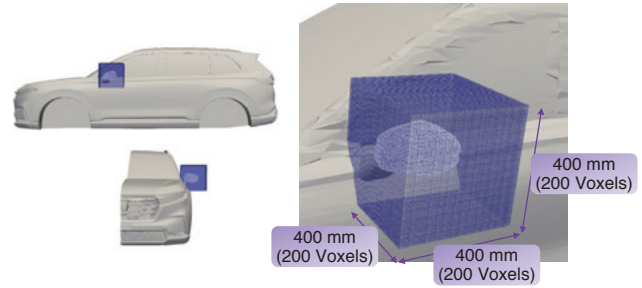


Fig. 13 Training data space for deep learning

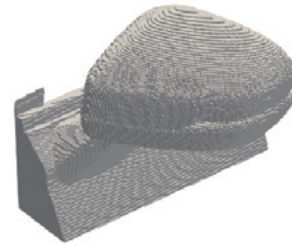


Fig. 14 Side mirror voxel model as noise and condition label

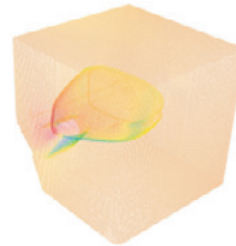


Fig. 15 Example of pressure distribution as training data

RTX A6000 GPU was used for training, and training took approximately 190 hours.

4. Verification of Prediction Tool Accuracy and Effect of Prediction Tool

Verification was conducted for the predicted results of spatial pressure distribution and spatial velocity distribution obtained using this tool, and the time-shortening effect when executed on the created operation screen was also confirmed.

4.1. Spatial Pressure Distribution Prediction Results

Figure 16 compares the CFD results and the surface pressure distribution obtained by projecting the inferred results of spatial pressure distribution onto the surfaces. Although the resolution decreases and unevenness appears in the pressure distribution due to the effects of voxelization, the locations where the pressure increases and decreases qualitatively matched the CFD results.

Figure 17 shows the MSE of the predicted spatial pressure distribution. The horizontal axis represents the ID number assigned to each mirror shape, and the MSE is plotted on the vertical axis. The white circles indicate the MSE for 80 cases of training data in the developed deep learning tool, and the black circles indicate the MSE for 21 cases of verification data. The MSE was about 160 for the overall training data, and 330 for the overall verification

data, both of which were within the accuracy target of 10% or less. However, the MSE for the most severe case was 841, which slightly exceeded 10%.

4.2. Spatial Flow Velocity Distribution Prediction Results

Figure 18 compares the CFD results and the surface flow velocity distribution and streamlines obtained by projecting the inferred results of spatial flow velocity distribution onto the surfaces. Mirror vibrations originate from the surface velocity distribution and the direction of the streamlines, which are impacted by factors such as flow

