

Simulation Study on Axis Offset Prediction of Convolutional Neural Network Based on Structured Light

Jingwen Zhang¹, Junkai Yu¹, Kun Xia², Zhe Gou¹, Jianping Tan^{1*}

¹The College of Mechanical and Electrical Engineering, Central South University, Changsha, China

²Hunan Aviation Powerplant Research Institute, AECC, Zhuzhou, China

Email: 2506566876@qq.com, 19156281871@163.com, capi7026@qq.com, gouzhe@csu.edu.cn, *jptan@csu.edu.cn

How to cite this paper: Zhang, J.W., Yu, J.K., Xia, K., Gou, Z. and Tan, J.P. (2025) Simulation Study on Axis Offset Prediction of Convolutional Neural Network Based on Structured Light. *Open Journal of Applied Sciences*, 15, 3553-3565.

<https://doi.org/10.4236/ojapps.2025.1511230>

Received: September 18, 2025

Accepted: November 14, 2025

Published: November 17, 2025

Copyright © 2025 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Helicopters assume a pivotal role in domains such as disaster relief, transportation, and military operations. The transmission system stands as one of the three essential components of a helicopter, with the tail drive shaft system tasked with transmitting power to the tail rotor. Although the supercritical design of helicopter tail drive shaft systems presents numerous merits both at home and abroad, the amplitude experiences a sharp increase when traversing the first-order critical speed. In the absence of vibration measurement, monitoring, and control, this can readily lead to overload, triggering collisions between the shaft system and other components, consequently resulting in accidents. With the escalating demand for measurement accuracy in the industrial realm, detecting and predicting the offset of high-speed rotating shafts represents an extremely challenging yet crucial objective in mechanical engineering. This research employs structured light technology to conduct simulation experiments on the projected structured light patterns of both straight and bent shafts. Structured light projects known patterns or light rays (e.g., laser beams or checkerboard patterns) onto the surfaces of objects, and surface deformation is computed based on the altered light patterns. Through 3Ds Max software model rendering and post-processing using OpenCV, we observed the variations of projected patterns across different shaft offsets, trained a Convolutional Neural Network (CNN) model, and applied the CNN to predict the offset values of unknown shafts.

Keywords

Checked Plate Structure Light, Shaft Offset, Convolutional Neural Network, OpenCV Post-Processing

1. Introduction

In the realm of mechanics, shaft misalignment leads to unstable equipment operation, which has an impact on service life and accuracy. Traditional detection approaches employ mechanical devices such as laser rangefinders or calipers. However, typical contact-based measurement methods—such as dial indicator techniques—are unsuitable for high-speed spindle inspection, while single-point methods often suffer from limited accuracy and restricted applicability [1]. In recent years, structured light technology has made progress in 3D scanning and object modeling, facilitating the high-precision acquisition of surface geometric information. When combined with pattern recognition, particularly Convolutional Neural Networks (CNN), it can extract features from 3D data for misalignment prediction.

Structured light, an active optical measurement technique, projects light patterns onto the surface of an object and utilizes a camera to capture information for 3D reconstruction [2]. It possesses advantages in terms of precision, real-time performance, and anti-interference compared to passive stereo vision and is widely utilized in industrial inspection [3]. For shaft misalignment, it employs laser lines or fringe projection to measure deformation and position changes. For instance, shaft misalignment causes deformation of the light band, and through algorithmic or model-based analysis, the misalignment can be calculated.

CNN, a deep-learning model for image processing, extracts features from images by means of local receptive fields, weight sharing, and pooling [4]. In shaft misalignment detection, it can automatically extract misalignment features and conduct regression by learning light stripe deformation, thereby overcoming the limitations of traditional methods [5]. In comparison with other methods, such as classical geometric analytical methods (prone to errors in complex scenarios), fully connected neural networks (resulting in the loss of spatial structure and overfitting), and Transformer (associated with high computational and memory costs [6]), CNN is more appropriate.

This research proposes a shaft misalignment prediction scheme based on virtual prototyping, structured light technology, and CNN. Its objective is to predict minor misalignments with a precision of less than 8 μm , thereby enhancing equipment stability and accuracy in practical applications.

