VNUHCM Journal of Engineering and Technology 2026, 9(2): 2849-2858

Open Access Full Text Article

Research article

Using Deep Learning Model in Prediction of Surface Roughness

Bao Vo Quang, Tai To Tan, Than Trong Khanh Dat *

Use your smartphone to scan this QR code and download this article

ABSTRACT

In contemporary industrial operations, machining precision has undergone significant advancements, placing surface roughness measurement at the forefront of quality assessment. However, current methods for measuring surface roughness are not only time-consuming but also labor-intensive, often requiring additional effort for equipment preparation and fixture setup. Moreover, the current practice often involves direct measurement on the sample, necessitating the removal of the sample from the machining equipment, which can introduce setup errors and potentially compromise the original machining standards. Recognizing these challenges, our project aimed to streamline the post-machining inspection process by evaluating surface roughness through imaging of the machined surface. This article explores the application of deep learning models, facilitated by MATLAB software, to diagnose surface roughness in machining. Additionally, we propose a comprehensive method for sampling measurement, including procedures and conditions, to guide further project development based on collected samples or by incorporating any missing conditions to enhance the project in the future. Leveraging non-contact measurement methods ensures precise surface details regarding roughness and glossiness, making them suitable for processing. This advancement represents a significant step forward in testing and measurement, with wide-ranging applications in mechanical machining aimed at boosting productivity, reducing costs, and machining time, ultimately optimizing profits and yielding substantial economic benefits in the long run. The results obtained from our study exhibit promising signals and a high level of feasibility in diagnosing and verifying surface quality in machining. The number of measured samples is combined, supplemented, and provided to relevant parties to significantly increase the sample size, thereby enhancing the accuracy of the AI model and accelerating prediction capabilities through large and diverse dataset data. To enhance the reliability of these findings, it is imperative to augment the model with additional data, maximizing its effectiveness and applicability in real-world scenarios.

Key words: Convolutional neural network (CNN), Deep learning, Prediction, Surface roughness

Faculty of Mechanical Engineering, Ho Chi Minh City University of Technology (HCMUT), Viet Nam National University Ho Chi Minh City (VNU-HCM)

Correspondence

Than Trong Khanh Dat , Faculty of Mechanical Engineering, Ho Chi Minh City University of Technology (HCMUT), Viet Nam National University Ho Chi Minh City (VNU-HCM)

Email: ttkdat@hcmut.edu.vn

History

- Received: 22-08-2025
- Revised: 07-04-2026
- Accepted: 13-04-2026
- Published Online: 28-05-2026

DOI : https://doi.org/10.32508/vnuhcmj-et.v9i2.1451

Check for updates

INTRODUCTION

Surface roughness measurement is crucial in manufacturing and machining industries. Traditional methods using direct-contact tools, such as probes and sensors, are accurate but time-consuming, require machine downtime, and can cause wear on both the tool and the sample. Non-contact techniques, like laser-based systems and optical coherence tomography (OCT), offer high precision but involve complex setups, high costs, and strict environmental controls, making them impractical for real-world applications. This project addresses these challenges by utilizing a practical non-contact measurement method based on Convolutional Neural Networks (CNNs) and standard camera systems. It builds on existing research, employing deep learning models such as a 14-layer CNN, VGG19, and RESNET50, and optimizing performance using the ADAM optimizer. The project prioritizes cost-efficiency and adaptability to various lighting conditions, enabling use in typical workshop settings. By focusing on real-world feasibility, it tests actual processed samples with minimal light isolation, making it more suitable for industrial environments than lab-dependent methods.

Overall, the project offers a balanced solution, combining speed and non-destructive measurement with affordable equipment, demonstrating potential for large-scale production applications.

MATERIAL AND METHOD

Surface roughness evaluation criteria and standards

To assess surface roughness, geometric factors of the roughness profile are commonly used as criteria. These criteria are determined within the standard-length range l and are calculated relative to the mean line of the surface profile. The mean line in millimeters is referred to as the standard line.

The standard line represents the nominal profile of the surface, and within the standard-length range, it di-

Cite this article : Vo Quang B, Tan T T, T T K D. **Using Deep Learning Model in Prediction of Surface Roughness**. *VNUHCM J. Eng. Technol.* 2026; 9(2): 2849-2858.

