

Machine Learning for Scanning Path Prediction in Laser Forming - Application of Structured Patterns and CNN -

Ping-Hsien Chou^{1,2}, Keiji Yamada^{*1}, Yean-Ren Hwang², Eisuke Sentoku³, Ryutaro Tanaka¹, and Katsuhiko Sekiya¹

¹Graduate School of Advanced Science and Engineering, Hiroshima University, Japan

²Department of Mechanical Engineering, National Central University, Taiwan (R.O.C)

³Department of Mechanical Engineering, National Institute of Technology, Fukui College, Japan

*Corresponding author's e-mail: keiji@hiroshima-u.ac.jp

For prototype design and small batch production, laser forming has more advantages for manufacturing metal sheet products than conventional processes. However, predicting and optimizing the required conditions for the product's final shape is difficult, especially in multi-stage forming, where many governing factors affect the shape intricately. In this study, the convolution neural network (CNN) is proposed to simulate the correlations between the scanning paths and the final deformations of a metal sheet. The imaginary data, which used values to present the distribution of deformation height of metal sheets, was used to examine the feasibility of applying CNN. On the other hand, the simulated images of a structured pattern projected on the sheet surface were used to train and test the CNN. The results demonstrate that CNN can use imaginary data in the training dataset to predict the scanning path determined by two points on the edge of the metal sheet with high accuracy. Although the performance of the test dataset needed to be better to prove the general-purpose ability, this research validates the feasibility of applying CNN for scanning path prediction in laser forming.

DOI: 10.2961/ilmn.2024.02.3001

Keywords: laser forming, convolution neural network, scanning path prediction, structured pattern, machine learning

1. Introduction

Laser forming is a process using thermal stress to deform a metal sheet without expensive dies and models. Due to the high flexibility and faster process [1-4], custom-made production, prototyping, and reshaping of formed products are fields where laser forming has the advantage.

With the extensive investigation of laser forming, many researchers have studied the relationship between deformation and process conditions [5-8]. Difficulty in controlling the process is attributed to that many governing factors change during the process. For example, repeatedly bending metal plates using laser forming increases the anisotropy of bending and accumulates the work hardening during consecutive stages. Furthermore, laser irradiation also causes thermal influences such as changes in metallographic structures, hardness, and absorptivity of the laser beam.

Machine Learning (ML) is a computer algorithm that extracts valuable knowledge called features from teaching data and effectively improves machining manufacturing [9-11]. A mathematical model builds in the ML algorithm based on sample data, known as "training data," to make predictions or decisions. And then, "test data" is used to prove that the accuracy and robustness of the model are suitable for the application. In recent years, ML has been frequently used to solve problems that only can be solved with a lot of experience or data.

In many ML methods, CNN can extract valuable features from images or matrices and be applied to classification or regression work. Based on the convenience of using images to predict and the powerful convolution ability to extract the features, CNN has been applied to many types of research in

mechanical engineering field. For example, Xuefeng et al. used CNN to develop a detection system that automatically recognizes tool wear types and obtains the flank wear width in the face milling process. It can collect the wear images of all the inserts of a face milling cutter during the machining gap, and no downtime for measurement [12]. Guo et al. proposed a novel hierarchical learning rate adaptive deep convolution neural network based on an improved algorithm, and applied it to diagnose the bearing faults and to identify both size and severity of faults in a bearing [13]. The results of experiments with bearing data demonstrated the superiority of the proposed ADCNN model to other fault-diagnosis methods, such as traditional deep CNN. Rifai et al. compared 5 kinds of loss functions for the convolutional neural network proposed to evaluate the surface roughness from the image of machined surface [14]. The average error of surface roughness prediction using CNN was verified to be within 10% of the actual measured surface roughness. At the same time, the computational time for estimation is within the desired range.

Thus, ML methods excel at recognizing patterns and extracting meaningful information from the amount of teaching data. The authors also applied a back propagation neural network (BPNN) to identify the condition of grinding wheel surface [15]. The grinding sound was recorded and used for training, and then appropriate dressing interval was estimated for Resinoid-bonded cubic boron nitride wheels. In addition, the corner wear of drill bit was predicted by learned model using BPNN, too [16]. Specially defined static and dynamic features were extracted from different frequency bands in frequency spectra of forces. Besides, ML

algorithms can effectively model complex nonlinear relationships between input and output variables, allowing for more accurate predictions and efficient strategies. Therefore, the authors also applied the neural network for forward prediction, where the metal sheet deformation was predicted from the process parameters in laser forming [16]. The forward prediction acts as a laser forming simulator giving the foresight of effects by process parameters and preventing possible problems.

This paper reported the results when the CNN method is applied to predicting laser scanning passes in multi-stage laser forming processing. The input and output data were the deformation change of a metal sheet after the processing and the sequential laser scanning pass during the processing, respectively. The pass has a significant impact on the processing results.

ML encompasses supervised, unsupervised, and semi-supervised learning, distinguishing whether the data has corresponding labels. The deformation data of this research is obtained through laser scanning paths or simulations. However, this study used deformation data for backward prediction for the scanning path. Therefore, each deformation data is labeled with its corresponding path. Consequently, this research opts for supervised learning.

2. Method

This research used the metal sheet's shape as teaching data to predict the laser scanning path. However, the plate shape measuring to collect the deformational distribution takes too much time, generally. Therefore, in the present paper, 'Imaginary data' deductively expected distributions of deformation within metal sheets were used for each given irradiating conditions of laser [17]. Specifically, the imaginary data was deduced by the superposition of plate bending

Table 1 Process conditions.

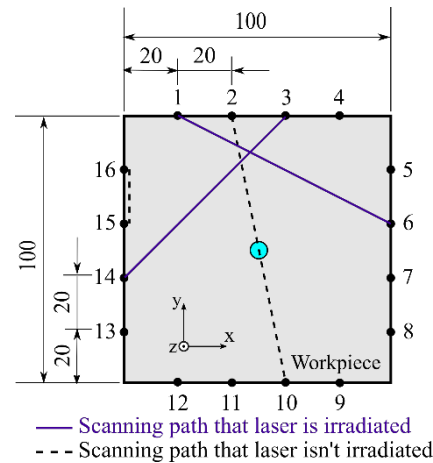
Laser forming parameter	
Number of stages	4
Number of scans per stages	4
Bending angle per stage [°]	4
Workpiece	
Material	JIS SS400
Size [mm]	100 × 100 × 1.5
Diode laser	
Average power [W]	220.9±1.5
Scan speed [mm/min]	800
Spot diameter <i>d</i> [mm]	1.7

consecutively generated in forming stages with the particular values of average laser power, scanning speed, and spot diameter shown in Table 1. Thus, this simulation assumes that the bending angle $\theta = 4^\circ$ is determined without influences of the gradual change in plate rigidity by previous stages. Using simulated data is expected to give enough plate shape diversity to acquire high versatility in path prediction. The feasibility of applying CNN was examined through the training with imaginary data and the verification of predicted paths for target shapes.

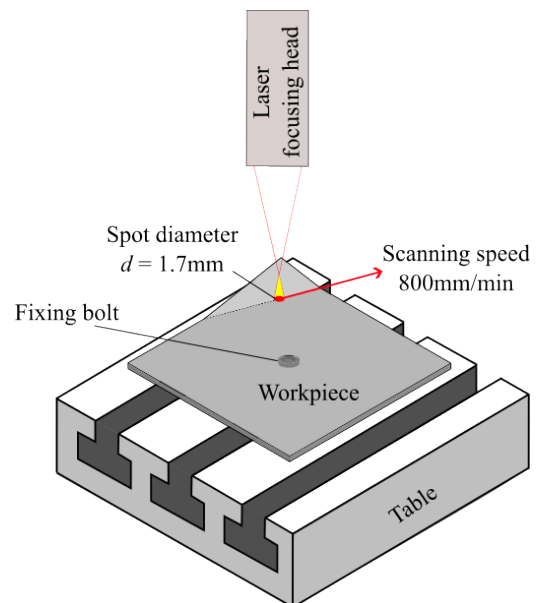
2.1 Collecting data on laser forming

For simplification in the feasibility study, forming stages are limited to 4, and the laser conditions are shown in Table 1. Scanning paths are randomly chosen to acquire general predictability for various shapes caused by superposing consecutive bending along those paths. In addition, the laser beam is scanned 4 times (= 2 reciprocating on a path) at a single stage of forming to obtain adequately large deformation.

