

Improving the Stability of CNC Machine Tools Based on Artificial Intelligence

JunGuo, Shuangyan Long*, ShiyunMo

School of Information Engineering, Hunan Automotive Engineering Vocational University, Zhuzhou, 412001 China

Received 15 Sep 2025

Accepted 12 Feb 2026

Abstract

The stability of machining accuracy in CNC machine tools is often affected by multiple factors such as fixture conditions, tool wear, ambient temperature, and vibration, which can easily cause the actual machining trajectory to deviate from the theoretical path. To suppress such errors and improve machining stability, it is necessary to dynamically correct the machining process through real time compensation parameters. For this purpose, this paper proposes an artificial intelligence model that integrates deep learning and machine learning, aiming to provide a real time, adaptive, and high precision dynamic parameter compensation scheme for lathes during machining. In terms of data modeling, the aforementioned influencing factors are quantified into feature variables, each associated with a set of adjustable compensation parameters. Each data sample consists of a feature vector, a compensation parameter vector, and the corresponding measured machining accuracy value, thereby constructing a structured dataset for error compensation. In terms of methodology implementation, first, based on this dataset, a deep learning framework is employed to build an error compensation model, achieving real time optimization of machining parameters and enhancement of accuracy. Second, by collecting a large number of image samples of qualified and unqualified workpieces, a machine learning based visual inspection model is trained to automatically discriminate the quality of workpieces after compensation. Simulation experiments show that the proposed error compensation model improves machining accuracy on the training set and test set from 0.66 to 0.98 and from 0.65 to 0.967, respectively; the visual inspection model achieves fully accurate recognition on both types of datasets. In practical machining verification, after intelligent compensation machining of 50 gears, all were judged as qualified by the visual inspection system. The results indicate that the artificial intelligence model constructed in this paper can significantly enhance the machining accuracy and product qualification rate of CNC machine tools.

© 2026 Jordan Journal of Mechanical and Industrial Engineering. All rights reserved

Keywords: CNC; Apriori algorithm; Deep learning; Machine learning; Intelligent error compensation; Visual inspection.

1. Introduction

CNC machine tools[1, 2] are core equipment in the field of high-end precision manufacturing, and their machining accuracy directly affects the quality and overall performance of components in key industries such as aerospace and precision instruments. During the long-term operation of the machine tool, the combined effect of mechanical wear, thermal deformation and deformation caused by cutting force leads to the gradual accumulation of comprehensive errors and the decline of machining accuracy. Therefore, the development of efficient, accurate and adaptive error compensation technology has become a key research direction to ensure the continuous and stable operation of the manufacturing system and the consistency of processing quality.

In the research of error compensation for traditional CNC machine tools, scholars mainly build explicit mathematical models based on the principles of geometry and rigid body kinematics to describe the influence

mechanism of specific error sources. For example, the document[3] established a kinematics model for the geometric error of the turntable of a two axis CNC machine tool, separated and compensated the coupling error between the axes through coordinate transformation, effectively suppressed the impact of error transmission on the workpiece accuracy in multi axis linkage, but its method relies on high-precision measuring devices, the measurement range is limited by hardware, and the process of model parameter identification is complex. The document[4] established the angular positioning error identification and geometric compensation model based on the circular trajectory by using the double ball and rod instrument detection system. This method has good effect in improving the positioning accuracy of the rotary axis. However, the model does not fully incorporate the coupling effect of linear axis translation error and guide rail nonlinearity, so its compensation ability for spatial straightness error is limited. Starting from the mechanical characteristics of the cutting process. The document[5] conducted static compensation for the error caused by the

* Corresponding author e-mail: ShuangyanLong@outlook.com.

cutting force through experimental modeling. However, the dynamic stiffness distribution of each component of the machine tool structure is uneven, and the frequency response characteristics are different, resulting in the fluctuation of the compensation effect with the change of working conditions and insufficient stability.

With the deep integration of intelligent manufacturing and information technology, error modeling methods are gradually developing from traditional mechanism driven to data-driven and hybrid modeling to deal with the complex error scenarios of multi-source, time-varying and coupling of CNC machine tools. For example, The document[6] built a prediction model of thermal error of machine tool spindle based on neural network, and achieved high-precision fitting of the nonlinear mapping relationship between temperature field and thermal deformation, effectively improving the thermal processing accuracy. The document[7] further introduced clustering fuzzy algorithm to intelligently divide the working conditions of machine tools, and combined with artificial neural network to establish a dynamic compensation model for radial thermal error, which enhanced the adaptability of the model under variable working conditions. The document [8] proposed an error compensation framework based on closed-loop manufacturing process from the perspective of system control, and realized the independent optimization of the processing process by adjusting the process parameters through real-time feedback.

At present, the research of artificial intelligence in the field of NC machine tool error compensation has made significant progress. However, the existing AI based compensation methods still face prominent bottlenecks in the promotion and application. First, most models rely on a large number of offline data, and the training and optimization process is time-consuming; Second, its compensation effect is often limited to specific experiments or steady-state conditions, and its dynamic adaptability to the complex and changeable environment in actual production is insufficient. Therefore, this paper aims to break through the static limitations of existing methods, study how to deeply integrate artificial intelligence algorithm and real-time industrial big data, and explore an intelligent error compensation technology that can dynamically perceive and adapt to different processing environments, so as to provide adaptive compensation for CNC machine tools under variable working conditions, and

ultimately improve its processing stability and accuracy reliability.

2. DATA ANALYSIS

2.1. Error compensation data analysis

The machining accuracy of CNC machine tools is often subject to the combined influence of various factors such as fixtures, cutting tools, ambient temperature, and vibrations. To mitigate the actual impact of these external factors on machining outcomes, it is typically necessary to introduce corresponding compensation parameters to ensure the stability of machining precision. This study is based on a dataset of 100,000 entries provided during actual production processes in enterprises. In this dataset, features w_0 to w_{17} correspond to the characteristic values of influencing factors such as fixtures, cutting tools, ambient temperature, and vibrations, while compensation c_0 to c_7 represent the corresponding error compensation parameter characteristic values, as shown in Figure 1. Since different compensation parameters have varying effects on machining results, to enhance the relevance of subsequent analysis, it is essential to preprocess the raw data first, thereby constructing a dataset more suitable for modeling tasks.