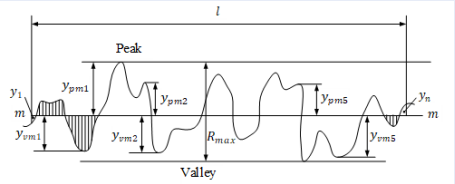

Figure 1: Overview of the evaluated profile [1,2]

vides the actual surface profile in such a way that the sum of the squared distances from points on the profile to the mean line (y_1, y_2, y_3,…,y_n) is minimized. In other words, the mean line is the line that divides the surface profile so that the total area (formed by it and the profile) on both sides of the line is equal. In other words:

$$F_1 + F_2 + \cdots + F_n = F_1' + F_2' + \cdots + F_n' \quad (1)$$

The Vietnamese National Standard (TCVN 2511 – 95) specifies two criteria commonly used to evaluate surface roughness:

Arithmetic Mean Deviation of the profile Ra: It is the arithmetic mean of the absolute values of the profile deviations (y) within the standard-length range. Profile deviation (y) is the distance from points on the profile to the mean line, measured perpendicular to the mean line.

$$R_a = \frac{1}{l}\int_0^1 |y_x| = \frac{1}{n}\sum_{i=1}^{n} |y_i| \quad (2)$$

The ten-point height of the profile Rz: It is the average value of the sum of the absolute values of the heights of the five highest peaks and the depths of the five lowest valleys of the profile within the standard-length range.

$$R_z = \frac{\Sigma_{i=1}^{5} |Y_{pmi}| + \Sigma_{i=1}^{5} |Y_{vmi}|}{5} \quad (3)$$

Pre-processing step

Data acquisition process

From the Measuring Laboratory, we obtain the data of surface roughness. The instrument is Mitutoyo Surface Roughness Measuring Instrument (Surftest SJ-310) – Serial number: 910169 with the tolerance given by Mitutoyo is upper/lower limit values for three parameters.

Setup place is shown as figure 2, and the data is captured using a smartphone camera with support light system to partly ignore the noise of nature light and enhance quality.

In order to standardize the measurement and pitch of the camera, we draw an area with a width of 1 cm and keep the camera's capture point within that 1 cm (figure 3).

Figure 2: Setup the measuring area in measuring lab

Figure 3: Standard of the workpiece

Constraint of image quality and error considerations

The factors influencing image quality were predominantly due to lighting conditions, camera specifications, and the preservation of samples. To mitigate these effects, the following strategies were implemented:

• Lighting: The experiments were conducted within a controlled laboratory environment to minimize interference from external light sources. Image capture was carried out at specific periods, namely early morning (6:00 AM to 8:30 AM) and late afternoon (4:00 PM to 6:00 PM), when natural light levels were softer and more stable. Additionally, an artificial light source was utilized to enhance brightness and ensure consistent lighting conditions.

• Camera: A standardized camera with a 12 MP resolution and a fixed focal length was used consistently throughout the process. A 10 mm × 10 mm calibration scale was incorporated in the images to maintain uniformity and facilitate accurate measurements.

• Sample Preservation: Exposure to external environmental factors may cause sample degradation, such as rusting, which can negatively impact image quality and lead to inaccuracies in AI recognition. To address this issue, two approaches were adopted: (1) reprocessing the samples to produce a new, cleaner dataset; and (2) integrating degraded samples into the dataset with a noise inclusion ratio of 5%, thereby training the AI model to recognize both pristine and noisy data accurately.

• Light noise: Using 3D edge detection, any region of the image with more than 20% overexposure was classified as noise. From this noise dataset, 5% of the noisy samples were selected for augmented training. Currently, this process is done manually.

On the other hand, some tolerances and deviations have been considered as follows.

• Measurement tolerance**:** The surface roughness measurement device was used as the standard for comparison. Additionally, the MAPE evaluation method only provides a relative accuracy metric, so the accuracy of the model is a relative comparison between the model and the measuring instrument.

• Learning process tolerance**:** During training, images were randomly selected in different iterations, so there is a probability that some samples may be trained at varying frequencies.

• Noise tolerance**:** Noise from lighting directly affects the sampling process. The more noise in the sample, the more inaccurate the result, as fewer surface texture lines will reflect light. In such cases, the model tends to learn from the noise, leading to incorrect training outcomes.

Processing phases

All samples were collected and manually measured by the authors, involving the following processes:

-Machining: The machining was performed on iron workpieces under different cutting conditions.

-Measurement: A direct contact probe was used to measure the surface roughness of the samples.