2. Measurement Principle Introduction

2.1. Structural Light and Shaft Displacement

In accordance with the disparate forms of projection light, structured-light measurement primarily encompasses the following approaches:

- 1) Single-line laser scanning
- 2) Structural stripe projection
- 3) Multi-laser light band measurement
- 4) Coding structure light
- 5) Checkered plate structure light:

Checkerboard structured light projects checkerboard patterns (usually alternating black and white squares) onto an object and captures the deformation of the pattern via a camera for three-dimensional object surface reconstruction. The inner corner points of the checkerboard are second-order geometric features marked by substantial grayscale/gradient variations, which can be detected by corner detectors and refined to sub-pixel accuracy. This improves the precision of the mapping from labels to physical measurements, which holds great significance for regression tasks. This method is widely utilized in three-dimensional scanning and object surface detection.

Through rendering in 3DMax software, rendered images of structured light with different offsets (horizontal shifts) combined with the shaft body were obtained (**Figure 1**).

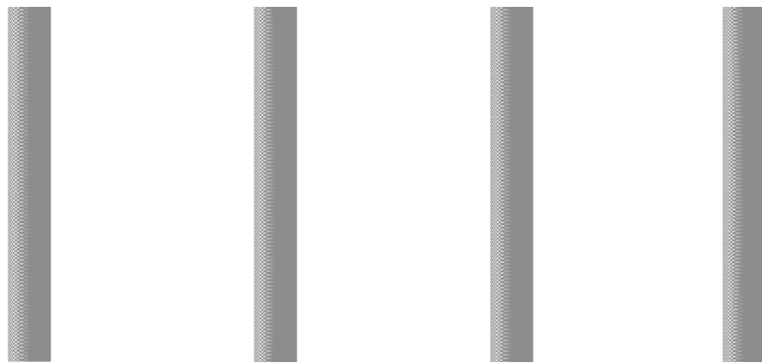


Figure 1. Axis offset rendering diagrams at 0 mm, 0.5 mm, 2 mm, and 5 mm.

Post-process the generated rendered images using OpenCV in Python, specifically performing grayscale conversion to generate comparison images (**Figures 2-4**).

The results are as follows:

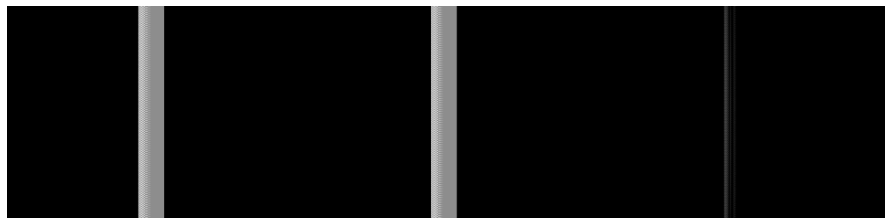


Figure 2. Comparison diagram of 0 mm vs. 0.5 mm axis offset.

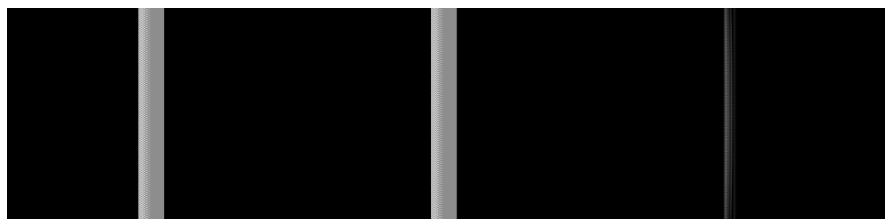


Figure 3. Comparison diagram of 0 mm and 2 mm shaft offset.

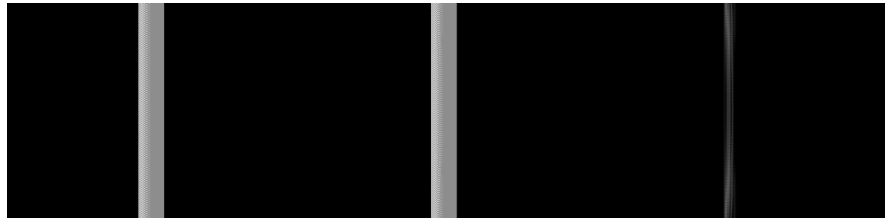


Figure 4. Comparison diagram of 0 mm and 5 mm shaft offset.