2.2. Image recognition data analysis

In image data acquisition, the quality of the captured images is susceptible to environmental interference, and the acquired images often contain various types of noise, which may distort the grayscale distribution. to address these issues, mean filtering[9], median filtering[10], and sharpening filtering[11] can be applied for image pre-processing.

2.2.1. Mean filter

Mean filtering achieves a balance between detail preservation and noise reduction by selecting an appropriate convolution kernel size. Taking a 3×3 neighborhood as an example, the window elements (30, 106, 22, 196, 223, 216, 36, 52, 187) are summed and averaged to obtain the mean value of 119, which then replaces the central element 223. The principle of mean filtering is illustrated in Figure 2.

	w0	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10	w11	w12	w13	w14	w15	w16	w17	c0	c1	c2	c3	c4	c5	c6	c7	accuracy
1	2.73	0.186	0.935	0.194	0.4	0.176	0.52	0.797	0.406	0.096	0.110000111	0.668	0.261	0.468	0.387	0.096	0.672	0.594	1.842	1.474	0.582	1.378	0.993	0.855	1.599	1.577	0.90204
2	0.355	0.03	0.708	0.997	0.13	0.924	0.837	0.953	0.224	0.513	0.110000111	0.364	0.168	0.835	0.266	0.031	0.487	0.91	1.702	0.534	1.367	1.444	1.551	1.349	0.892	1.183	0.91157
3	0.082	0.249	0.883	0.725	23.677	0.110000111	0.922	0.706	0.824	nan	0.337	0.204	0.045	0.186	0.048	0.476	0.931	0.789	1.131	1.328	1.584	0.483	1.482	1.055	0.553	1.204	0.91009
4	0.018	0.583	0.002	0.126	0.437	0.110000111	0.992	nan	0.93	0.886	0.553	0.819	0.813	0.36	0.372	0.9	0.541	0.908	1.202	0.278	0.974	0.706	0.805	1.304	1.748	1.869	0.90204
5	0.829	0.955	0.795	0.215	0.837	0.383	0.142	0.110000111	0.278	0.883	0.802	0.428	0.616	0.07	0.273	0.023	0.956	0.407	1.344	0.19	0.931	0.869	1.836	1.828	0.406	0.432	0.91064
6	1.044	9.11	0.997	0.159	0.553	0.643	0.976	0.482	0.089	0.173	0.661	0.793	0.292	0.127	0.304	0.435	0.842	1.982	1.637	1.745	1.721	1.151	0.909	1.781	0.346	0.88384	
7	1.044	9.11	0.997	0.159	0.553	0.643	0.976	0.482	0.392	0.089	0.173	0.661	0.793	0.292	0.127	0.304	0.435	0.842	0.843	0.02	0.286	0.761	0.327	1.329	1.726	1.975	0.85296
8	1.044	9.11	0.997	0.159	0.553	0.643	0.976	0.482	0.392	0.089	0.173	0.661	0.793	0.292	0.127	0.304	0.435	0.842	0.205	0.205	1.538	0.07	0.683	0.45	1.751	1.677	0.85901
9	1.044	9.11	0.997	0.159	0.553	0.643	0.976	0.482	0.392	0.089	0.173	0.661	0.793	0.292	0.127	0.304	0.435	0.842	1.06	1.271	1.436	0.486	1.675	0.137	1.733	1.979	0.8789
10	0.044	9.11	0.997	0.159	0.553	0.643	0.976	0.482	0.392	0.089	0.173	0.661	0.793	0.292	0.127	0.304	0.435	0.842	0.863	0.918	1.021	0.024	1.066	0.553	1.655	1.771	0.86039
11	0.044	0.11	0.997	0.159	0.553	0.643	0.976	0.482	0.392	0.089	0.173	0.661	0.793	0.292	0.127	0.304	0.435	0.842	1.846	1.51	0.612	1.668	1.705	0.861	0.575	1.743	0.86755

Figure 1. Partial datasets provided by enterprises

2.2.2. Median filter

Median filtering can smooth the image and eliminate isolated noise points. Taking the 3x3 neighborhood as an example, the window elements 22, 30, 36, 52, 106, 187, 196, 216, 223 are sorted from smallest to largest to get the median 106. The specific principle is shown in Figure 3.

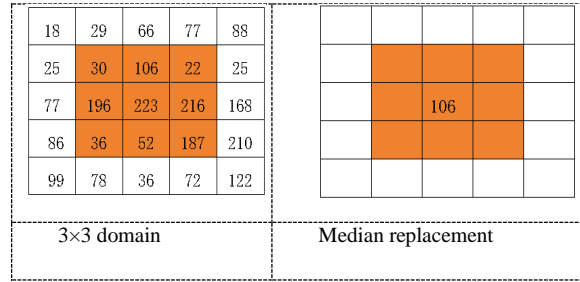


Figure 3. Principle of median filtering

2.2.3. Sharpening filter

The Laplacian in the sharpening filter is used to calculate the image edges, the Laplacian can seek the edges in both directions at the same time, Figure 4 calculates the edge values in the horizontal and vertical directions with $i(x,y)$ as the center.