-Image Capture: A uniform camera was used throughout the sampling process.

Then, all data were compiled into a directory. A total of 8,500 samples were collected and divided into three sets with the ratio 70% for train, 15% for validation and 15% for testing. Then, all of the train and validation image files will process in the next steps:

-Image Processing: Two stages were involved in image processing:

1) Using third-party software to enhance image sharpness and crop the images to the appropriate ratio.

2) Processing the images using MATLAB code (Local Laplacian Filtering, Histogram Equalization).

Before feeding into the artificial intelligence model, the input image must be processed, which normalizes the input data, thereby improving the performance and efficiency of the model. The following are the steps for processing images before the training process.

```
for slice = 1 : length(files)
temp = files(slice).name;
temp = str2double(regexp(temp,'\d*','Match'));
temp = temp(1,1) + temp(1,2)/100;
Ydata(slice,1) = temp;
filename = fullfile(folder, files(slice).name);
thisImage = imread(filename);
[rows, columns, numberOfColorChannels] = size(thisImage);
if numberOfColorChannels < 3
message = 'Error: Image is not RGB Color!';
uiwait(warndlg(message));
continue;
end
if rows ~= nrow || columns ~= ncol
message = 'Error: Image is not 224x224!';
uiwait(warndlg(message));
continue;
end
sigmaLaplace = 0.1; alphaLaplace = 0.1;
thisImage = locallapfilt(thisImage, sigmaLaplace, alphaLaplace);
n_binsHistoEqui = 256;
thisImage = histeq(thisImage,n_binsHistoEqui);
Xdata(:,:,:,slice) = rescale(thisImage);
End
```

The Labeling is to assign labels to input images, while Image size check is to check if the size is 224 × 224 × 3, if the size is not suitable, an error will be displayed in a dialog box with the corresponding content. We use fast local Laplacian filtering (locallapfilt) to smooth (filtering) or enhance (enhancement)

image data. This function helps to filter RGB images in edge amplitude recognition. There are two important parameters for the function: sigmaLaplace - edge amplitude (amplitude of edge, [0,1]) and alphaLaplace - smoothing of details (Smoothing of details, [0,1]). The Histogram Equalization (histeq) increases the contrast of the input image by using histogram equalization. This method is often used to make images easier to read or to highlight important details in an image. The value of the equalization used is 2^8 = 256, which is equivalent to the value of a byte, making it easier to distribute (distribution) and normalize (normalization). Normalization - normalization (rescale) converts the entire value of the input image (currently has a distribution of [0;255] to [0;1].

For the model in use, it comprises 16 layers connected in a simple sequential chain and is capable of processing 104,000 input images sized 224 x 224 x 3.

Convolutional neural network (CNN

CNN is a neural network widely used in detection, consisting of overlapping convolution layers and using activation functions such as ReLU to activate weights in nodes. Convolutional layers have parameters (kernels) that have been learned to self-adjust to get the most accurate information without having to select features. Specifically designed for image and spatial data processing, CNN uses convolutional and pooling layers to effectively learn features and represent image data.

Convolutional layer, often abbreviated as Conv Layer, is an important part of the Convolutional Neural Network (CNN) architecture in deep learning. Convolutional layers are primarily used to process data with a grid structure, such as images, videos, or spatial data. It helps the neural network learn the local features of the data and reduces the number of parameters that need to be learned compared to the fully connected layer.

A Convolutional Layer consists of small filters or parameters (kernels) that are slid across the entire digitized input image matrix. These filters perform a convolution operation with the image and generate feature maps as output. These feature maps contain information about the local features of the image, such as edges, corners, or specific features.

The activities of the Convolutional Layer are used as below:

```
convolution2dLayer([3,
3],64,"Name","conv_2","Padding","same","Stride",[2,2])
```

For Filters (kernels), there usually are 3 × 3 or 5 × 5. Each filter is a small matrix containing weights that are learned by the network. These weights represent the specific pattern that the convolutional layer is trying to learn. Then, there are sliding, convolution and padding. Padding is used to maintain the size of the output feature maps and padding is often used by adding 0 values around the input image before performing convolution. With the same mode, after padding, the output pixel matrix size is the same as the input. Stride determines the distance between the positions where the filter is slid over. A larger stride will reduce the size of the output feature map.