As can be clearly seen from the aforementioned illustrations and comparison diagrams, when a slight unidirectional deviation occurs in the shaft body, the checkerboard-structured light demonstrates a notable amplification effect. Through direct visual inspection, it can be observed that the deformation magnitude of the checkerboard varies according to different deviations.

2.2. Key Components of CNN

Convolutional Neural Networks (CNNs) are predominantly constituted by convolutional layers, pooling layers, fully connected layers, and activation functions. Every component assumes a pivotal role in the process of image feature extraction (Figure 5).

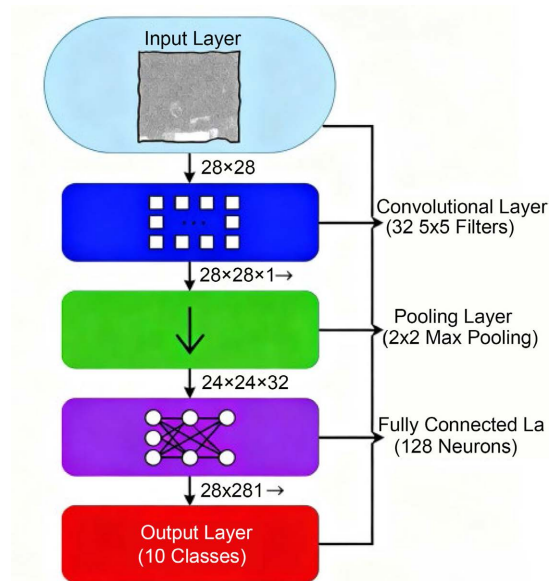


Figure 5. Schematic diagram of convolutional neural network.

1) Convolutional Layer

As the core constituent of Convolutional Neural Networks (CNN), convolutional layers extract local features, including edges, corners, and curves, via learnable kernels that perform convolution operations on input images (Figure 6). In the context of axisymmetric offset detection, these layers capture the curvature patterns of structured light stripes and acquire the offset patterns. Utilizing 3×3 or 5×5 kernels facilitates the detection of localized offset features at diverse scales.

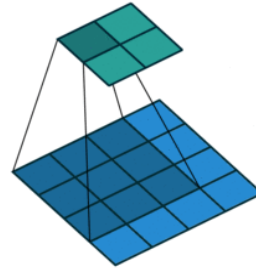


Figure 6. Schematic diagram of local offset features scanned by convolutional kernel.

2) Activation Function

Convolutional Neural Networks (CNNs) utilize nonlinear activation functions, for instance, the Rectified Linear Unit (ReLU), to augment the nonlinear capacity of the model and improve its learning ability for complex offset patterns.

$$\text{ReLu: } f(x) = \max(0, x)$$

In shaft offset detection, the variation pattern of the light band generally demonstrates nonlinear characteristics. The utilization of the Rectified Linear Unit (ReLU) can prevent gradient vanishing and improve the stability of the training process.

3) Pooling Layer

The pooling layer serves the purposes of reducing data dimensionality, enhancing computational efficiency, and augmenting the model's robustness with respect to variations in spectral band shifts.

4) Fully Connected Layer

At the termination of the Convolutional Neural Network (CNN), the fully-connected layer converts the extracted features into the ultimate offset prediction outcome. Generally, a regression model is utilized to yield the axial displacement.