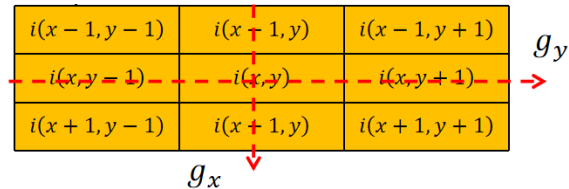


Figure 4 Horizontal and vertical edges

Taking $i(x,y)$ as the center point, and $i(x,y-1)$, $i(x,y+1)$ are the edges in the horizontal direction of $i(x,y)$, and $i(x-1,y)$, $i(x+1,y)$ then are the edges in the vertical direction of $i(x,y)$. Find the difference between the horizontal and vertical directions and $i(x,y)$, respectively, as shown in Equation (1). Figure 5 Calculation of edge values in the diagonal direction centered on $i(x,y)$.

Taking $i(x,y)$ as the center point, $i(x-1,y-1)$, $i(x-1,y+1)$, $i(x+1,y-1)$, $i(x+1,y+1)$ are the edges in the diagonal direction of $i(x,y)$, and the difference between the diagonal direction and $i(x,y)$, respectively, as shown in Equation (2).

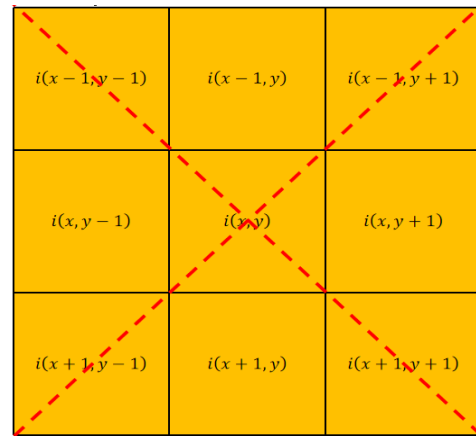


Figure 5. Edge in diagonal direction

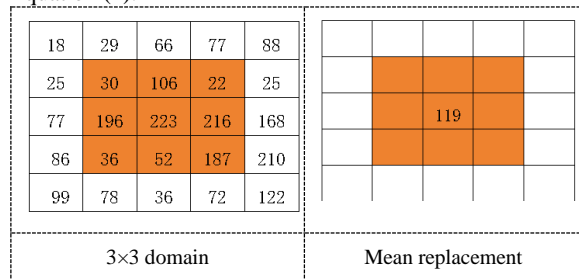


Figure 2. Principle of mean filtering

$$\begin{aligned}
 f(x,y) &= i(x+1,y) - i(x,y) + i(x-1,y) - i(x,y) + i(x+1,y) - i(x,y) + i(x,y+1) - i(x,y) \\
 &= i(x+1,y) + i(x-1,y) + i(x+1,y) + i(x,y+1) - 4i(x,y) = \sum_m \sum_n \begin{pmatrix} 0 & i(x-1,y) & 0 \\ i(x,y-1) & -4i(x,y) & i(x,y+1) \\ 0 & i(x+1,y) & 0 \end{pmatrix} \quad (1)
 \end{aligned}$$

$$= \sum_m \sum_n \begin{pmatrix} i(x-1,y-1) & i(x-1,y) & i(x-1,y+1) \\ i(x,y-1) & -4i(x,y) & i(x,y+1) \\ i(x+1,y-1) & i(x+1,y) & i(x+1,y+1) \end{pmatrix} * \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$\begin{aligned}
 g(x,y) &= i(x+1,y-1) - i(x,y) + i(x-1,y-1) - i(x,y) + i(x+1,y+1) - i(x,y) + i(x-1,y+1) - i(x,y) \\
 &= i(x+1,y-1) + i(x-1,y-1) + i(x+1,y+1) + i(x-1,y+1) - 4i(x,y) \\
 &= \sum_m \sum_n \begin{pmatrix} i(x-1,y-1) & 0 & i(x-1,y+1) \\ 0 & -4i(x,y) & 0 \\ i(x+1,y-1) & 0 & i(x+1,y+1) \end{pmatrix} \\
 &= \sum_m \sum_n \begin{pmatrix} i(x-1,y-1) & i(x-1,y) & i(x-1,y+1) \\ i(x,y-1) & -4i(x,y) & i(x,y+1) \\ i(x+1,y-1) & i(x+1,y) & i(x+1,y+1) \end{pmatrix} * \begin{pmatrix} 1 & 0 & 1 \\ 0 & -4 & 0 \\ 1 & 0 & 1 \end{pmatrix} \quad (2)
 \end{aligned}$$

3. INDUSTRIAL BIG DATA ALGORITHM ANALYSIS

3.1. Analysis of deep learning algorithms

Deep learning algorithms [12,13] typically fit complex nonlinear mapping relationships by constructing neural networks with a multi-layer structure. This study adopts a BP neural network to establish an error compensation model, where the input layer X_1, X_2, \dots, X_n corresponds to the eigenvalues of 18 influencing factors extracted from the dataset, such as fixtures, cutting tools, ambient temperature, and vibration. The output layer O_1, O_2, \dots, O_m corresponds to the eight compensation parameters to be predicted, as specifically illustrated in Figure 6. During the model training process, to achieve optimal performance, it is necessary to systematically tune and optimize hyperparameters, including the number of network layers, the number of neurons in the hidden layers, batch size, loss function, optimizer type and learning rate, as well as the number of training iterations.

3.2. Analysis of machine learning algorithms

Machine learning algorithms [14,15] possess strong capabilities in learning from data, enabling them to uncover patterns from existing complex data to predict future behavioral outcomes and trends. There are various types of machine learning algorithms. In this paper, the random forest algorithm is employed to construct a gear quality visual inspection model for automatically determining the eligibility of workpieces after error compensation. The application of this algorithm primarily consists of two stages: In the first stage, multiple training subsets are generated from the preprocessed gear image dataset—which undergoes mean filtering, median filtering, and sharpening filtering—using the Bootstrap sampling method. A decision tree is then constructed in parallel for each subset. In the second stage, during model prediction, the classification results of all decision trees on the input image (features extracted from tooth profile contours, surface textures, etc.) are aggregated through voting. The class with the highest number of votes (eligible or ineligible) is taken as the final recognition result. By integrating the judgments of multiple decision trees, this method effectively enhances the model's generalization capability and recognition robustness for

complex image features. The principle of the random forest algorithm is illustrated in Figure 7.