When representing a matrix, we need two indices for rows and columns: i and j. Therefore, when representing in the form of a 3-dimensional tensor, an additional depth index k is needed. So, the index for each element in the tensor is x_{ijk}.

$$\begin{aligned} y_{11} = b + (&x_{111} \times w_{111} + x_{121} \times w_{121} + \dots \\ &+ x_{313} \times w_{313} \\ &+ x_{323} \times w_{323}) = -25 \end{aligned} \tag{4}$$

Pooling (also known as subsampling or downsampling) is used to reduce the size of feature maps after convolutional layers and reduce the complexity of the neural network while retaining important features. There are two common types of pooling, Max pooling and Average pooling. In Max pooling, each spatial region (usually a square of a certain size) of the feature map is selected, and the largest value in that region is selected to represent that region in the output feature map. Max pooling helps to retain important features and reduce the size of the feature map. In Average pooling, instead of choosing the largest value, the average of the values in the spatial region is calculated and used as the representative value. Average pooling also reduces the size of the feature map but retains a level of average information. For research purposes, Max pooling is more effective, as it selects the highest value within a kernel at each instance compared to Average pooling. This enables AI to filter out bright pixels, optimizes the reading and detection of bright points in the image, and helps the learning matrix retain only bright point values. However, this process causes the model to discard other data—data that is not the maximum within its kernel during a single movement—leading to an increase in the loss function, signifying an increase in loss value [6–8]. Consequently, Max pooling is suitable for detecting distinctive elements in the image, such as corners, edges, or bright features on a dark background, by retaining only brighter pixels in alignment with the task at hand. On the other hand, Average pooling contributes to a smoother and more refined filtering of the image.

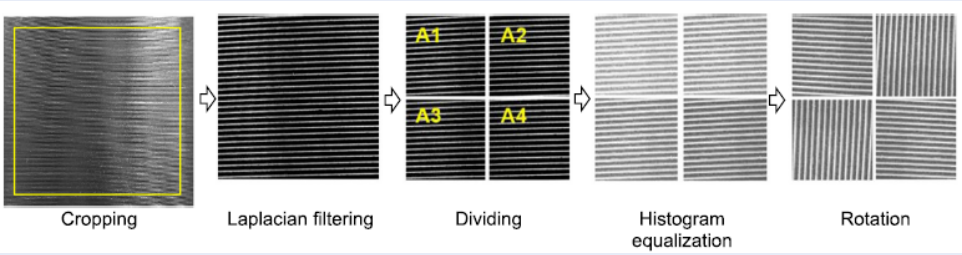

Figure 4: Refer to the steps for processing images before feeding into the model [3]

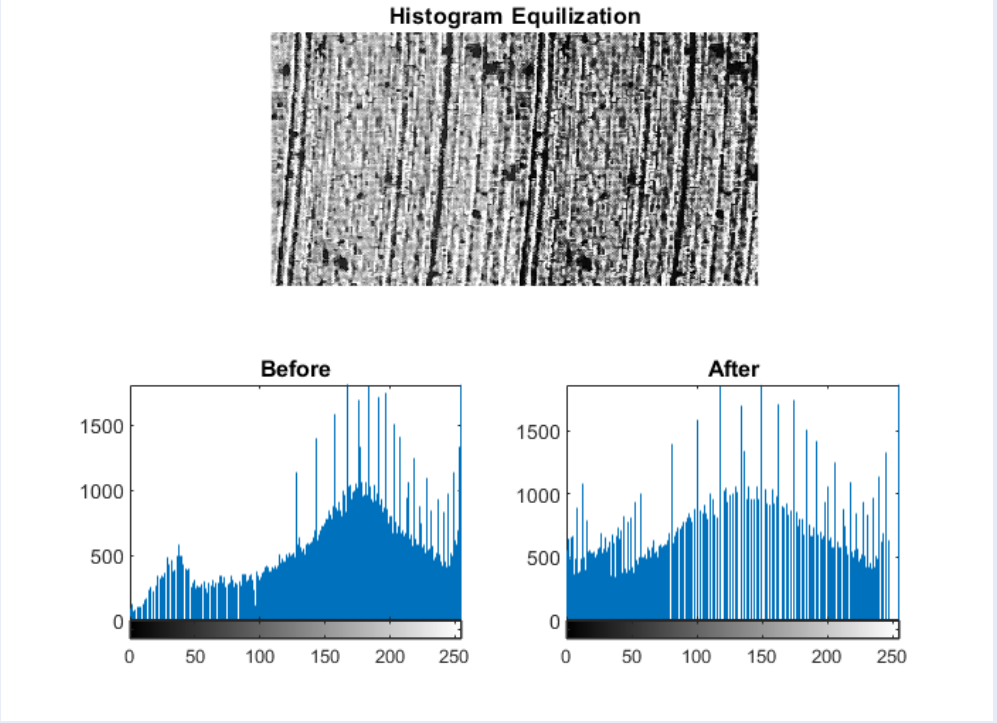

Figure 5: Comparison between before and after Histogram Equalization in MATLAB [3]