As shown in Fig. 1 (a), two endpoints were randomly chosen from discrete 16 candidate points to consider as each straight scanning path. Those candidates were arranged with even spacing of 20 mm on each edge, and two endpoints had not to be chosen on the same edge to avoid making the path overlaid the edge. In addition, lines interfering with the fixing bolt were also excluded. For example, solid lines <1-6> and <3-14> are the correct paths, while broken line <2-10> is not chosen because of the interference with the fixing bolt, as shown in Fig. 1 (b).



(a) Simulated multiple scanning paths



(b) Experimental apparatus on NC table

Fig. 1 Experimental procedure in laser forming tests.

In the laser forming experiment, a laser distance meter was used to sample the surface of the workpiece and then get the height of each sampling point after each forming stage by simple calculation. Therefore, the simulation also built enough imaginary height distribution data, like Fig. 4 (a), to train the machine learning model [17]. Additionally, CNNs excel at extracting features from images. This study aims to introduce a novel optical data collection method by directly capturing images of the workpiece or projecting particular patterns onto the workpiece surface and inputting them to the CNN for feature extraction to predict the laser scanning path. This research tried to validate the feasibility of path prediction using imaginary images with particular patterns at first. The imaginary images with particular patterns will be introduced in Chapter 2.3.

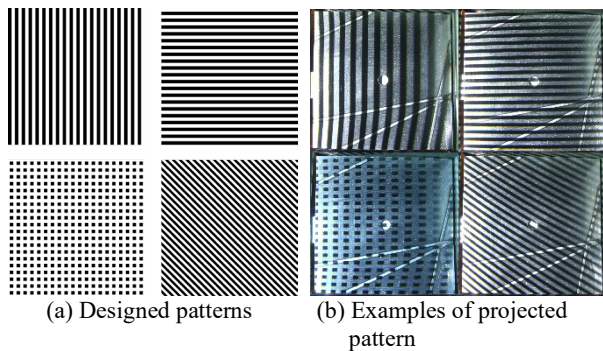


Fig. 2 Four different structured patterns used.

Regularly repeated geometric patterns enable the detection of distortion of a surface on which the patterns are projected. In three-dimensional measurement, using fringe projection techniques for generating surface information is one of the most active research areas in optical metrology [18-20]. Most of these methods use sinusoidal fringe patterns, and phase analysis is conducted to obtain the surface information of the target, including the human face, apple, and any 3-D diffuse objects. However, this research proposed an idea that applied the binary fringe pattern in different directions, as shown in Fig. 2(a), and used the projected patterns of the metal surface through CNN to predict irradiation conditions. In contrast to the reference paper, this study employed binary fringe patterns instead of sinusoidal ones, and the phase analysis was not considered to obtain surface deformation. This research tried to apply CNN to extract the features of bending lines on the workpiece surface and predict the scanning path. In order to ensure omnidirectionality in scanning path prediction, designed patterns had varying

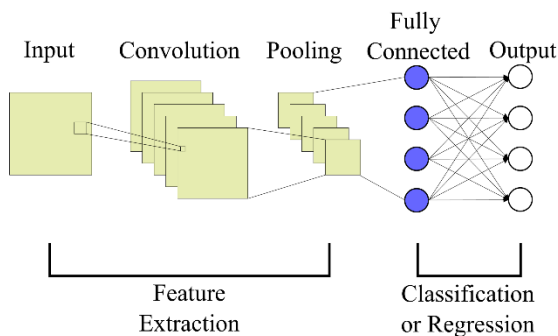


Fig. 3 Structure of convolution neural network used.

orientations. To avoid situations where the fringe patterns do not deviate when parallel with the scanning path, potentially affecting the CNN from extracting helpful features in that region. Figure. 2(b) shows the patterns projected onto the metal plates bent during laser forming tests. However, capturing many of these images presents challenges due to two disturbance influences. Firstly, the clarity of the fringe patterns on the workpiece's surface is susceptible to variations in lighting conditions, including both the projector's light source and ambient lighting. Maintaining consistent illumination is crucial to ensure the quality of captured images. Secondly, the relative positions of the workpiece and the camera cannot be perfectly fixed between each machining operation and image capture step. This variability complicates following data processing steps, requiring careful calibration and registration procedures to reduce these positional variations.

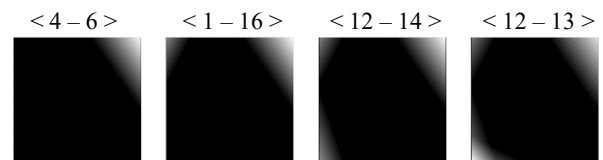
Nevertheless, the present study aims to demonstrate the efficacy of proposed path prediction method using CNN. Therefore, the authors considered that inputting those captured photos directly to the CNN should be continuously investigated in our future study, rather than included prematurely in the present study. Instead, imaginary data of the projected patterns was simulated on the assumption that influences of any disturbance were removed.

2.2 Convolution neural network and datasets

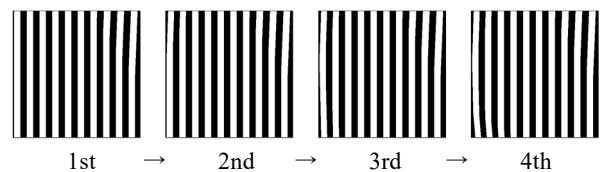
In order to predict laser scanning paths from the sheet shape, machine learning was conducted using a CNN shown in Fig. 3. Structure of the CNN contains two parts: convolutional layers for extraction of features from the input, and fully connected layers for extraction of features in shapes of calculating the prediction by imitating the nonlinear function.

Training and testing of the CNN are conducted with teaching datasets consisting of the matrix data for input and the labels for output. 1000 datasets were prepared in this study, 80% were used for training CNN, and the rest were for testing. These data were obtained by deductive reasoning so that they could be obtained in a shorter time than the experiment and exclude disturbance influences.

Tables 2 to 7 list all the convolution neural network structures. These listed networks have not been fully optimized, and the author believes there is still room for adjustment in each network. Each training for the CNN involves



(a) Gray images presenting height distribution



(b) Vertical fringes converted from upper gray images
Fig. 4 Transition on plate shape during four stages.

loading data and training the network for 1-4 forming stages, which consumes a substantial amount of time. While not optimal, the results obtained in this experiment are generated through training with the respective network architectures. The training parameters not explicitly listed are all default ones of TensorFlow. All training used the Adam optimizer in this study.

2.3 Input data for CNN

As mentioned in Section 2.1, the imaginary data contains:
 1. Height distribution data is used to simulate the data collected by the laser distance meter.

Table 2 CNN structure of Height distribution input, input data scheme A, and binary encoding method.

	First stage	Second stage	Third stage	Fourth stage
Input	$400 \times 400 \times 1$	←	←	←
Conv2D + ReLu	(5×5) $396 \times 396 \times 8$	(3×3) $398 \times 398 \times 8$	←	←
MaxPool	$198 \times 198 \times 8$	$199 \times 199 \times 8$	←	←
Con2D + ReLu		(3×3) $197 \times 197 \times 8$	(3×3) $197 \times 197 \times 16$	←
MaxPool		$98 \times 98 \times 8$	$98 \times 98 \times 16$	←
Con2D + ReLu		(3×3) $96 \times 96 \times 8$	(3×3) $96 \times 96 \times 32$	←
MaxPool		$48 \times 48 \times 8$	$48 \times 48 \times 32$	←
Con2D + ReLu		(3×3) $46 \times 46 \times 8$	(3×3) $46 \times 46 \times 32$	←
Flatten	313632	33856	67712	←
Dense + ReLu	32	1280	←	←
Dense + ReLu		320	←	640
Sigmoid	16	32	48	64

Table 3 CNN structure of Height distribution input, input data scheme A, and Cartesian encoding.

	First stage	Second stage	Third stage	Fourth stage
Input	$400 \times 400 \times 1$	←	←	←
Con2D + ReLu	(3×3) $398 \times 398 \times 4$	(5×5) $396 \times 396 \times 8$	(5×5) $396 \times 396 \times 4$	(5×5) $396 \times 396 \times 8$
MaxPool	$199 \times 199 \times 4$	$198 \times 198 \times 8$	$198 \times 198 \times 4$	$198 \times 198 \times 8$
Con2D + ReLu	(3×3) $197 \times 197 \times 8$	(5×5) $194 \times 194 \times 8$	←	(5×5) $194 \times 194 \times 16$
MaxPool	$98 \times 98 \times 8$	$97 \times 97 \times 8$	←	$97 \times 97 \times 16$
Con2D + ReLu		(5×5) $93 \times 93 \times 8$	(5×5) $93 \times 93 \times 16$	(5×5) $93 \times 93 \times 64$
MaxPool		$46 \times 46 \times 8$	$46 \times 46 \times 16$	$46 \times 46 \times 64$
Con2D + ReLu				$42 \times 42 \times 64$
Flatten	36864	16928	33856	112896
Dense + ReLu	128	2000	←	←
Dense + ReLu	64	2000	320	←
Dropout			50%	←
Sigmoid	4	8	12	16

Table 4 CNN structure of Height distribution input, input data scheme B, and binary encoding.