Application of CNN in Structured Light Axis Offset Detection

The optical axis offset detection based on structured light mainly relies on CNN for feature extraction and regression prediction. The typical process is as follows:

1) Data pre-processing:

- ◆ The structured light stripe image is converted to a grayscale image to reduce the computational complexity.
- ◆ Normalized pixel values (such as $[0, 1]$ $[0, 1]$ $[0, 1]$ or $[-1, 1]$ $[-1, 1]$ $[-1, 1]$) improve training stability.
- ◆ In the actual experiments, the following deterministic pre-processing and data-augmentation pipeline was applied before feeding the images to the CNN. Each rendered RGB image was first converted to grayscale, resized to 256×256 pixels, and linearly normalized to the range $[-1, 1]$ $[-1, 1]$ $[-1, 1]$. On this normalized image, we then performed small "micro-augmentations" designed to mimic camera jitter and illumination fluctuations:
- ◆ Pixel translation: the image is padded by 2 pixels on each side (edge padding) and then cropped back after applying a random integer shift $(\Delta x, \Delta y)$ $(\Delta x, \Delta y)$ $(\Delta x, \Delta y)$, where $\Delta x, \Delta y \in \{-2, -1, 0, 1, 2\}$ $\Delta x, \Delta y \in \{-2, -1, 0, 1, 2\}$ $\Delta x, \Delta y$

$\in -2, -1, 0, 1, 2$. This corresponds to horizontal and vertical translations of at most ± 2 pixels.

- ◆ Brightness/contrast scaling: with probability 0.8, the entire image is multiplied by a random scale factor $s \in [0.92, 1.08]$ and in a second augmentation stage a milder scaling with $s \in [0.95, 1.05]$ is always applied at the tensor level.
- ◆ Additive Gaussian noise: with probability 0.5 in the first stage and always in the second stage, pixel-wise Gaussian noise $n \sim N(0, 0.02^2)$ is added in the normalized intensity domain.
- ◆ Clipping: after augmentation, pixel values are clipped to the range $[-2, 2]$ to avoid numerical outliers.
- ◆ These operations are only applied to the training set and preserve the one-to-one correspondence between each structured-light pattern and its ground-truth offset, making the pre-processing pipeline straightforward to replicate in other implementations.
- ◆ Perform data augmentation (such as rotation, translation, noise interference) to improve the model's generalization capability.

2) Feature extraction:

- ◆ A multi-layer CNN is used to extract the offset features of the structured light stripe.
- ◆ Early convolutional layers learn low-level features (edges, light band widths), while deeper layers learn high-level features (light band curvature patterns).

3) Regression prediction:

- ◆ The unidirectional offset value of the shaft body is output through the fully connected layer.
- ◆ The mean square error loss (MSE Loss) or absolute error (MAE) is used to calculate the error between the predicted value and the real offset, and the model parameters are optimized.

4) Optimization and Training:

- ◆ Use the Adam or SGD optimizer to adjust CNN weights to minimize prediction errors.
- ◆ Combining TensorBoard to visualize the loss curve and adjusting hyperparameters to improve accuracy

3. Measurement Scheme and Simulation

3.1. Data Collection and Pre-Processing

This experiment employed the automated rendering technology based on the 3DsMax software script. It projected checkerboard-patterned light with specific width and length onto the shaft body and captured three-dimensional images of the shaft under different offset states. The shaft phantom model was a standard cylinder with a radius of 50 mm and a length of 1000 mm. The captured images were saved in the PNG format with three color channels (RGB 24-bit depth) and processed for uniform dimensions (**Figure 7**).

Using 3DsMax scripts, automated rendering generated projection images of the shaft randomly offset in the negative y-direction within the range [0.1 μm , 50 μm], producing a total of 1500 rendered images at 256×256 resolution.

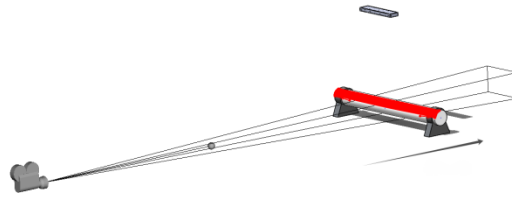


Figure 7. Schematic diagram of light source and camera positions as well as axis offset direction.

In 3DsMax, achieve unidirectional offset of the shaft body by selecting the lattice translation, with the post-offset effect shown in the following figure (**Figure 8**):

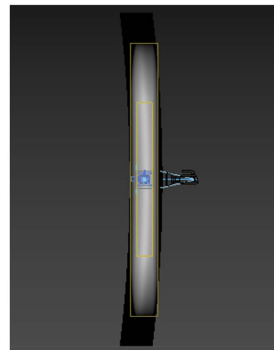


Figure 8. Schematic diagram of one-way offset of simulation axis.