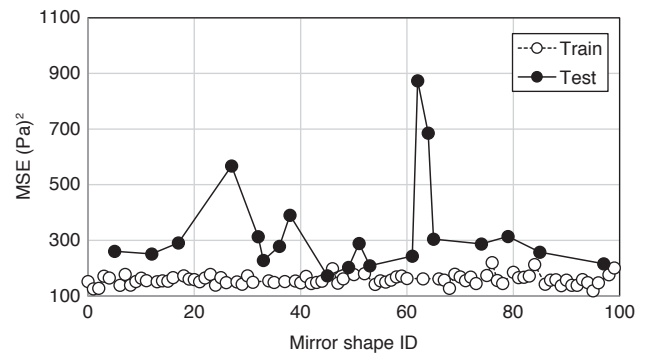


Fig. 17 MSE of inference of pressure distribution

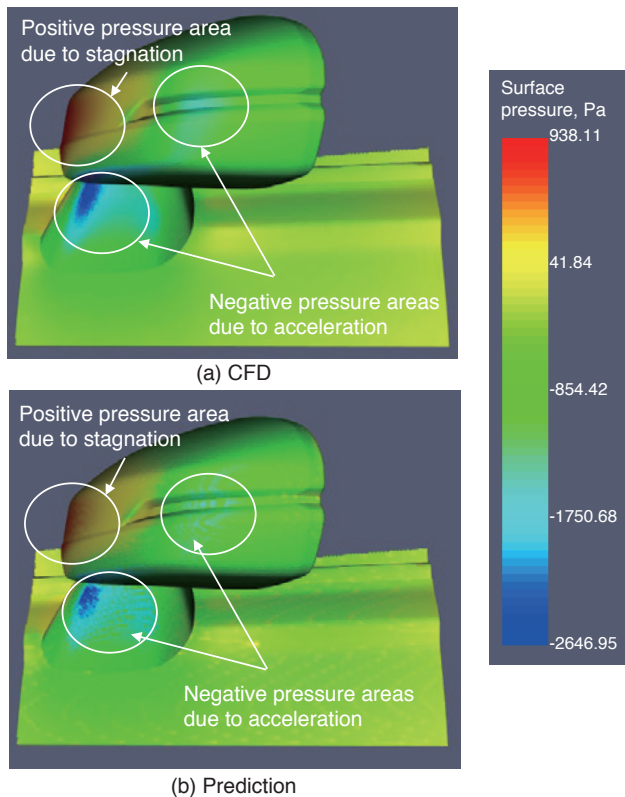


Fig. 16 Comparison of predicted and CFD pressure distribution

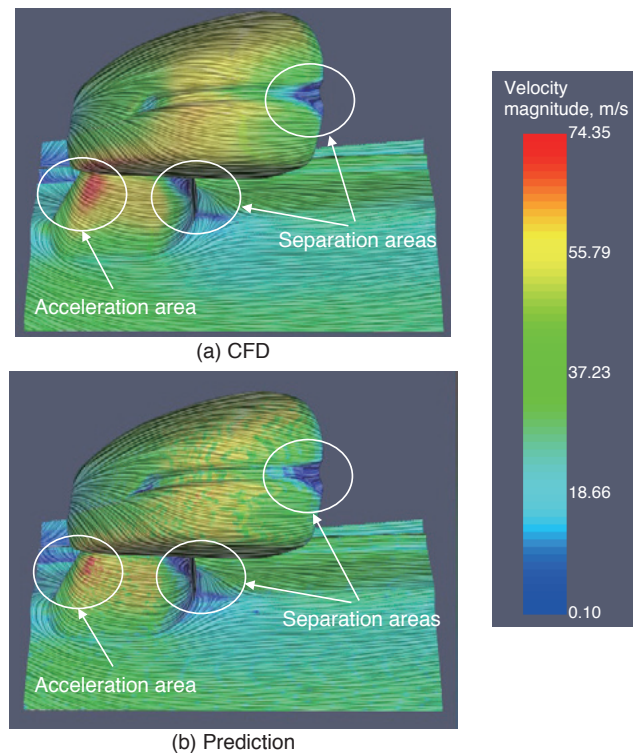


Fig. 18 Comparison of predicted and CFD velocity distribution

separation. From this perspective, the prediction results are generally consistent with the CFD results.

The above results show that the developed prediction tool provides sufficient accuracy even with 80 training data.

Figure 19 shows the MSE for the predicted spatial flow velocity distribution of each mirror shape. The MSE was about 0.1 for the overall training data (80 cases) and about 1.0 for the overall verification data (21 cases). Converted to velocity, the MSE for the overall verification data was about ± 1 m/s, which was within the accuracy target of 10% or less. The MSE for even the most severe case was 4.9 (velocity ± 1.1 m/s), which was also within 10% or less. These results show that sufficient accuracy can be provided when utilizing this tool, even with a small amount of training data.

4.3. Operation Screen

Figure 20 shows the operation screen created in this study. The buttons for the steps from loading the input STL to outputting the results are arranged in order, enabling operation on the same screen. The required output displays

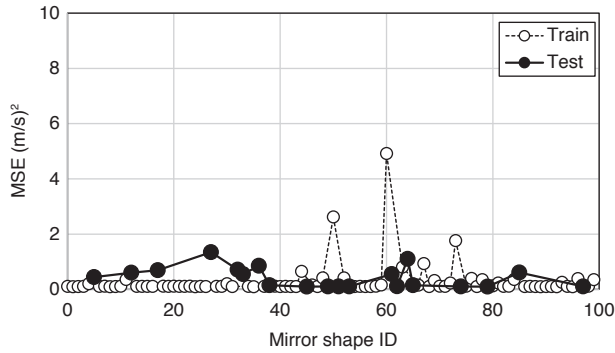


Fig. 19 MSE of inference of velocity distribution

Table 5 Computer specifications

	CFD	Deep learning
CPU	AMS EPYC7702	Intel Xeon w9-3475X 2.21GHz
Number of CPUs	128 core/node	1
GPU	-	NVIDIA RTX6000 Ada
Memory	512 GB/node	1 TB
OS	Red Hat Enterprise Linux release 8.3	Windows Server 2022

have been narrowed down, and all operations can be easily performed using only buttons.

4.4. Calculation Time

Table 5 shows the specifications of the equipment used for each calculation when evaluating the prediction tool time. Additionally, Fig. 21 shows the overall workflow of this prediction tool. Prediction work previously that took 52 hours with CFD could be performed in approximately 100 s.