	First stage	Second stage	Third stage	Fourth stage
Input	$400 \times 400 \times 1$	$400 \times 400 \times 2$	$400 \times 400 \times 3$	$400 \times 400 \times 4$
Con2D + ReLu	(5×5) $396 \times 396 \times 8$	←	(5×5) $396 \times 396 \times 4$	←
MaxPool	$198 \times 198 \times 8$	←	$198 \times 198 \times 4$	←
Con2D + ReLu		(5×5) $194 \times 194 \times 8$	←	←
MaxPool		$97 \times 97 \times 8$	←	←
Con2D + ReLu		(5×5) $93 \times 93 \times 8$	(5×5) $93 \times 93 \times 16$	←
MaxPool		$46 \times 46 \times 8$	$46 \times 46 \times 16$	←
Flatten	313632	16928	33856	←
Dense + ReLu	32	2000	←	←
Dropout				50%
Dense + ReLu			320	←
Dropout				50%
Sigmoid	16	32	48	64

Table 5 CNN structure of Height distribution input, input data scheme B, and Cartesian encoding.

	First stage	Second stage	Third stage	Fourth stage
Input	$400 \times 400 \times 1$	$400 \times 400 \times 2$	$400 \times 400 \times 3$	$400 \times 400 \times 4$
Con2D + ReLu	(3×3) $398 \times 398 \times 4$	(5×5) $396 \times 396 \times 8$	←	(5×5) $396 \times 396 \times 4$
MaxPool	$199 \times 199 \times 4$	$198 \times 198 \times 8$	←	$198 \times 198 \times 4$
Con2D + ReLu	(3×3) $197 \times 197 \times 8$	(5×5) $194 \times 194 \times 8$	←	←
MaxPool	$98 \times 98 \times 8$	$97 \times 97 \times 8$	←	←
Con2D + ReLu		(5×5) $93 \times 93 \times 8$	←	(5×5) $93 \times 93 \times 16$
MaxPool		$46 \times 46 \times 8$	←	$46 \times 46 \times 16$
Flatten	36864	16928	←	33856
Dense + ReLu	320	2000	←	←
Dropout			50%	←
Dense + ReLu	64	2000		
Sigmoid	4	8	12	16

2. Fringe image data is used to simulate data of specific patterns projected onto the workpiece surface.

Figure 4(a) gives height distributions visualized by grayscale images converted from the vertical deformation. The grayscale image enables easy observation of the change in sheet shape through forming stages. Figure 4(b) shows another type of input data for CNN: imaginary photo images simulated as if the pattern is projected with an incident angle

of 45° . Each image corresponds to the height distribution in Fig. 4(a).

Because the bending angle in laser forming is small, multi-stage forming is required to obtain complex shapes or to draw deeply. Target shapes at each stage are provided by process-design work, and the difference between two consecutive shapes is that the deformation ought to be gained at each stage. This study also tried different sizes of input data and investigated the influences of that. Table 8 shows two

Table 6 CNN structure of fringe image input, input data scheme B, and binary encoding.

	First stage	Second stage	Third stage	Fourth stage
Input	$400 \times 400 \times 1$	$400 \times 400 \times 2$	$400 \times 400 \times 3$	$400 \times 400 \times 4$
Con2D + ReLu	(5×5) $396 \times 396 \times 8$	←	(5×5) $396 \times 396 \times 4$	←
MaxPool	$198 \times 198 \times 8$	←	$198 \times 198 \times 4$	←
Con2D + ReLu		(5×5) $194 \times 194 \times 8$	←	←
MaxPool		$97 \times 97 \times 8$	←	←
Con2D + ReLu		(5×5) $93 \times 93 \times 8$	(5×5) $93 \times 93 \times 16$	←
MaxPool		$46 \times 46 \times 8$	$46 \times 46 \times 16$	←
Flatten	313632	16928	33856	←
Dense + ReLu	128	2000	←	←
Dropout			50%	←
Dense + ReLu		2000	320	←
Dropout			50%	←
Sigmoid	16	32	48	64

Table 7 CNN structure of fringe image input, input data scheme B, and Cartesian encoding.

	First stage	Second stage	Third stage	Fourth stage
Input	$400 \times 400 \times 1$	$400 \times 400 \times 2$	$400 \times 400 \times 3$	$400 \times 400 \times 4$
Con2D + ReLu	(5×5) $396 \times 396 \times 4$	←	(5×5) $396 \times 396 \times 8$	(5×5) $396 \times 396 \times 4$
MaxPool	$198 \times 198 \times 4$	←	$198 \times 198 \times 8$	$198 \times 198 \times 4$
Con2D + ReLu	(5×5) $194 \times 194 \times 8$	←	←	←
MaxPool	$97 \times 97 \times 8$	←	←	←
Con2D + ReLu	(5×5) $93 \times 93 \times 16$	←	(5×5) $93 \times 93 \times 8$	(5×5) $93 \times 93 \times 16$
MaxPool	$46 \times 46 \times 16$	←	$46 \times 46 \times 8$	$46 \times 46 \times 16$
Flatten	33856	←	16928	33856
Dense + ReLu	640	←	2000	←
Dropout			50%	
Dense + ReLu	64	←		320
Dropout				50%
Sigmoid	4	8	12	16

schemes to prepare the data. In Scheme A, only the present target shape, meaning the formed workpiece shape, was given in the data for any number of stages. For example, for the path prediction during 3-stage forming, paths in all stages were predicted from the shape just after the third stage, as shown on the left side of Table 9. As indicated in Table 8, the dimension of input data arrays was the same regardless of the number of stages. So, the CNN is expected to use less model memory footprint in training progress.

On the other hand, in Scheme B, the input data included all target shapes, which are consecutively obtained at each stage. For example, target shapes at the first, second, and third stages were used to predict scanning paths during 3-stage forming, as shown on the right side of Table 9. As indicated in Table 8, the dimension of input data arrays increased proportionally to the number of stages. High prediction accuracy and adaptability for complex shapes are expected because the shape change of each stage is included in

the input data. Due to space constraints, complete explanatory diagrams for all stages 1-4 are not provided here. Therefore, only schematic representations for stages 2 and 3 are presented.

2.4 Output data for CNN

The output data for CNN was a vector (an array of variables) to which the laser scanning path positions were encoded. For encoding these paths, two methods were employed, and their prediction performance was compared. Firstly, as the ‘binary encoding method,’ a 16-digit binary number was made by putting a flag on two digits corresponding to two endpoints of the scanning path. For example, to present path <1-6>, the values in index 1 and 6 of the array were 1, and the others were set as 0 (Fig. 5(a)). Therefore, the encoded data represents a distribution of certainty in the

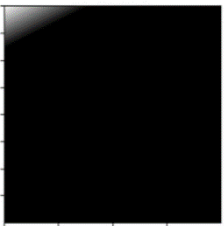
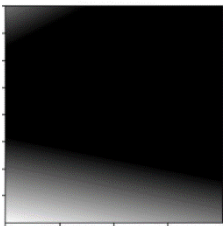
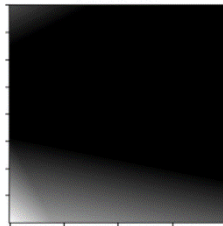
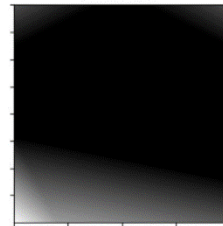
classification problem. The sigmoid function was used as the activation function from fully connected layer to output layer, and a BCE (binary cross entropy) was employed as the loss function. The following equation gives the BCE for the binary encoding, which calculates the difference, L_{BCE} , between a correct path and a predicted path.

$$L_{BCE} = \frac{1}{N_d} \sum_{d} l_{BCE}$$

$$= \frac{1}{N_d} \frac{-1}{N_o} \sum_{i} \sum_{j} \{y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)\}, \quad (1)$$

Table 8 Input data given for CNN.