The checkerboard pattern adopts a size of $1 \text{ mm} \times 1 \text{ mm}$, generates a bitmap in Python, and then projects it onto the axis body in 3Dsmax (**Table 1**) (**Figure 8** and **Figure 9**).

Table 1. Checkerboard python parameters.

Chessboard structure light width (pixels)	Height of the checkerboard structure (pixels)	Checkerboard single grid size (pixels)
2500	250	10

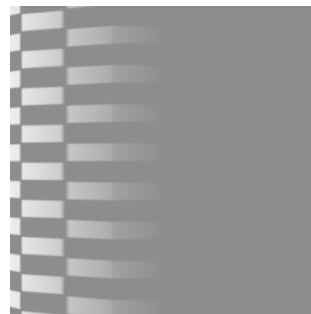


Figure 9. Rendering images used in the experiment.

3.2. Training and Validation

This experimental design adopts a classic convolutional neural network architecture, which consists of three convolutional layers, three max-pooling layers, and two fully-connected layers. After each convolutional layer, a max-pooling layer is utilized for downsampling processing. The design of the convolution kernel captures low-level features (e.g., edges and textures) in images and facilitates the extraction of high-level features through subsequent learning processes.

The model structure is presented as follows, with each structure accompanied by a description of the design principle:

1) Conv1: Input image size is 256×256 , channels are 1, convolution kernel is 5×5 , stride is 1, and zero padding is 2. The 5×5 kernel captures richer local features for encoded stripes. Stride 1 and zero padding 2 keep input-output dimensions unchanged ($256 \times 256 \rightarrow 256 \times 256$) to prevent edge feature loss. 16 channels are a moderate start, laying a foundation for feature stacking and avoiding overfitting.

2) Conv2: The input channels are set to 16, while the output channels are configured as 32. The convolution kernel has a size of 5×5 , the stride is 1, and zero-padding is applied with a value of 2. The doubling of the channels serves to integrate the low-level features obtained from Conv1 into mid-level features. Maintaining the kernel size and stride guarantees the continuity of the feature map dimensions (the feature map size changes from 256×256 to 128×128 through MaxPool1).

3) Conv3: The input channels are set to 32, while the output channels are configured as 64. The convolution kernel size is 5×5 , the stride is 1, and the zero-padding is 2. The 64 channels are capable of extracting high-level features for regression tasks. Subsequent to MaxPool3, the feature map is downsampled to 32×32 , which effectively preserves the core features and maintains parameter manageability.

4) Fully connected layer: A hidden layer consisting of 1024 units is employed, and the output layer comprises 1 unit for the regression task. The 1024-unit hidden layer compresses the flattened $64 \times 32 \times 32$ features derived from Conv3 and integrates high-level features. The 1-unit output layer is utilized for the regression task, where the output corresponds to the quantitative prediction of axial deviation for measurement accuracy.

Overall, the chosen CNN architecture reflects a trade-off between representation capacity and overfitting risk given the relatively small dataset size (1500 images). With 256×256 grayscale inputs, three convolutional blocks with 5×5 kernels and channel widths of 16-32-64 are sufficient to capture both the local deformation of individual checkerboard corners and the global curvature of the structured-light stripes. The larger 5×5 kernels allow each layer to observe a slightly wider neighbourhood, so deeper stacks of 3×3 kernels are not required, which helps to keep the architecture simple and stable. After the final max-pooling layer, the feature map has size $64 \times 32 \times 32$; flattening this tensor and feeding it into a 1024-unit fully connected layer yields a model with about 6.7×10^7 trainable pa-

rameters, dominated by the last fully connected layer. This size is large enough to model the nonlinear mapping from stripe deformation to axis offset, but still compact enough to be trained reliably on a single GPU without severe overfitting.