4. DATA PREPROCESSING AND ALGORITHM IMPLEMENTATION

4.1. Error compensation data preprocessing realization

In the dataset of 100,000 entries provided by the enterprise, issues such as inconsistent data types, duplicate rows, null values, and non-uniform feature scales exist. To improve data quality and ensure the reliability of subsequent modeling, preprocessing is required for the error compensation sample data using tools such as NumPy and Pandas. This primarily includes the following four steps: first, standardizing data types to numeric values to avoid calculation errors caused by type inconsistencies; second, removing duplicate rows to ensure sample uniqueness; third, eliminating records containing null values to reduce noise and redundancy; fourth, performing data normalization to mitigate the impact of differing feature scales. After processing the dataset described in Section 2.1 using the above methods, non-numeric information (e.g., "0b110000110" and "666int57.4197") in the original data has been converted to numeric types, duplicate rows and rows containing null values have been removed, and all compensation feature values have been normalized to the range (-1, 1). A sample of the preprocessed data is shown in Figure 8.

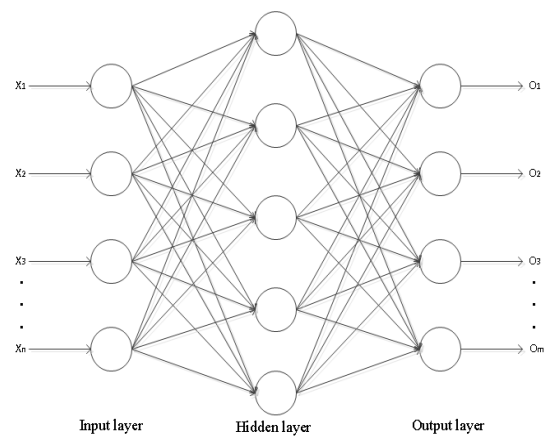


Figure 6. Neural network structure

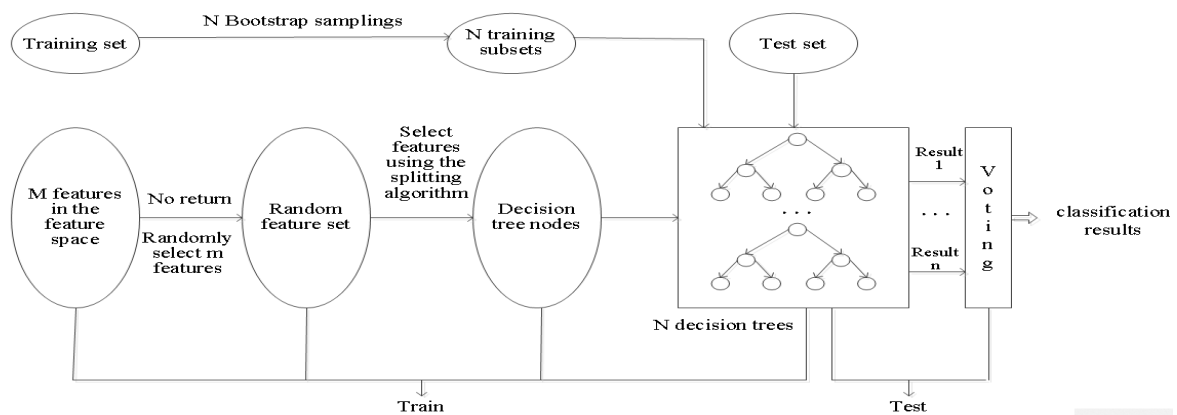


Figure 7. Schematic diagram of random forest algorithm

4.2. Picture training data collection

A sufficient number of images were collected to form the training set. These image samples were categorized into qualified and unqualified groups, as illustrated in Figures 9

and 10. Since images often contain various types of noise during the acquisition process, and the quantity and quality of the collected sample data directly influence subsequent model training, it is necessary to preprocess the image samples.

1	w0	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10	w11	w12	w13	w14	w15	w16	w17	c0	c1	c2	c3	c4	c5	c6	c7	accuracy
2	0.73	0.186	0.935	0.194	0.4	0.176	0.52	0.797	0.406	0.096	0.776	0.668	0.261	0.488	0.387	0.096	0.672	0.584	1.842	1.474	0.562	1.378	0.993	0.855	1.599	1.577	0.90204
3	0.355	0.03	0.708	0.997	0.13	0.924	0.837	0.953	0.224	0.513	0.654	0.364	0.188	0.835	0.266	0.031	0.487	0.91	1.702	0.534	1.367	1.444	1.551	1.349	0.692	1.183	0.91157
4	0.062	0.249	0.883	0.725	0.677	0.878	0.922	0.706	0.624	0.789	0.337	0.204	0.045	0.186	0.048	0.476	0.931	0.789	1.131	1.328	1.594	0.483	1.482	1.055	0.553	1.204	0.91009
5	0.010	0.583	0.002	0.126	0.437	0.678	0.992	0.659	0.93	0.886	0.553	0.819	0.913	0.36	0.372	0.9	0.541	0.908	1.202	0.278	0.974	0.706	0.805	1.304	1.748	1.869	0.90094
6	0.928	0.955	0.795	0.215	0.837	0.383	0.142	0.635	0.278	0.893	0.802	0.428	0.616	0.07	0.273	0.023	0.956	0.407	1.344	0.19	0.931	0.869	1.836	1.828	0.406	0.432	0.91064
7	0.044	0.11	0.997	0.159	0.553	0.643	0.976	0.482	0.393	0.069	0.173	0.661	0.793	0.292	0.127	0.304	0.435	0.842	1.992	1.637	1.745	1.721	1.151	0.909	1.781	0.346	0.86384
8	0.044	0.11	0.997	0.159	0.553	0.643	0.976	0.482	0.392	0.069	0.173	0.661	0.793	0.292	0.127	0.304	0.435	0.842	0.843	0.02	0.286	0.761	0.327	1.329	1.726	1.975	0.85296
9	0.044	0.11	0.997	0.159	0.553	0.643	0.976	0.482	0.392	0.069	0.173	0.661	0.793	0.292	0.127	0.304	0.435	0.842	0.205	0.205	1.538	0.07	0.683	0.45	1.751	1.677	0.85901
10	0.044	0.11	0.997	0.159	0.553	0.643	0.976	0.482	0.392	0.069	0.173	0.661	0.793	0.292	0.127	0.304	0.435	0.842	1.06	1.271	1.436	0.488	1.675	0.137	1.733	1.979	0.8799
11	0.044	0.11	0.997	0.159	0.553	0.643	0.976	0.482	0.392	0.069	0.173	0.661	0.793	0.292	0.127	0.304	0.435	0.842	0.863	0.918	1.021	0.024	1.066	0.553	1.855	1.771	0.86039
12	0.044	0.11	0.997	0.159	0.553	0.643	0.976	0.482	0.392	0.069	0.173	0.661	0.793	0.292	0.127	0.304	0.435	0.842	1.846	1.51	0.812	1.658	1.705	0.861	0.575	1.743	0.86755