Scheme A: Target shape at present stage is used as input for CNN

Number of stages n	1	2	3	4
Input data				
Dimension of data	400×400×1	400×400×1	400×400×1	400×400×1

Scheme B: Target shapes before present stage are used as input for CNN

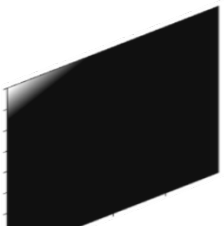
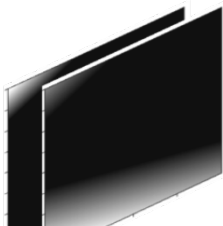
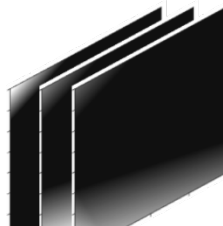
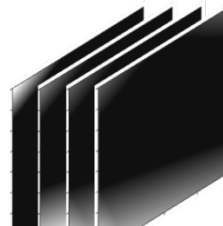
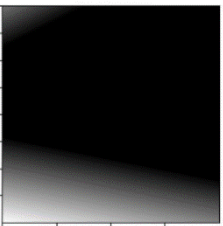
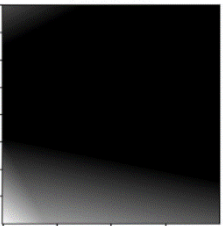
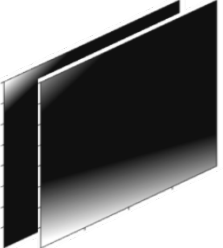
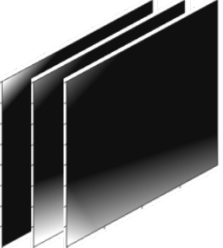
Number of stages n	1	2	3	4
Input data				
Dimension of data	400×400×1	400×400×2	400×400×3	400×400×4

Table 9 Different training conditions in 2nd and 3rd forming stage.

Scheme	A		B	
	Second	Third	Second	Third
Input				
Output	Two scanning paths	Three scanning paths	Two scanning paths	Three scanning paths

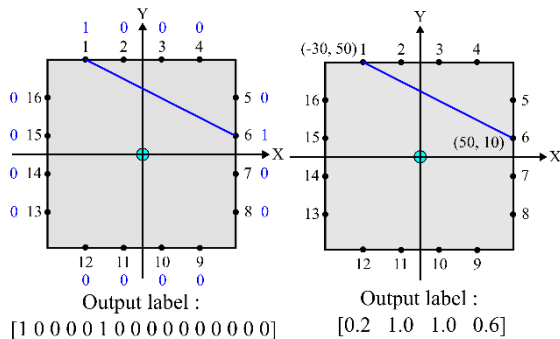
where the size of the output label is $N_o=16n$ (n : the number of stages), N_d is the number of datasets, y_i is the correct path in the teaching data, and \hat{y}_i is the path predicted by CNN.

Secondly, as ‘Cartesian encoding method’, the output was values of Cartesian coordinates x and y for each of the two endpoints of the path. In addition, normalization was employed for the coordinates values to avoid loss diverges in training progress. For example, to present path <1-6>, the normalized value in output (0.2, 1.0, 1.0, 0.6) was converted from the actual coordinates (-30, 50, 50, 10), as shown in Fig. 5(b). Therefore, the encoded coordinates are similar to the dependent variable in the regression problem. The Sigmoid function was used as the activation function from fully connected layer to output layer, and a GD (geometric distance) defined in equation (2) was employed as the loss function. The loss function was a mean geometric distance between a correct and predicted path, L_{GD} . In this study, it was defined as the mean value of length from the endpoint of the correct path to the endpoint of a corresponding predicted path.

$$L_{GD} = \frac{1}{N_d} \sum l_{GD}$$

$$= \frac{1}{N_d} \sum \sum (S_{pred}S_{cor} + G_{pred}G_{cor}), \quad (2)$$

where n is the number of stages, and S and G represent the endpoint of a path as shown in Fig. 6, and subscripts ‘cor’ and ‘pred’ represent correct and predicted path, respectively.



(a) Binary encoding (b) Cartesian encoding
Fig. 5 Two encoding methods of scanning path.

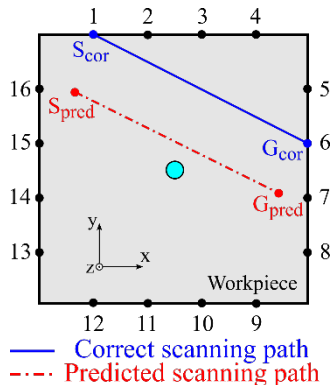


Fig. 6 End points of laser scanning paths chosen and predicted.

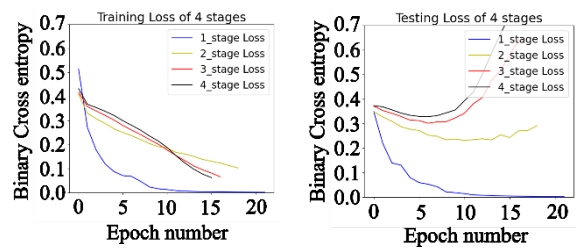
3. Results and Discussions

This research applied different methods to examine the feasibility of CNN, including two encoding methods and different sizes of input data. As shown in Table 10, the height distribution of metal sheet and imaginary fringe image were two kinds of input types of CNN. The label of the output of CNN was given by two encoding methods: binary encoding and Cartesian encoding. CNN was employed as a machine learning tool for analyzing different sizes of input data as shown in Table 8. These results are introduced in the following sections.

3.1 Prediction results by means of height distribution

Figure 7 shows the transition of loss function as training by scheme A for the path prediction in multi-stage (n up to 4). The input data was height distributions with a size of $(400 \times 400 \times 1)$, and the scanning paths were encoded with the binary encoding method. The loss function for binary encoding, the binary cross entropy L_{BCE} decreased and converged to a small value for the training datasets. It should be noted that the training was completed for a very short period compared to the previous study [17]. The same dataset was used to train CNN and a fully connected network with the binary encoding method. Figure 9 shows the training loss trend of the first forming stage. It can be observed that the number of epochs and training time required for CNN is significantly lower than that of fully connected network. Nevertheless, the binary cross entropy L_{BCE} converged only in single stage forming for the test datasets, while it diverged in multi-stage forming.

For considering the influences of input data preparation procedure, Fig. 8 shows the transition of binary cross entropy L_{BCE} as training by scheme B where the size of input data was $(400 \times 400 \times n)$. There was no significant difference for training datasets when comparing Figs 7(a) and 8(a). In contrast, comparing Figs 7(b) and 8(b), the binary cross entropy L_{BCE} was obviously improved by scheme B, while it was still large values for larger forming stages. This suggests that scheme B acquired higher versatility than scheme A. The detailed loss trends of each stage are shown in Figs 18 to 21 on the final page.

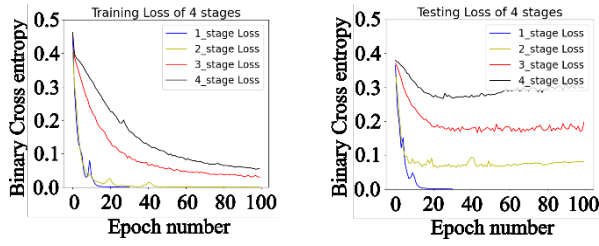


(a) For training datasets (b) For test datasets

Fig. 7 Trends of binary cross entropy L_{BCE} in training process by scheme A.

Table 10 Different conditions of network training.

Input	Network	Output
Height distribution	Convolution neuronal network	Binary encoding
Imaginary fringe image		Cartesian encoding



(a) For training datasets (b) For test datasets

Fig. 8 Trends of binary cross entropy L_{BCE} in training process by scheme B.

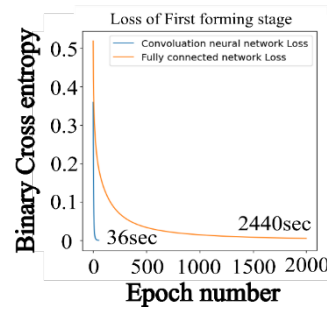


Fig. 9 Trends of binary cross entropy L_{BCE} of two different neural networks.

The performance of prediction using the completely trained CNN is presented in Fig. 10. Figure 10(a) contains the bar charts illustrating distributions of the endpoints of correct laser scanning paths against the ones of paths predicted by the trained CNN shown in Fig. 7. On the other hand, charts in Fig. 10(b) were obtained using the CNN trained as shown in Fig. 8. In all bar charts, two axes on the bottom represent the candidate points chosen as endpoints of the correct and predicted paths. Each tile on the bottom corresponds to a combination of the correct and predicted endpoints. At the same time, the height of the bar graph indicates the frequency of the corresponding endpoint. Therefore, if the prediction coincides with the correct data, tall bars stand on a diagonal of the square bottom. On the

contrary, difficulty in the path prediction is presented as dispersion of the frequency distribution. Because 80% of the whole datasets were used for training, the number of candidate points was $800 \times 2 \times n$, where two endpoints in each scanning path, and n is the number of stages. Therefore, the upper limit of the scale vertical axis increases with the number of stages, but the frequency distribution can be compared between those charts.