Compared with deeper residual networks or Transformer-based encoders, which typically contain many more parameters and require substantially larger datasets, the present three-block CNN is better aligned with the scale and complexity of our problem: the images contain highly structured, low-texture patterns, and the target is a single scalar regression value. On the other hand, significantly shallower architectures (e.g. with only one or two convolutional layers) would not have a large enough receptive field to capture the smooth curvature changes of the structured-light stripes across the entire shaft surface. For these reasons, we adopt a “medium-depth” CNN with three convolutional blocks and one hidden fully connected layer as a principled compromise between accuracy, robustness and computational cost.

Data Set: A training set containing 1050 images was used, with the validation set and test set each containing 225 images.

For reproducibility, the 1500 offsets were first drawn from μ (0.1 μm - 50 μm) and then split into training, validation and test subsets using the 70%/15%/15% ratio described above. Because all three subsets are created from the same underlying sampling procedure, their empirical offset histograms are almost identical and cover the full 0.0001 - 0.05 mm range.

Hardware Configuration: The experiments were conducted on an NVIDIA RTX 4060 GPU using the PyTorch framework.

Evaluation Metrics: The prediction error was assessed using absolute error as the evaluation criterion, with a required prediction accuracy of 0.008 mm.

During the training process, the sum of the Mean Squared Error (MSE) and the Mean Absolute Error (MAE) was employed as the loss function, which was optimized using the Adam optimizer. The data set was partitioned into a training set, a validation set, and a test set. The training set was utilized for model learning, the validation set for hyperparameter adjustment, and the test set for assessing the model’s generalization ability. Specifically, the samples were divided such that 70% constituted the training set, 15% the validation set, and 15% the test set. All these operations were implemented on the Python platform.

3.3. Prediction and Evaluation

After the model training is completed, the model is evaluated with the test set, and the absolute error between the predicted value and the true value is calculated.

4. Experimental Results

4.1. Interpretation of Result

The training process of the model exhibited stability, without any program-related errors. Leveraging Python’s matplotlib library, plots depicting the training and validation losses were generated. These plots indicated that the model achieved

favorable convergence after 50 epochs. Additionally, a comparison plot of the predicted and actual values was constructed. The horizontal axis of this plot represents the image index of the test set, while the vertical axis represents the offset value (mm) for each shaft image, as illustrated in the figure (Figure 10 and Figure 11).

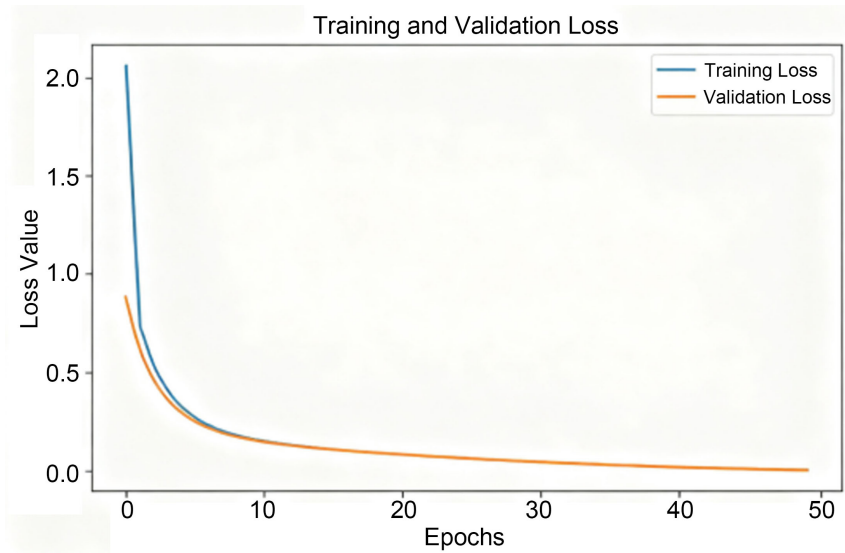


Figure 10. Training and validation loss convergence plot.