Figure 8. Processed partial dataset

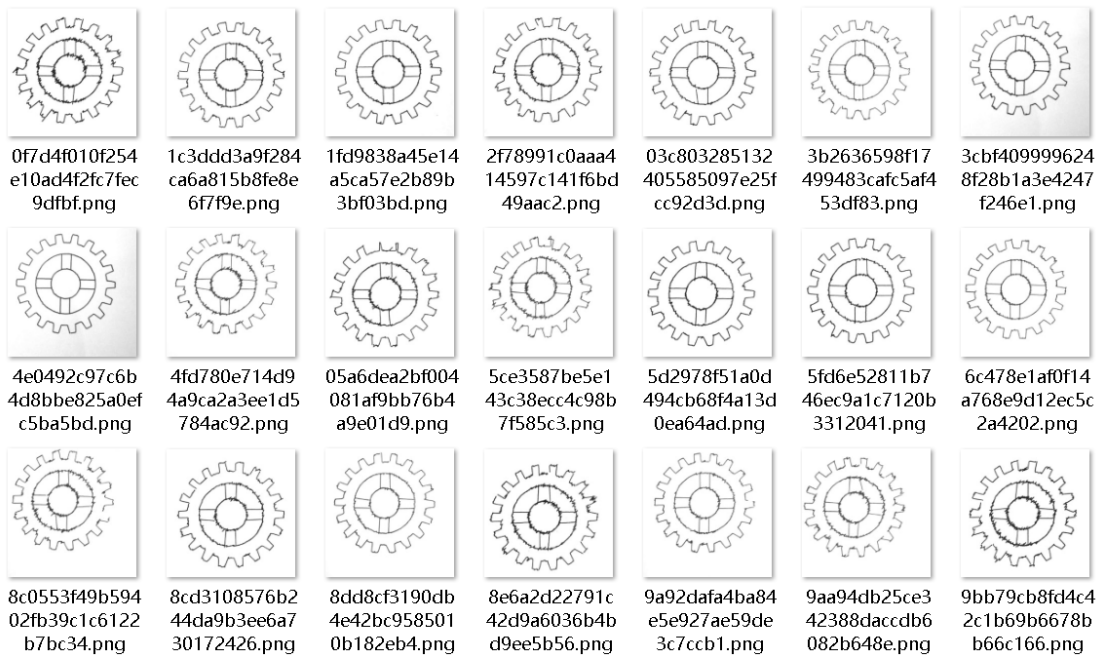


Figure 9. Collection of qualified gear samples

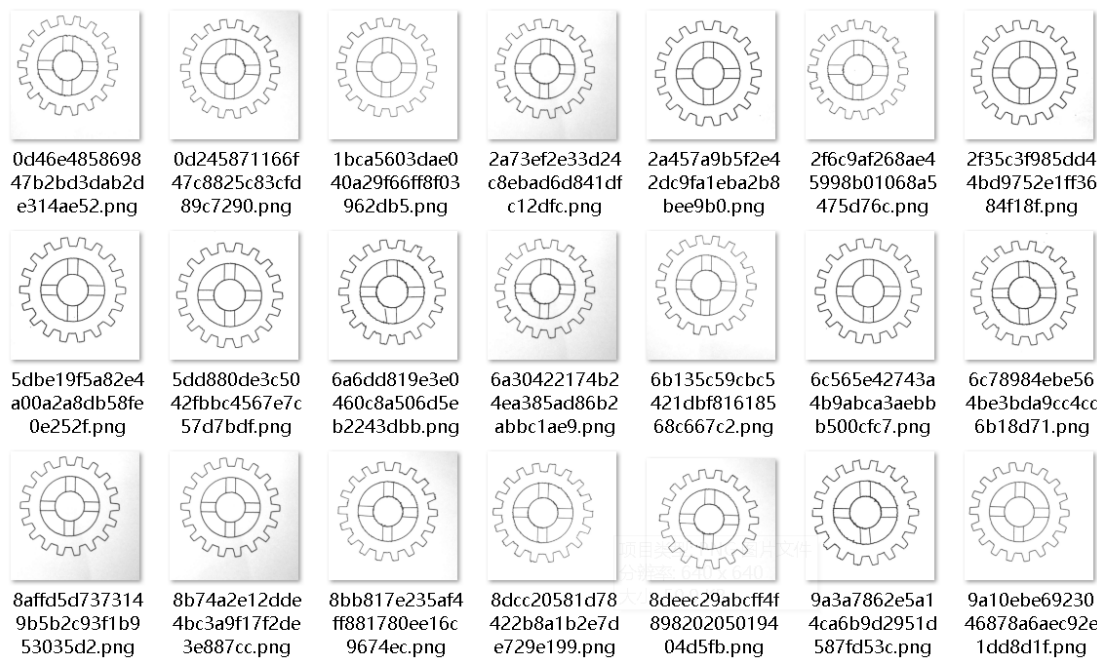


Figure 10. Collection of unqualified gear samples

4.2.1. Realization of image preprocessing based on mean value filtering

To smooth the pictures, mean value filtering was applied to the collected images based on the principle described in subsection 2.2.1. The specific implementation code is provided below:

```
defblur_filter(img_path, save_path):
    image = cv2.imread(img_path)
    median = cv2.blur(image, (3, 3))
    cv2.imwrite(save_path, median)
```