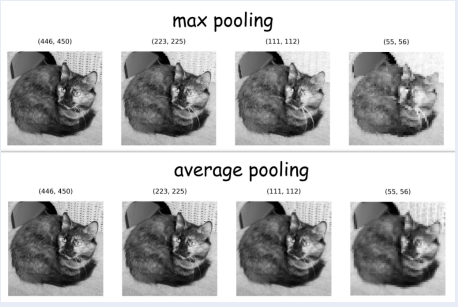

Figure 9: A Comparison between Max pooling and Average pooling [9]

In terms of methodology, there will be two types as follows: Max pooling and Average pooling.

Average pooling:

$$output_{ij} = \frac{\sum input_{ij}}{f \times f} \tag{5}$$

With kernel having the size of $f \times f$ as known as the size of the filter.

Furthermore, the formula to calculate the size of the output matrix is:

$$\left(\frac{n_h - f + 1}{s}\right) \times \left(\frac{n_w - f + 1}{s}\right) \times n_c \tag{6}$$

With the input size of $n_h \times n_w \times n_c$, f the size of kernel, s the size of stride.

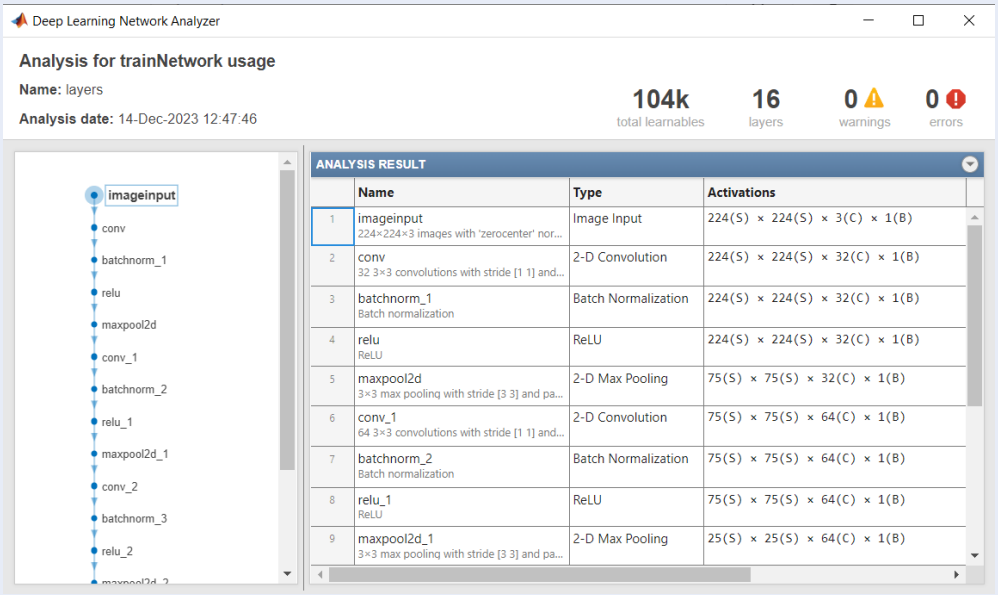

Figure 6: Window displays the analysis of the AI model and parameters for each learning layer

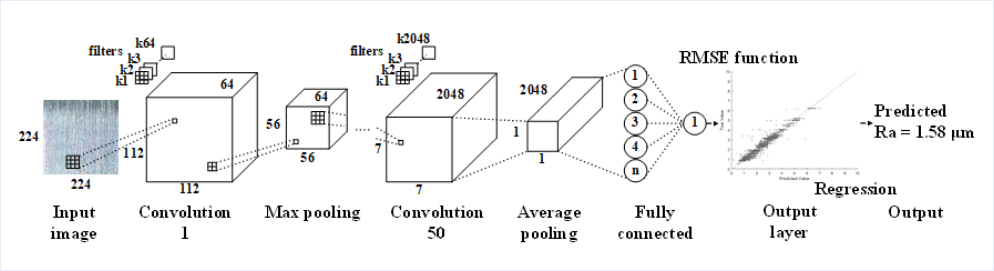

Figure 7: Description of the CNN structure [4]

For instance, considering input data size to be: 4×4, kernel size: (2×2), and stride size: (2, 2).