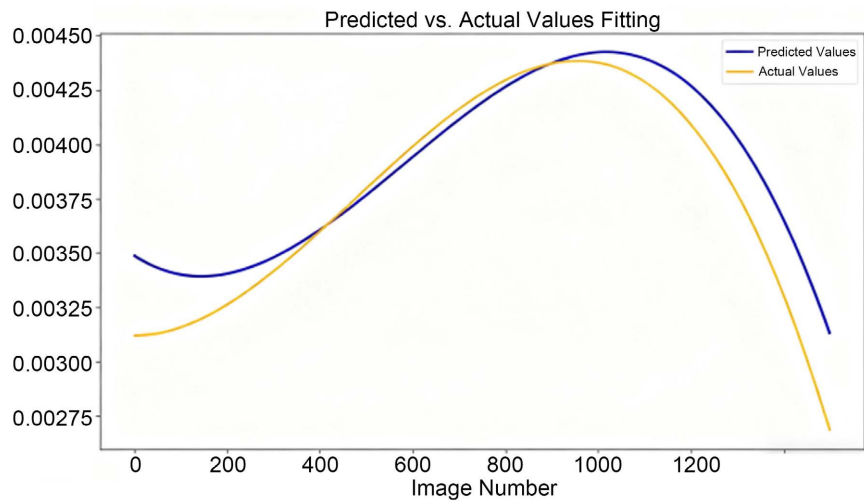


Figure 11. Test set predicted values versus actual values curve fitting plot.

It is observable that the predicted value and the actual value are nearly in a state of fitting, suggesting that the current model training effect is satisfactory.

4.2. Recognition Effectiveness of External Images

The aforementioned results represent the fitting outcomes of the predicted values and the actual values of the test set during the model training process. Subsequently, rendering of an offset image of the structured light axis body outside the

test set is carried out to examine whether our model can accomplish accurate pattern recognition within the permissible error range.

In 3Dmax, the shaft body is manually offset by 8 micrometers in the negative y-axis direction, after which its rendered image is output. After undergoing the same pre-processing as described previously, the trained model is invoked to predict its offset value (**Table 2**).

Table 2. Real offset and predicted offset scale.

Real axial body offset (μm)	The model predicts the axis offset (μm)	error, deviation (μm)
8	8.99	0.99

It is important to note that the 8 μm case used for external testing was generated independently of the original 1500 images and was strictly excluded from the training, validation and test splits. Although 8 μm (0.008 mm) lies inside the continuous offset range [0.0001 mm, 0.05 mm] [0.0001 mm, 0.05 mm] [0.0001 mm, 0.05 mm], there is no training sample whose ground-truth offset is exactly 8 μm because all offsets were sampled from a continuous uniform distribution. The CNN has therefore only seen offsets slightly smaller and larger than 8 μm during training, and its accurate prediction on this external case demonstrates its ability to interpolate within the learned offset range rather than memorize discrete training values.

Based on the experimental data presented in the above table, it is evident that the thoroughly trained deep learning model demonstrates superior performance on the training set and exhibits outstanding recognition ability for unknown test images. It can precisely capture the key features of images and make accurate classification judgments, which indicates robust generalization capabilities and accomplishes the pattern recognition process from data learning to application. The recognition workflow, encompassing feature extraction, pattern matching, and classification decision-making, operates effectively, validating the rationality of the model architecture and the efficacy of the training methodology.

5. Discussion

5.1. Methodological Advantages

The axis offset prediction method proposed in this study, which is based on structured light and the convolutional neural network (CNN) model, exhibits the following advantages:

- 1) High precision: The three-dimensional data acquired through structured light technology offers high-resolution information regarding the object's surface, which contributes to enhancing the prediction precision.
- 2) Automation: The automatic extraction of image features via Convolutional Neural Network (CNN) mitigates the workload associated with traditional manual feature extraction.
- 3) Low-resolution requirement: In comparison with conventional image match-

ing, this method imposes lower demands on camera resolution. The input samples in this study are merely 256×256 , significantly diminishing the cost requirements during camera procurement.

In terms of computational cost, the proposed CNN remains practical for deployment. Running on an NVIDIA RTX 4060 GPU in our PyTorch implementation, the end-to-end inference time (including data transfer and pre-/post-processing) for a single 256×256 grayscale image is about 1 s. This speed is sufficient for our current offline simulation studies and for quasi real-time shaft monitoring scenarios where images are acquired at a relatively low frame rate, and it also leaves room for further optimization via model pruning or lightweight backbones in future work.