In this process, `img_path` represents the original image, `save_path` denotes the processed image, `blur()` refers to the mean filter function, and `(3, 3)` specifies the size of the convolution kernel. A comparison of the images before and after mean filter processing is presented in Figure 11.

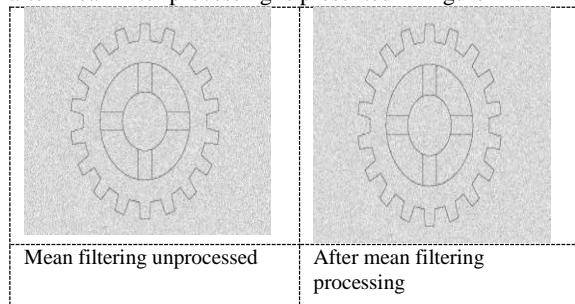


Figure 11. Comparison chart before and after mean filtering processing

4.2.2. Realization of image preprocessing based on median filtering

The median filter, described in Subsection 2.2.2, can be applied to images corrupted by impulse noise. The following code implements this filtering process:

```
defmedian_filter(img_path, save_path):
    image = cv2.imread(img_path)
    median = cv2.medianBlur(image, 3)
    cv2.imwrite(save_path, median)
```

Here, `img_path` denotes the original image path, `save_path` represents the output path for the processed image, `medianBlur()` is the median filter function, and the parameter 3 specifies the kernel size (neighborhood). A comparison of the images before and after median filter processing is shown in Figure 12.

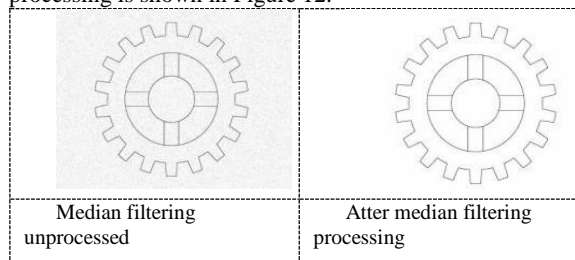


Figure 12. Comparison chart before and after median filtering processing

4.2.3. Sharpening filtering based on the realization of image preprocessing

In this paper, the sharpening filter is mainly based on the implementation of the Laplace operator, Laplace operator processing picture can be divided into two cases, one is based on the horizontal and vertical direction; the other is based on the diagonal direction, using the principle of sharpening filtering in subsection 2.2.3, the Laplace horizontal and vertical direction of the main implementation of the core code is as follows:

```
defsharpen_filter(img_path, save_path):
    image=cv2.imread(img_path,
    cv2.IMREAD_GRAYSCALE)
    # Horizontal, vertical
    kernel1 = np.asarray([[0, 1, 0],
    [1, -4, 1],
    [0, 1, 0]])
    # Diagonal direction
    kernel2 = np.asarray([[1, 1, 1],
    [1, -8, 1],
    [1, 1, 1]])
```

The comparison before and after the sharpening filtering process is shown in Figure 13.

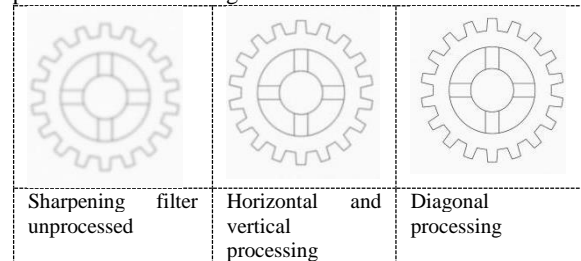


Figure 13. Comparison chart before and after sharpening filtering processing

4.3. Algorithm realization

4.3.1. Based on deep learning error compensation algorithm realization

The deep learning algorithm in subsection 3.1 is used to establish the error compensation algorithm model, and the steps of the error compensation algorithm model establishment are as follows:

- Step 1: Model construction


```
model = tf.keras.Sequential([ norm_layer,
      layers.Dense(100,train_features.shape[1],
      activation="relu"),
      layers.Dense(300, activation="relu"),
      layers.Dense(300, activation="relu"),
      layers.Dense(300, activation="relu"),
      layers.Dense(32,activation="relu",
      layers.Dense(train_labels.shape[1])) )
      model.compile(loss="mse",optimizer="adam")
```
- Step 2: Model training


```
model.fit(
      train_features,
      train_labels,
      epochs=200,
      batch_size=2048,
      verbose=2,)
```
- Step 3: Model prediction


```
test_preds = model.predict(test_features)
      print("MSE:%.4f" % mean_squared_error(test_labels,
      test_preds))
```
- Step 4: Model Deployment. The trained error compensation algorithm model is deployed to the cloud platform for machining error accuracy compensation to reduce machine tool machining errors.

4.3.2. Based on machine learning visual detection algorithm realization

The machine learning algorithm in subsection 3.2 is utilized to build the visual detection algorithm model, and the visual detection algorithm model building steps are as follows:

- Step 1: Model building

```

model=RandomForestClassifier(n_estimators=100,random_state=42,criterion='gini',)
model.fit(X_train, y_train)

```

- Step 2: Model Prediction
`y_pred = model.predict(X_test)`
`print("Accuracy:", accuracy_score(y_test, y_pred))`
- Step 3: Model Deployment Deploy the trained visual inspection model to the cloud platform for identifying qualified and unqualified workpieces.