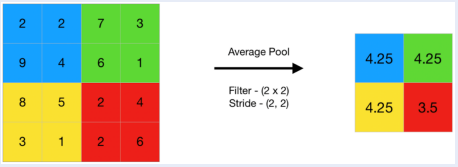

Figure 10: 10. Average pooling principle [10]

Therefore,

$$\begin{cases} output_{11} = \frac{input_{11}+input_{12}+input_{21}+input_{22}}{4} \\ output_{12} = \frac{input_{13}+input_{14}+input_{23}+input_{24}}{4} \\ output_{21} = \frac{input_{31}+input_{32}+input_{41}+input_{42}}{4} \\ output_{22} = \frac{input_{33}+input_{34}+input_{43}+input_{44}}{4} \end{cases}$$

Max pooling:

$$output_{ij} = max(input_{i',j'}; input_{i'(j'+1)}; ...; input_{(i'+n)(j'+n)}) \quad (7)$$

Like the input matrix, the formula to calculate the size of the output matrix is:

$$\left(\frac{n_h - f + 1}{s}\right) \times \left(\frac{n_w - f + 1}{s}\right) \times n_c \quad (8)$$

With the input size of $n_h \times n_w \times n_c$, f the size of kernel,s and the size of stride.

For instance, considering input data size to be: 4×4, kernel size: (2×2), and stride size: $(2, 2)$.

$$\begin{aligned} output_{11} &= max(input_{11}; input_{12}; input_{21}; input_{22}) \\ output_{12} &= max(input_{12}; input_{14}; input_{23}; input_{24}) \\ output_{21} &= max(input_{21}; input_{22}; input_{41}; input_{42}) \\ output_{22} &= max(input_{33}; input_{34}; input_{43}; input_{44}) \end{aligned}$$

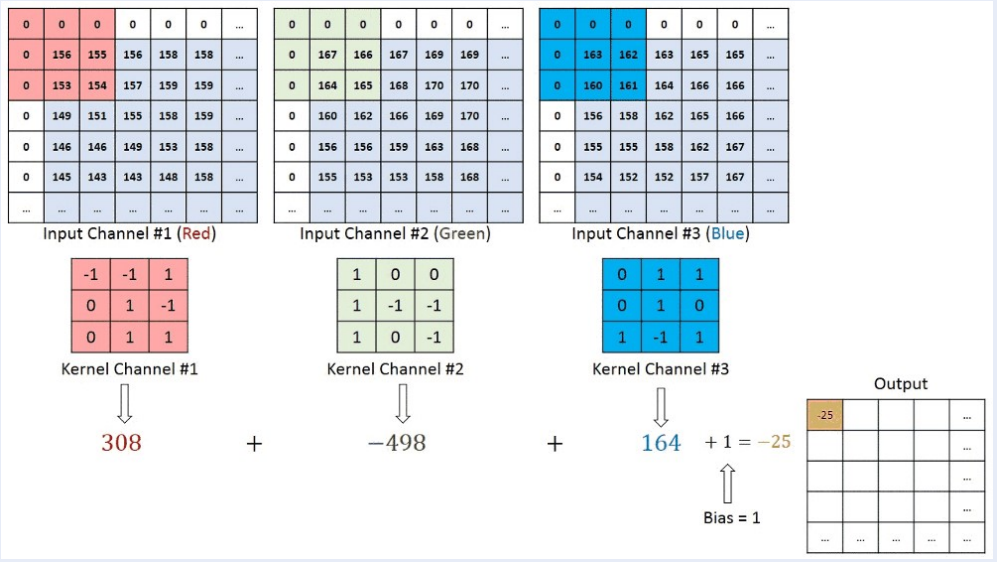

Figure 8: Convolutional layer principle[5]

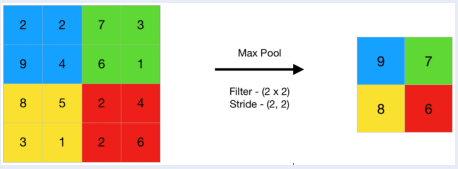

Figure 11: 11. Maximum pooling principle[10]