5.2. Existing Problems and Solutions

5.2.1. Existing Issues

The model exhibits outstanding performance on the virtual platform, with errors significantly lower than the stipulated 8-micron threshold. Nevertheless, this prediction is exclusively grounded in ideal static conditions featuring zero axial unidirectional displacement. In practical operational scenarios, during rotation at the first-level or higher critical speeds, multidirectional vibrations and displacements emerge concurrently with noise interference. Moreover, the actual reflection patterns of light from shaft materials deviate considerably from ideal conditions. Collectively, these factors lead to a substantial decline in the prediction accuracy of the model.

5.2.2. Solution

- 1) Add a data augmentation step. Simulate camera angle variations via random rotation within $\pm 5^\circ$, slight shooting offsets via random translation, and camera focal length changes via random scaling. Randomly crop to retain partial image content, enhancing the model's local sensitivity and sample diversity to closely approximate real-world working conditions.

- 2) Reasonably and appropriately increasing the sample size, extending the training epochs, applying L2 regularization to the model, and performing gradient clipping to prevent overfitting.

- 3) Within the convolutional neural network, an attempt is made to incorporate the self-attention mechanism to strengthen the latent relationship and recognition of the global line structured light deformation.

6. Conclusion

This research puts forward a methodology for forecasting axis-pair deviations by leveraging virtual platforms, structured light, and convolutional neural networks (CNN), enabling the accurate prediction of minute axis-pair offsets. In the 3DsMAX software, a checkerboard structured light pattern is projected onto the surface of the axis pair to generate 1500 training images. Subsequently, a CNN model is trained with the sum of the mean squared error (MSE) and the mean absolute

error (MAE) serving as the loss function. The trained model exhibits excellent curve-fitting performance between the true values and the predicted values, which validates both the rationality of the proposed method and the accuracy of the results. When predicting axis-pair deviations for unseen images within the 0.0001 - 0.05 mm offset range used during training, the errors are maintained within the allowable tolerance of 8 μm . This approach substantially improves the detection accuracy of axis-pair deviations, indicating considerable industrial application value. Future studies may concentrate on optimizing the model architecture based on structured light, reducing computational costs, and extending it to more intricate application scenarios.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Jiang, G., *et al.* (2024) Study on Detection Method of Spindle Rotation Error Considering System Vibration. *Journal of Vibration, Measurement & Diagnosis*, **44**, 740-747+828-829.
- [2] Pan, B., Guan, Q., Wang, X. and Chen, S.Y. (2010) Strategies of Real-Time 3D Reconstruction by Structured Light. 2010 *Chinese Conference on Pattern Recognition (CCPR)*, Chongqing, 21-23 October 2010, 1-5. <https://doi.org/10.1109/CCPR.2010.5659126>
- [3] Shen, Y., Kong, L., Tang, X., Song, H., Hua, S. and Ding, Z. (2025) Large Depth-of-Field Structured Light System for 3D Microscale Measurement with Integrated Speckle Denoising. *Optics & Laser Technology*, **192**, Article ID: 113688. <https://doi.org/10.1016/j.optlastec.2025.113688>
- [4] Lyu, H. (2023) Research on Corrosion Recognition Method of Steel Based on Convolutional Neural Network. 2023 *IEEE 6th International Conference on Information Systems and Computer Aided Education (ICISCAE)*, Dalian, 23-25 September 2023, 507-511. <https://doi.org/10.1109/ICISCAE59047.2023.10393077>
- [5] Jiang, Q., Xiao, Y., Zhou, G., *et al.* (2025) Integrating Radius Margin Constraints and Class Variance for Improved CNN-Based Image Recognition. *Knowledge-Based Systems*, **327**, Article ID: 114188. <https://doi.org/10.1016/j.knosys.2025.114188>
- [6] Yan, Y., Wang, D., Ye, J., *et al.* (2025) DCHF_T: A Multi-Dimensional Adaptive Compression Approach for Transformer-Based Models. *Neurocomputing*, **656**, Article ID: 131071. <https://doi.org/10.1016/j.neucom.2025.131071>