5. RESULTS TEST

5.1. Error compensation simulation results

Based on the deep learning model constructed in Section 4.3.1, two approaches were implemented: one utilized the dataset provided by enterprises in Section 2.1 without any processing, resulting in Figure 14; the other employed the dataset preprocessed with error compensation in Section 4.1, yielding Figure 15. As shown in the figures, the accuracy of the training and test sets after error compensation data preprocessing can reach 0.98 and 0.967, respectively, representing improvements of 0.32 and 0.317 compared to the unprocessed data.

5.2. Visual detection simulation results

Based on the machine learning models constructed in Section 4.3.2, two approaches were employed. The first used the unprocessed image dataset collected in Section 4.2, which produced Figure 16. The second used the image dataset after three filtering steps, yielding Figure 17. As shown in the figures, the training and test set accuracy after image preprocessing reached 0.98 and 0.92, respectively. These values represent improvements of 0.33 and 0.34 over the results from the unprocessed data.

5.3. Actual processing

The 50 gears were processed using a deep learning-based error compensation algorithm. Each gear was then inspected by a machine learning-based visual detection algorithm. Qualified gears were marked with a check, and unqualified ones with a cross. The final results showed that all 50 gears were qualified, resulting in a 100% qualification rate. Figure 18 shows a photograph of the processed gears.

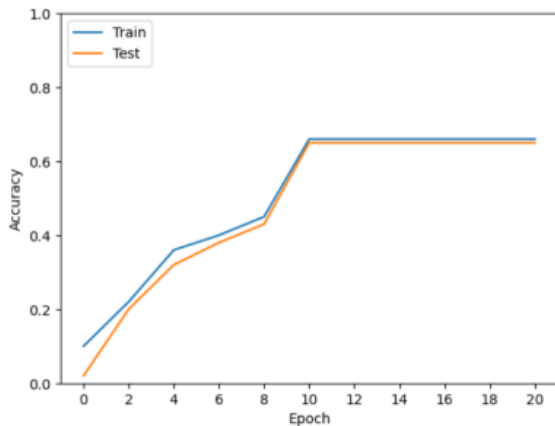


Figure 14. Error data not processed

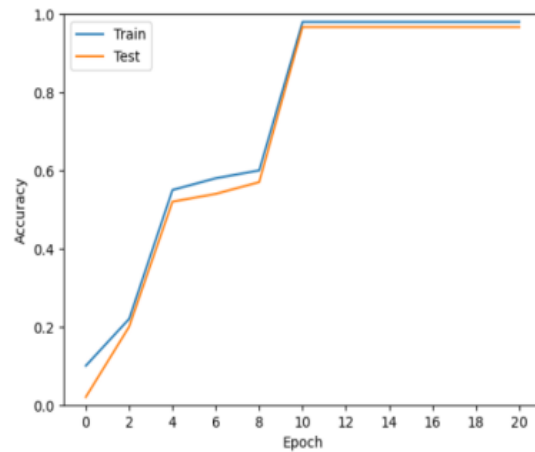


Figure 15. Error data preprocessing model accuracy

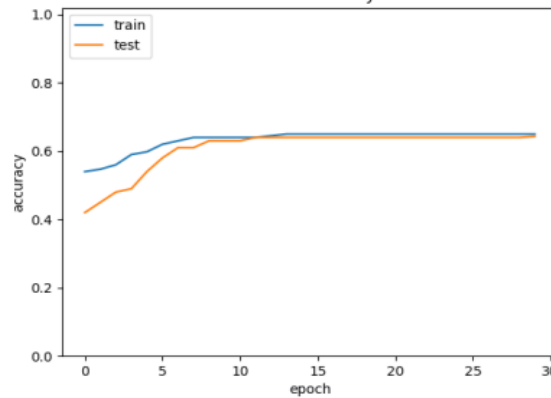


Figure 16. Image sample not processed

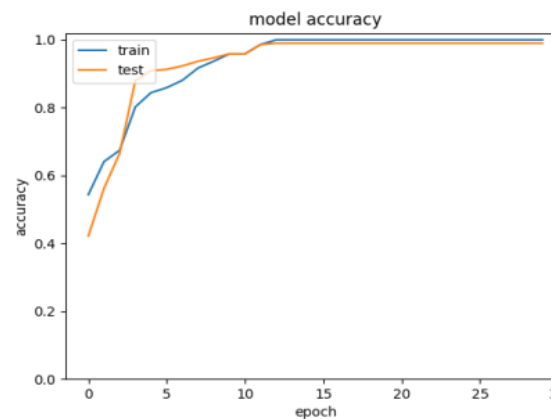


Figure 17. Image Sample Preprocessing

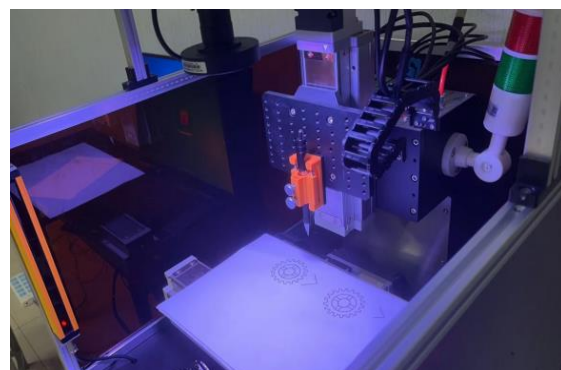


Figure 18. Gear processing

6. CONCLUSIONS

To improve the machining stability of CNC machine tools under complex environmental and inherent factors, this study approaches the issue from two perspectives: error compensation and quality inspection. First, an intelligent compensation model is developed within a deep learning framework, using preprocessed error data. Through adaptive learning and parameter optimization, the model dynamically predicts and compensates for machining errors, thereby enhancing workpiece accuracy. Second, a visual inspection model is established based on a machine learning framework, combined with image preprocessing techniques such as mean filtering, median filtering, and sharpening filtering, to automatically assess workpiece qualification after compensation. Simulation results indicate that for error compensation, the model achieves machining accuracies of 0.98 and 0.967 on the training and test sets, respectively. In visual inspection, recognition accuracies reach 0.98 and 0.92 on the training and test sets, respectively. Experimental validation further demonstrates that after machining, error compensation, and visual inspection of 50 gear samples, all workpieces are judged as qualified, confirming the effectiveness and reliability of the integrated approach. Future work will explore more diverse neural network architectures and larger datasets to further analyze the method's stability and generalization capability under different working conditions.