As shown in all charts of Figs. 10(a) and 10(b), the prediction exhibited high accuracy, but scheme B seems slightly better than scheme A. Based on this finding, it is believed that scheme B, which contains the target shape of each stage, helps to predict the sequence of scanning paths and is more accurate. For a detailed comparison, Fig. 10 also includes a

Number of stages n	1	2	3	4
(a) Scheme A				
Correct prediction rate	$\frac{\text{Correct predicted}}{\text{whole predicted}} = 100\%$	87.32 %	83.54 %	83.23 %
(b) Scheme B				
Correct prediction rate	$\frac{\text{Correct predicted}}{\text{whole predicted}} = 100\%$	100 %	100 %	100 %

Fig. 10 Frequency distribution of predicted endpoints against ones for training datasets (in binary encoding, epoch for training ≤ 100).

correct prediction rate (=correctly predicted paths / whole predicted paths). As can be seen, it is obvious that scheme B achieved higher prediction accuracy.

Test datasets are used to evaluate and assess the model's performance and versatility. Figure 11 shows the performance of prediction obtained by test datasets. The number of candidate points is $200 \times 2 \times n$, where the test datasets number is 200 and n is the number of stages. Scheme A had a high dispersion of prediction for the multi-forming stages in Fig. 11(a), but scheme B maintained its impressive performance in Fig. 11(b). According to these results, effective and reliable path prediction was achievable in the initial two forming stages, as the input data contained target shapes at each stage. However, it is noteworthy that in the third and fourth forming stages, as illustrated in Fig.

11(b), overfitting issues were observed, indicating the need for resolution. This result suggests that further optimization and addressing overfitting challenges are essential for extending the effectiveness of path prediction across all forming stages.

Figures 12(a) and 12(b) show the transition of loss function for results using the input data prepared by schemes A and B, respectively. The laser scanning paths were encoded by the Cartesian method. Geometrical distance has a length dimension, and the value can be easily compared with other quantities. In both Fig. 12(a) and 12(b), it is confirmed that the loss functions converged for tens of epochs in the training process. In the transition of L_{GD} for test datasets, though the divergence was not observed, the distance values were still large, especially in higher stages. This result means the

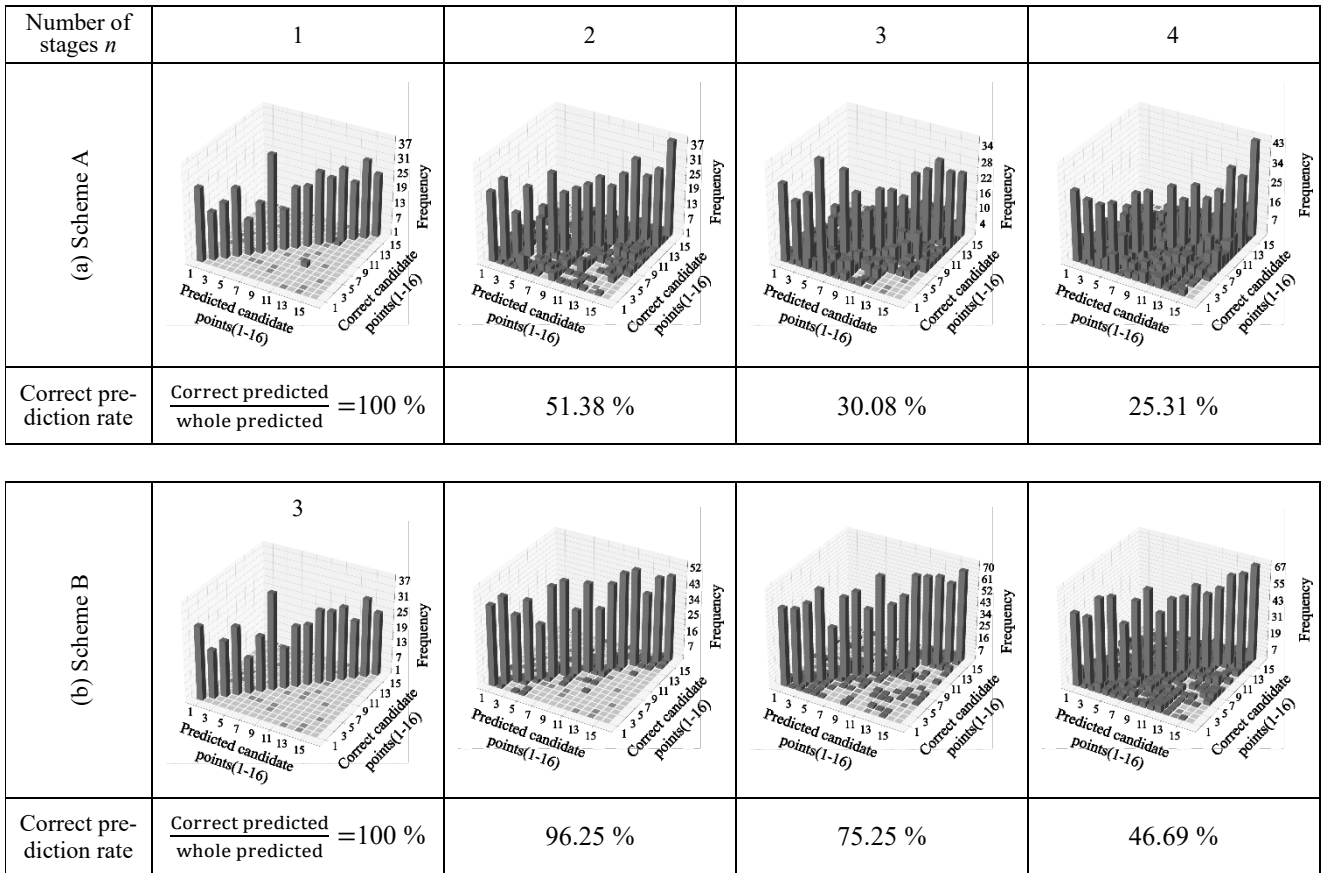


Fig. 11 Frequency distribution of predicted endpoints against ones in test datasets (in binary encoding, epoch for training ≤ 100).

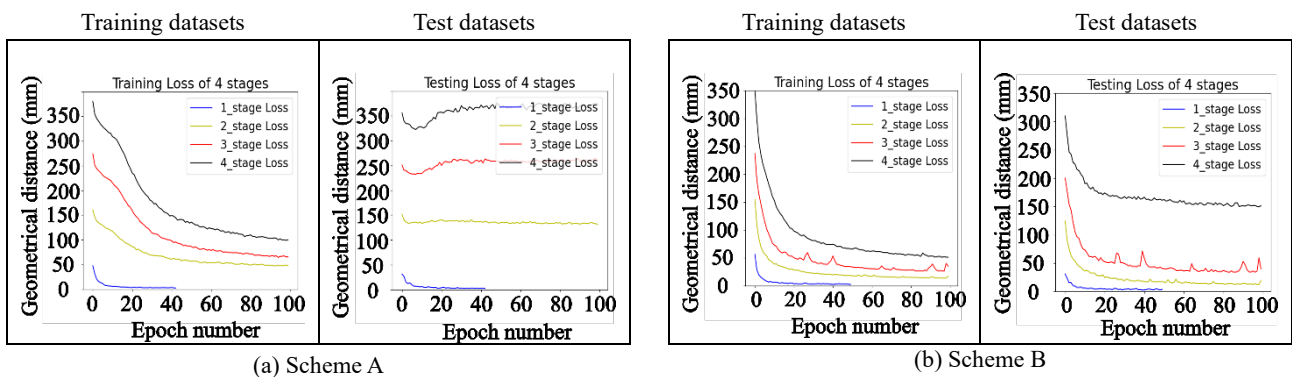


Fig. 12 Trend of mean geometrical distance L_{GD} in training process with paths encoded by Cartesian.