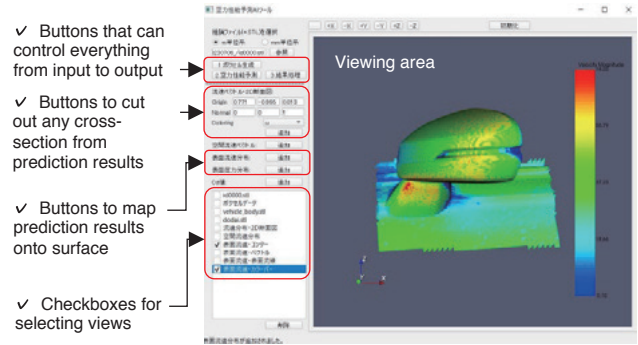


Fig. 20 Graphics of user interface

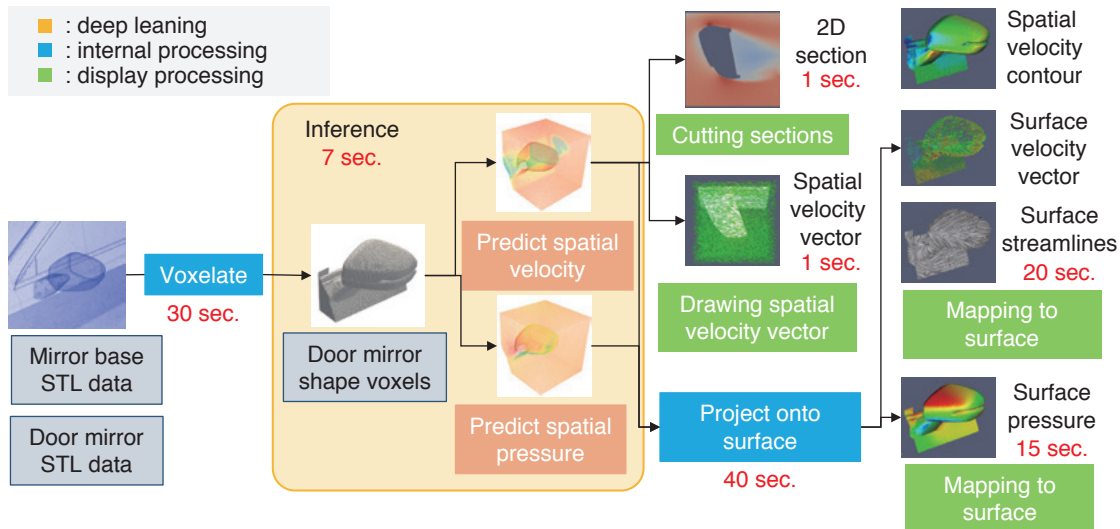


Fig. 21 Workflow of proposed system and total time

4.5. Summary

From the above, use of the developed prediction tool enables designers to make design decisions while exchanging opinions in real-time with styling designers.

Furthermore, continuing discussions will promote deeper exploration and refinement of ideas and stimulate both parties with differing perspectives, which can be expected to lead to the creation of various ideas.

5. Conclusion

This study used Pix2Pix GAN to develop a tool with the following characteristics for predicting flow fields around side mirrors.

- (1) In addition to shorter prediction times using a deep learning model, the contributions of the GAN and U-Net architectures, which are characteristics of Pix2Pix GAN, achieved further time reduction by omitting SDF, which has a high computational load. This enabled completion of the entire process from loading STL data to results display in approximately 100 s.
- (2) Generative models that use adversarial training are able to grasp the entire data distribution, enabling predictions with a smaller amount of data compared to conventional methods that are trained with mini-batches. As a result, predictions of pressure and flow velocity were within $\pm 10\%$ of the CFD results, even with the small amount of 80 training data.
- (3) By minimizing the display of fluid physical quantities required for verification and indicating the functions on the buttons, a display screen was created that enables designers to easily select all operations using only mouse controls.

This method is useful for users who find it challenging to collect training data, and for users who are about to begin introduction and development of such systems. It is anticipated that, going forward, this method will also be widely utilized in other prediction tools.

References

- (1) Akasaka, K., Chen, F., Teraguchi, T.: Surrogate Model Development for Prediction of Car Aerodynamics Using Machine Learning, Transactions of Society of Automotive Engineers of Japan, Vol. 52, Issue 3, p. 621-626, (2021) (in Japanese), doi: 10.11351/jsaeronbun.52.621
- (2) Guo, X., Li, W., Iorio, F.: Convolutional Neural Networks for Steady Flow Approximation, KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, p. 481-490, (2016), doi: 10.1145/2939672.2939738
- (3) Isola, P., Zhu, J-Y., Zhou, T., Efros, A. A.: Image-to-Image Translation with Conditional Adversarial Networks, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p. 5967-5976, (2017),

doi: 10.1109/CVPR.2017.632

- (4) Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks, Communications of the ACM, Vol. 63, Issue 11, p. 139-144, (2020), doi: 10.1145/3422622
- (5) Mirza, M., Osindero, S.: Conditional Generative Adversarial Nets, arXiv, (2014), doi: 10.48550/arXiv.1411.1784
- (6) Cirillo, M. D., Abramian, D., Eklund, A.: Vox2Vox: 3D-GAN for Brain Tumour Segmentation, Lecture Notes in Computer Science, Springer, Cham, Vol. 12658, p. 274-284, (2021), doi: 10.1007/978-3-030-72084-1_25
- (7) Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation, Lecture Notes in Computer Science, Vol. 9351, p. 234-241, (2015), doi: 10.1007/978-3-319-24574-4_28

■ Author ■



Takuro HASHIMOTO



Koji OKABE



Takeshi YAMATODA



Takafusa KIDO