Funding

This research is supported by the Hunan Provincial Natural Science Foundation Project (2023JJ60219).

Data Availability

The data supporting the findings of this study are available within the article.

Author Contributions

J.G. Conceptualization, S.L. Methodology, S.M. Investigation; Resources.

References

- [1] P. Shi, X. Tong, Y. Xu, M. Cai, "New classification method of nonstandard parts for CNC selection based on machine learning and geometric analysis", *Journal of Intelligent Manufacturing*, 2025, pp.1-24. <https://doi.org/10.1007/s10845-024-02547-4>.
- [2] X. Zhang, W. Zhu, S. Chen, X. Zhang, Y. Zhang, "Improvement of volumetric accuracy in CNC machine tools based on moving least squares", *The International Journal of Advanced Manufacturing Technology*, 2025, pp.1-13. <https://doi.org/10.1007/s00170-025-15768-5>.
- [3] S. Osei, W. Wang, Q. Ding, "A new method to identify the position-independent geometric errors in the rotary axes of five-axis machine tools", *Journal of Manufacturing Processes*, Vol. 87, 2023, pp.46-53. <https://doi.org/10.1016/j.jmapro.2023.01.001>.
- [4] Q. Li, W. Wang, J. Zhang, H. Li, "All position-dependent geometric error identification for rotary axes of five-axis machine tool using double ball bar", *The International Journal of Advanced Manufacturing Technology*, Vol. 110, No. 5, 2020, pp.1351-1366. <https://doi.org/10.1007/s00170-020-05962-y>.
- [5] B. Qi, C. Chen, Z. Liu, Y. Zhu, L. Lan, J. Chen, J. Guo, "Research on the diffusion kernel density estimation method for modeling the cutting force spectrum and program load spectrum considering multiple manufacturing conditions", *Journal of Manufacturing Processes*, Vol. 127, 2024, pp.140-159. <https://doi.org/10.1016/j.jmapro.2024.07.097>.
- [6] Z. Guo, S. Liu, M. Yan, J. Yang, Q. Liu, D. Li, "Thermal error prediction of electric spindle based on improved whale algorithm optimized neural network", *The International Journal of Advanced Manufacturing Technology*, 2025, pp.1-15. <https://doi.org/10.1007/s00170-025-15614-8>.
- [7] G. H. Chen, B. Zhou, T. Li, J. Mao, B. Li, Z. X. Fu, "Research on thermal error compensation modeling for the machine tool integrated drive system based on energy consumption big data and an optimized bidirectional network", *Precision Engineering*, Vol. 94, 2025, pp.91-112. <https://doi.org/10.1016/j.precisioneng.2025.02.024>.
- [8] Y. S. Zhou, Z. W. Tang, X. L. Shi, J. Y. Tang, Z. M. Q. Li, "Efficient and accurate worm grinding of spur face gears according to an advanced geometrical analysis and a closed-loop manufacturing process", *Journal of Central South University*, Vol. 29, No. 1, 2022, pp.1-13. <https://doi.org/10.1007/s11771-021-4830-7>.
- [9] Y. Wang, G. Li, Z. Hu, C. Sun, Z. Li, "A Novel D-InSAR and offset tracking fusion algorithm based on variance-masked mean filtering", *Electronics Letters*, Vol. 61, No. 1, 2025, pp.e70173. <https://doi.org/10.1049/ell2.70173>.
- [10] Y. Niu, X. Chen, H. Yin, "Median filtering forensics using spatial and frequency domain residuals", *The Journal of Supercomputing*, Vol. 81, No. 4, 2025, pp.1-20. <https://doi.org/10.1007/s11227-025-07099-9>.
- [11] H. Cui, H. Pan, K. Zhang, "SCU - Net++: A Nested U-Net based on sharpening filter and channel attention mechanism", *Wireless Communications and Mobile Computing*, 2022, pp.2848365. <https://doi.org/10.1155/2022/2848365>.
- [12] B. Zhou, G. H. Chen, J. Mao, Y. Li, S. W. Zhang, "A ball screw all-round error compensation technology based on novel hybrid deep learning for CNC machine tool", *Journal of Vibroengineering*, Vol. 26, No. 5, 2024, pp.1216-1236. <https://doi.org/10.21595/jve.2024.23841>.
- [13] B. Jiang, R. Sun, Z. L. Li, L. Xu, H. Liao, X. Y. Teng, B. Li, "Local corner smoothing based on deep learning for CNC machine tools", *Scientific Reports*, Vol. 15, No. 1, 2025, pp.404. <https://doi.org/10.1038/s41598-024-84577-9>.
- [14] P. Shi, X. Tong, Y. Xu, M. Cai, "New classification method of nonstandard parts for CNC selection based on machine learning and geometric analysis", *Journal of Intelligent Manufacturing*, 2025, pp.1-24. <https://doi.org/10.1007/s10845-024-02547-4>.
- [15] Y. Wang, Y. Cao, X. Qu, M. Wang, Y. Wang, C. Zhang, "A review of the application of machine learning techniques in thermal error compensation for CNC machine tools", *Measurement*, Vol. 243, 2025, pp.116341. <https://doi.org/10.1016/j.measurement.2024.116341>.