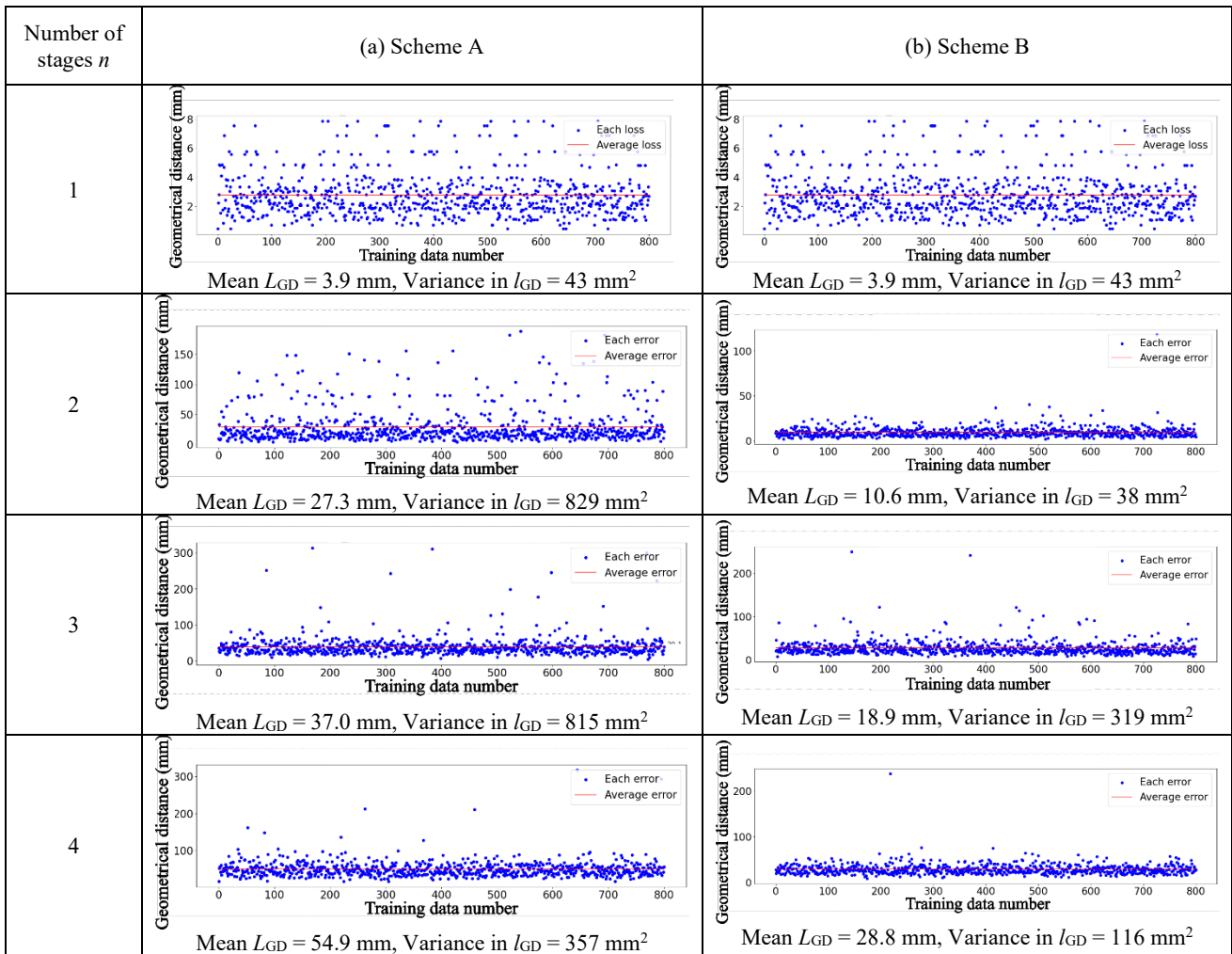


Fig. 13 Individual geometrical distance l_{GD} for each training dataset (in Cartesian encoding).

Table 11 Mean geometrical distance L_{GD} in different training conditions for height distribution. (unit: mm)

Encoding for paths	Input data size	Training datasets				Test datasets			
		Stage $n = 1$	Stage $n = 2$	Stage $n = 3$	Stage $n = 4$	Stage $n = 1$	Stage $n = 2$	Stage $n = 3$	Stage $n = 4$
Binary encoding	Scheme A	0.0	17.4	22.0	25.6	0.0	135.0	273.9	394.5
	Scheme B	0.0	0.0	0.0	0.0	0.0	14.0	138.3	221.8
Cartesian encoding	Scheme A	3.9	27.3	37.0	54.9	2.8	66.6	262.8	374.4
	Scheme B	3.9	10.6	18.9	28.8	2.8	16.0	55.4	142.6

CNN model has an overfitting problem in the test dataset. Thus, a highly effective ML procedure would be required for high prediction accuracy, including regularization and drop-out methods.

Figures 13(a) and 13(b) give the loss of prediction by the CNN trained using Cartesian encoded data. Blue dots represent the individual geometrical distance l_{GD} calculated for each dataset, and red lines represent the averaged distance for all datasets. A small distance means that the error between the correct and predicted path was small. In Fig. 13(a), the prediction results using input data prepared by the scheme A shows that the prediction error was accumulated,

and the averaged distance became larger as the number of laser forming stages n increased. The same trend is found in Fig. 13(b), but scheme B resulted in lower prediction errors. The variance of the individual geometrical distance l_{GD} also exhibited same trend; it increased with the number of laser forming stage n , and scheme B resulted in smaller prediction errors.

Figure 14 shows the predicted results obtained for test datasets. The number of test datasets is 200. The mean geometrical distance L_{GD} was larger than the training's result in both schemes, which presents the overfitting problem, similar to the loss trend shown in Fig. 12. Compared to the result

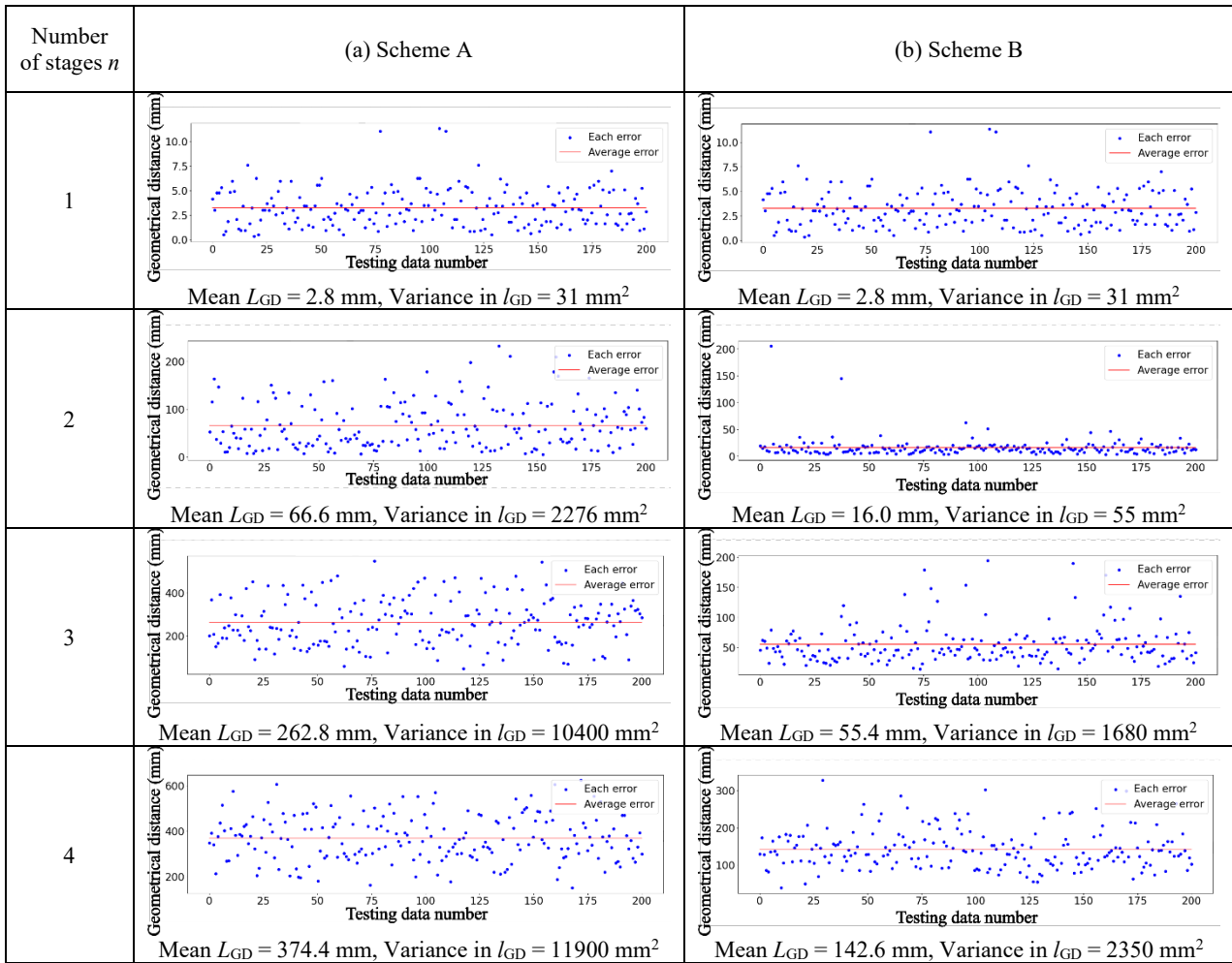
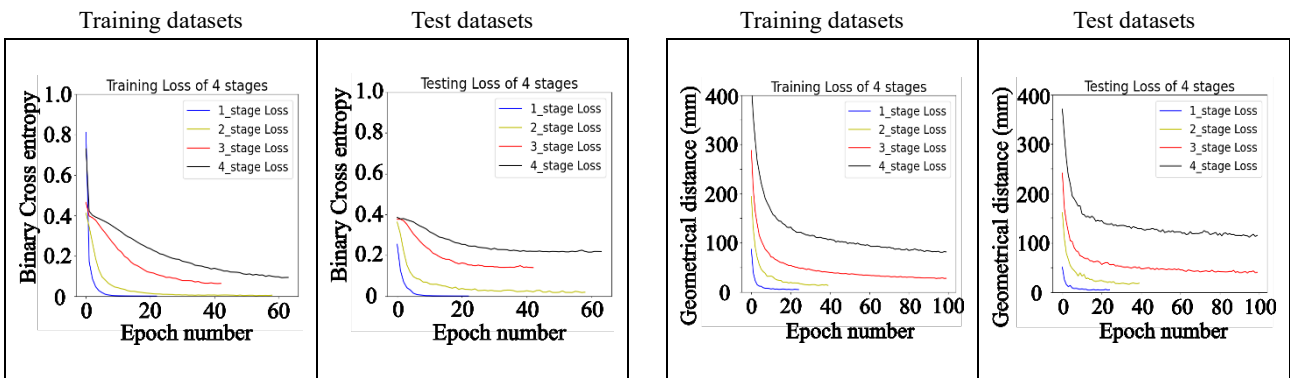