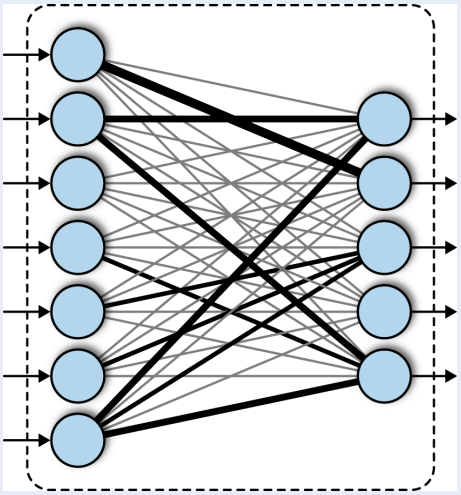

Figure 12: Visualization of FC Layer[11, 12]

Fully Connected Layer (fully connected layer), often called FC Layer or Dense Layer:

This layer is used to connect all inputs from the previous layer to all outputs in that layer, creating a fully connected neural network. This means that each input in the previous layer will be connected to all outputs in that layer.

Training

After completing the above 3 steps (data, layer, trainingOptions), we will start training using the trainNetwork function as follows:

net = trainNetwork(XTrain,YTrain,layers,options);

With XTrain is the processed data set with the YTrain set is the corresponding label set. Layers is the neural network model. Options are the learning settings.

During training, MATLAB displayed the following charts:

The chart will display the following notable parameters:

RMSE (Root Mean Squared Error): RMSE is a common error metric used in regression tasks or when predicting numerical values. It measures the average deviation between the predicted value of the model and the actual value (known). RMSE is calculated by taking the square root of the mean of the squared errors (errors are the difference between predictions and reality).

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \quad (9)$$

With:

$\hat{y}_i$: Predicted value

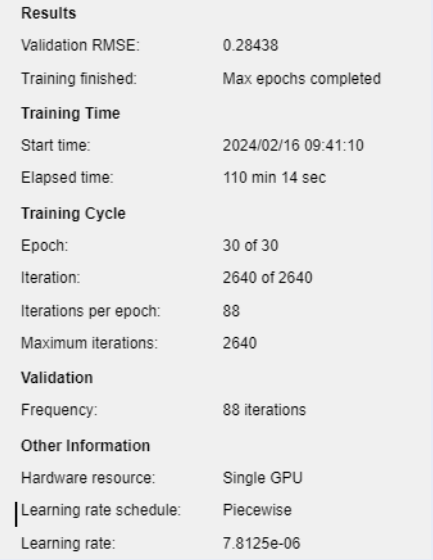

Figure 13: Window that displays the learning process of the neural system

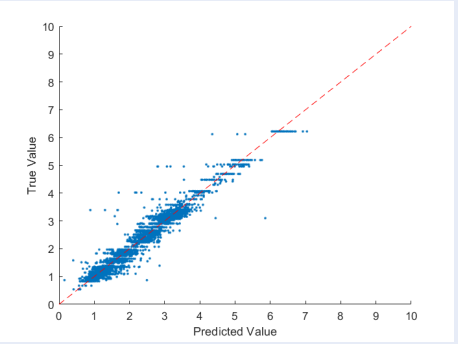

Figure 14: Result of predict function

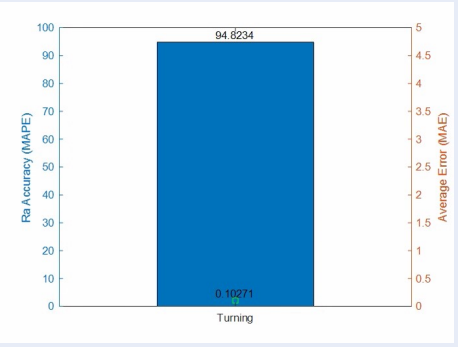

Figure 15: Evaluation methods (MAPE and MAE)

y_i :Actual value

n: Quantity

Loss (Loss function): Loss is a quantity that the model tries to minimize during training. In deep learning, loss is often a mathematical function that describes the difference between the predicted value and the actual value. The goal of the training process is to find the parameter values of the model such that the loss is as small as possible.