Fig. 14 Individual geometrical distance l_{GD} for each test dataset (in Cartesian encoding).



(a) binary cross entropy L_{BCE} for binary encoding

(b) mean geometrical distance L_{GD} for Cartesian encoding

Fig. 15 Trend of loss functions in training using fringe images (Scheme B).

of Scheme A, Scheme B still demonstrates impressive prediction performance using Cartesian encoding in test datasets. Figures 13 and 14 show that scheme B could suppress the mean geometrical distance L_{GD} lower than scheme A in all multi-forming stages, and it also means scheme B exhibits its sustained proficiency in the Cartesian encoding method.

In order to compare the prediction performance in different training conditions, the result of binary encoding was

shown by the geometrical distance and compared with Cartesian encoding in Table 11. Performance was substantially excellent in single stage forming by any encoding method and by any schemes for input data preparation. Scheme B resulted in better prediction accuracy than scheme A, but the error was still large. Thus, further optimization was expected.

3.2 Prediction results by means of fringe images

Except for the height distribution, the imaginary fringe image was also used as the input of CNN to simulate the

Table 12 Mean geometrical distance L_{GD} in different training conditions for fringe image. (unit: mm)

Encoding for paths	Input data size	Training datasets				Test datasets			
		Stage $n = 1$	Stage $n = 2$	Stage $n = 3$	Stage $n = 4$	Stage $n = 1$	Stage $n = 2$	Stage $n = 3$	Stage $n = 4$
Binary encoding	Scheme B	0.0	0.0	0.4	0.0	0.0	4.4	72.4	166.9
Cartesian encoding	Scheme B	5.3	10.9	15.3	19.0	5.2	16.9	43.4	74.1

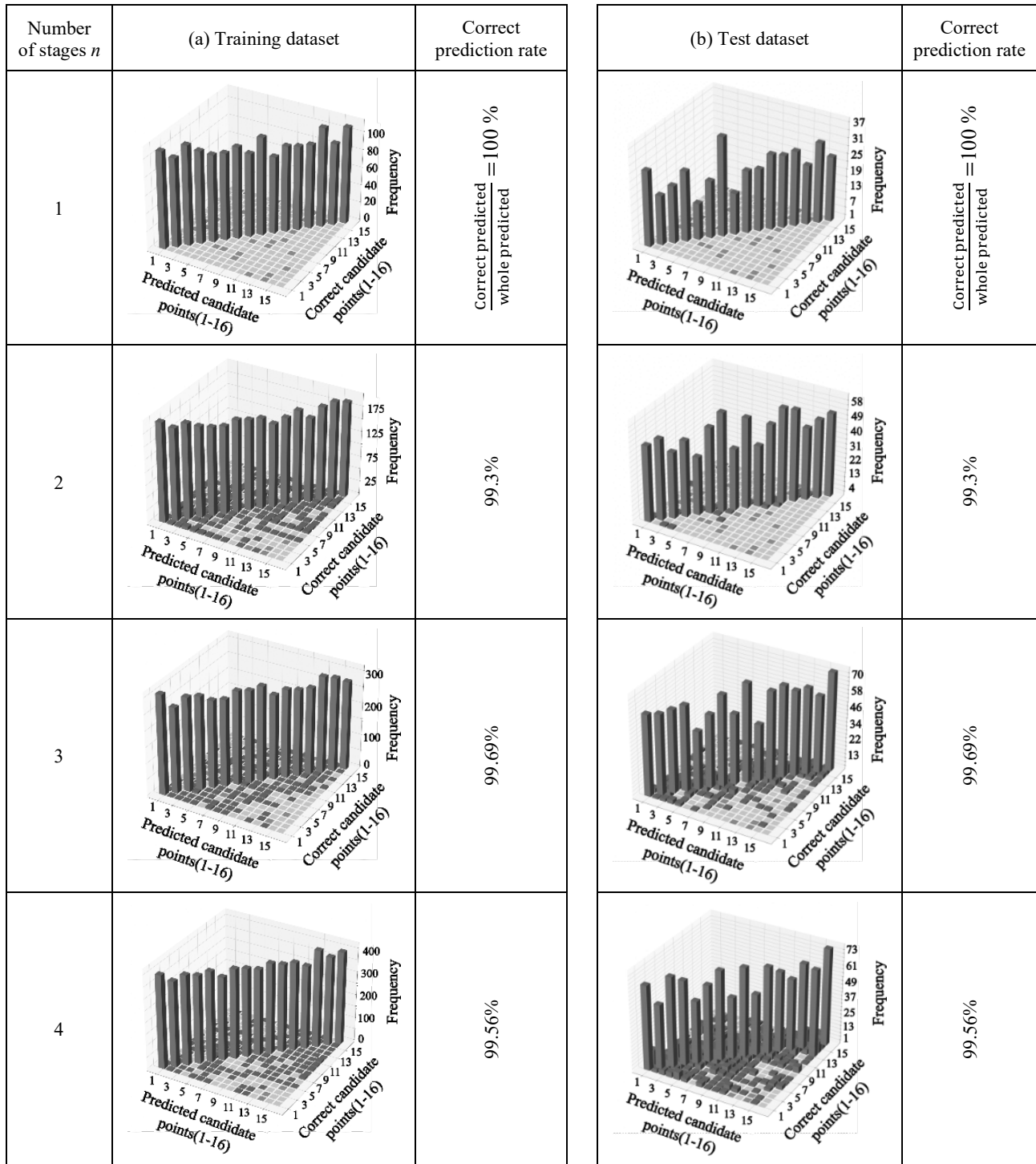


Fig. 16 Frequency distribution of predicted endpoints against ones for Scheme B (in binary encoding, epoch for training ≤ 100).

photo of the experiment. After observing the result of the 3.1 section, it is finding applying Scheme B to CNN has a better prediction performance than Scheme A. Therefore, in this section, only Scheme B and two encoding methods were used in training for the scanning path prediction by means of the fringe images.

Figures 15(a) and 15(b) show the transition of the loss function for results using the input data prepared by scheme B, where scanning paths in datasets were encoded by different methods. It is confirmed that both methods completed the training in a short time. And, comparing these results with Figs 7, 8 and 12, it should be noted that the overfitting was suppressed by the fringe image slightly smaller than the height distribution. The detailed loss trends of each stage are shown in Figs 22 and 23 at the final page.

Figure 16 contains the bar charts displaying distributions of the endpoints of correct laser scanning paths against those

predicted paths by the CNN trained using the binary encoded paths. Comparing Fig. 16 with Fig. 11, it is found that the fringe image archived higher prediction accuracy than the height distribution as input data to train CNN. The correct prediction rate was almost 100% for the training datasets in Fig. 16(a), but the rate decreased for the test datasets as the number of forming stage increased in Fig. 16(b).

The training and testing results of using the Cartesian encoding method in training CNN are shown in Figs.17(a) and (b). Comparison with Figs. 13(b) and 14(b) described that the mean geometric distance L_{GD} and the variance in individual geometric distance l_{GD} were improved by using fringe images. Thus, scheme B exhibited sustained proficiency in applying the imaginary fringe images to train CNN.

The mean geometrical distance L_{GD} using fringe images are also summarized in Table 12. With increase in the number of stages, the training performance using the binary

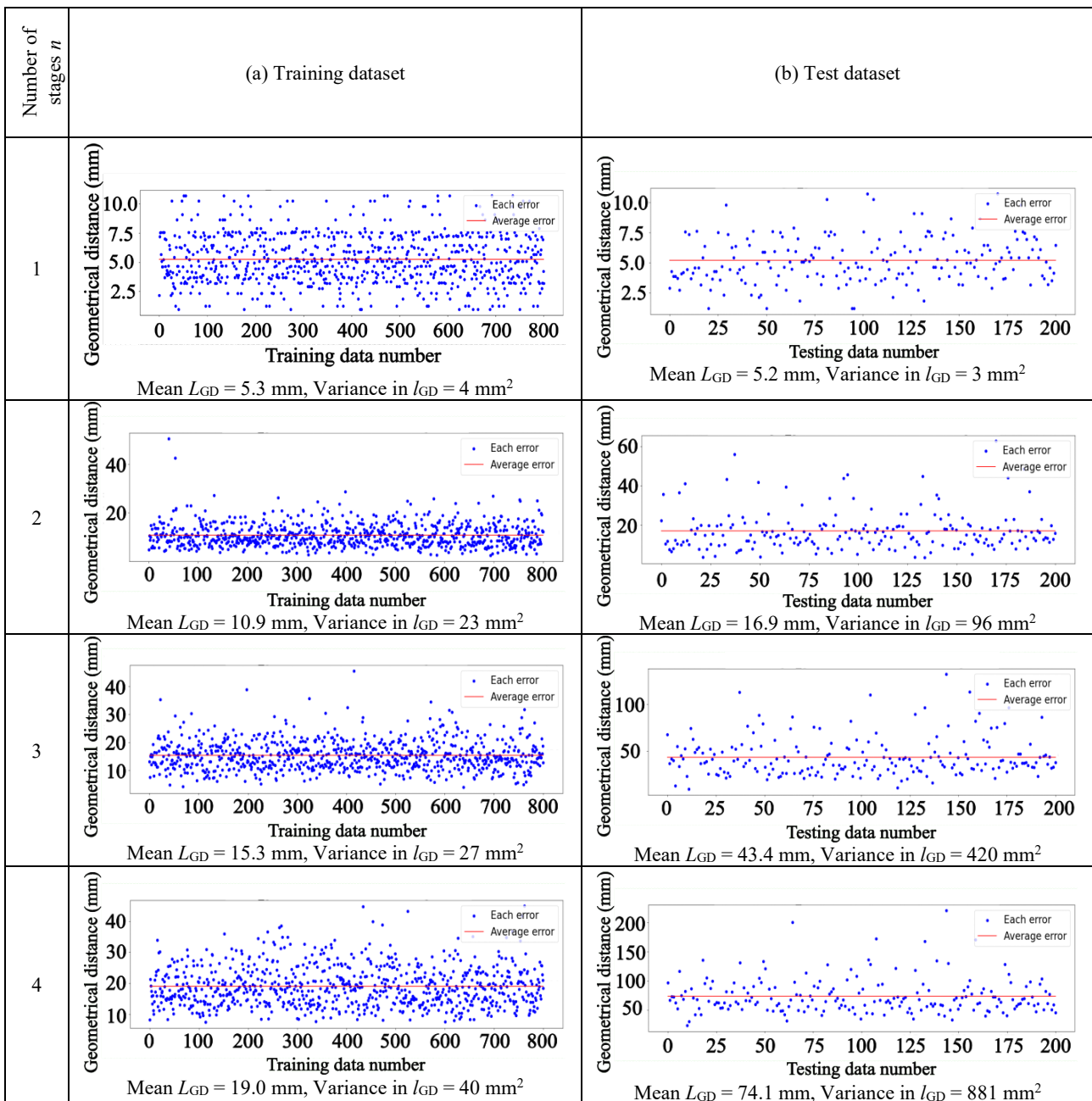


Fig. 17 Individual geometrical distance l_{GD} for each dataset with fringe images (in Cartesian encoding, Scheme B).

encoding exceeded that by Cartesian encoding. However, the prediction performance of Cartesian encoding was more excellent than the other ones in test datasets. Compared to Table 11, both errors of the two encoding methods were lower than the result using height distribution as input data. The result proved the efficacy of applying the imaginary fringe images to train CNN and gave the confidence to use projected photos in training CNN.

4. Conclusions

In this study, laser scanning paths were predicted from shapes of metal sheets using the CNN, which was trained by imaginary datasets containing height distribution or imaginary fringe images. The influences of encoding methods of scanning paths and the input data size were investigated, and then the feasibility of using projected images for path prediction was examined. Obtained results are concluded as follows:

- 1) The results demonstrated the feasibility of applying CNN to predict the scanning paths from height distribution and fringe images in multi-stage laser forming.
- 2) Scheme B, which contains each forming stage's target shape, helps CNN accurately predict the laser scanning path in each stage.
- 3) The binary encoding method has more limitations than Cartesian encoding in presenting the scanning path endpoints. Therefore, CNN used binary encoding predicted better than the used Cartesian ones.
- 4) CNN took less training time and epochs number than a fully connected neural network with two hidden layers in previous study, because fully connected neural networks can be considered a rudimentary network structure in deep learning.
- 5) It is worth noting that using imaginary fringe images as teaching data enables more precise prediction than using height distribution data in no matter encoding methods. From this result, the actual pattern projected picture will be used to build the CNN prediction system, which can identify the real picture, in future research.

References

- [1] Y. Namba: Proc. Inte. Conf. on Lasers '85, (1986) 403.
- [2] M. Geiger: CIRP Annals, 43, (1994) 563.
- [3] T. Ueda, E. Sentoku, Y. Wakimura, and A. Hosokawa: Opt. Lasers Eng., 47, (2009) 1097.
- [4] G. Dearden, S. P. Edwardson, E. Abed, K. Bartkowiak, and K. G. Watkins: Proc. ICALEO 2006, (2006) 505.
- [5] Wenchuan Li and Y. Lawrence Yao: ASME J. Manuf. Sci. Eng., 122, (2000) 445.
- [6] F. Vollertsen, Lasers Eng. 2, (1994) 261.
- [7] T. Ueda, K. Yamada, S. Oikawa, and A. Hosokawa: J. Jpn. Soc. Precis. Eng., 67, (2001) 300 (in Japanese).
- [8] Wenchuan Li and Y. Lawrence Yao: ASME J. Manuf. Sci. Eng., 123, (2001) 674.
- [9] N. Kushida, K. Yamada, R. Tanaka, and K. Sekiya: Proc. JSPE Semestrial Meeting, A (2020) 231.
- [10] P.J Cheng and S.C Lin: Inte. J. Mach. Tools Manuf., 40, (2000) 1185.
- [11] D.H. Kim, D.J. Kim, and B.M. Kim: Inte. J. Adv. Manuf. Technol., 15, (1999) 886.
- [12] Xuefengn Wu, Yahui Liu, Xianliang Zhou, and Aoiei Mou: Sensors, 19, (2019) 3817.
- [13] X. Guo, L. Chen, and C. Shen: Measurement, 93, (2016) 490.
- [14] A.P. Rifai, H. Aoyama, N.H. Tho, S.Z. Md Dawal, and N.A. Masruroh: Measurement, 161, (2020) Article 107860.
- [15] A. Hosokawa, K. Mashimo, K. Yamada, and T. Ueda: JSME Inte. J. Series C, 47, (2004) 52.
- [16] J. Xu, K. Yamada, K. Seikiya, R. Tanaka, and Y. Yamane: Precis. Eng., 38, (2014) 791.
- [17] S. Wada, N. Kushida, P.H. Chou, K. Yamada, E. Sentoku, R. Tanaka, and K. Sekiya: Proc. the 24th Inte. Symp. on Adv. Abras. Technol., (2022) 323.
- [18] S. Gorthi and P. Rastogi: Opt. Lasers Eng., 48, (2010) 133.
- [19] Y. Lu, R. Li, and R. Lu: Comput. Electron. Agric., 127, (2016) 652.
- [20] V. Srinivasan, H. C. Liu, and M. Halioua: Appl. Opt., 23, (1984) 3105.

(Received: June 15, 2023, Accepted: May 2, 2024)