Common loss functions in deep learning include Mean Squared Error (MSE) for regression and Cross-Entropy Loss (or log loss) for classification. When training a model in MATLAB, these two parameters help the developer track and check the performance of the model. We will try to bring the two parameters RMSE and loss close to 0, meaning that the model's predictions are more accurate and suitable for the actual test data.

RESULTS AND DISCUSSION

Using the predict function to train input values XValidation and output values YPredicted (predicted values), then matching them with the corresponding Yvalidation for the given XValidation (true values), visualizing these values, we have the following graph.

The dashed red line represents the desired correlation between actual and predicted values.

Evaluation methods (statistical models): MAPE and MAE

MAPE (Mean Absolute Percentage Error) or MAPD (Mean Absolute Percentage Deviation) is a commonly used measure to evaluate the accuracy of numerical predictions in forecasting and prediction tasks. MAPE calculates the average percentage deviation (non-negative) between the predicted values and the actual values. A lower MAPE value indicates higher accuracy in the predictive model. The advantages of using Mean Absolute Percentage Error (MAPE) in statistical evaluation include its ease of understanding and interpretation, as it expresses errors as a percentage relative to actual values, making model comparisons straightforward. MAPE is unit-independent, allowing it to be applied across datasets with varying scales and units. Additionally, it effectively identifies poor model performance, as large errors significantly impact the metric. Finally, MAPE aids in analyzing factors contributing to errors and supports model optimization to improve accuracy.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \widehat{y_i}}{y_i}\right| \quad (10)$$

With:

$\widehat{y_i}$: the predicted value

y_i : the actual value

n: the number of data points in the prediction set

MAE (Mean Absolute Error) is a commonly used error measurement to assess the accuracy of predictions in forecasting tasks. MAE computes the average deviation between the predicted values and the actual values in absolute terms (non-negative). This means MAE measures the average difference between predictions and actual values without considering the direction (positive or negative) of the deviation. MAE is a common error measurement in machine learning and statistics. A lower MAE value indicates higher accuracy in the predictive model, meaning that the model's predictions are more accurate and closer to the actual values.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (11)$$

CONCLUSION

The statistical analysis reveals both achievements and areas for improvement in the roughness diagnostic AI model. The diagnostic accuracy for extrapolation cases currently reaches 66.17%, with some predictions aligning with established linear trends, demonstrating their appropriateness. However, limitations arise due to inadequate data volume and suboptimal image quality, underscoring the need for a more comprehensive and reliable dataset. Furthermore, the model exhibits a tendency to overfit, which can be attributed to its high complexity relative to the limited data available. To ensure consistency in future diagnostics under uniform imaging conditions, a protocol has been established requiring 15 training images, 2 for validation, and 1 for testing.

Significant progress has been made, evidenced by a notable improvement in AI accuracy since December 2023, with interpolation accuracy increasing from 70% to 94% and extrapolation accuracy from 40% to 57%. Additional advancements include the systematic research and classification of data repository components, as well as the development of a user-friendly software interface integrated with the AI model. Nevertheless, certain challenges remain, such as the influence of external factors on measurement accuracy despite enhancements in image processing, the persistence of overfitting, and the ongoing requirement for a more robust and reliable training dataset.

ABBREVIATIONS

- AI: Artificial Intelligence
- CNN: Convolutional Neural Networks
- MAPE: Mean Absolute Percentage Error
- MAE: Mean Absolute Error
- MAPD: Mean Absolute Percentage Deviation
- MSE: Mean Squared Error
- RMSE: Root Mean Squared Error

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest or financial disclosures to report, and this article does not involve any studies with human or animal subjects conducted by the authors.

AUTHORS' CONTRIBUTIONS

Tai T. T. contributed to the conceptualization and methodology of the study. Bao V. Q. was responsible for data collection and performed the data analysis. Dat T. T. K. supervised the research and contributed to manuscript review and editing. All authors have read and approved the final version of the manuscript.

ACKNOWLEDGMENT

We acknowledge the support of time and facilities from Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for this study.

REFERENCES

1. Ton ND. Dung sai và Lắp ghép. Vietnam: Vietnam Education Publishing House Limited Company; 2020.
2. Thomas TR. Characterization of surface roughness. Precision Engineering. 1981;3(2):57–120.
3. Rifai AP, Aoyama H, Tho NH, Dawal SZM, Masruroh NA. Evaluation of turned and milled surfaces roughness using convolutional neural network. Journal Pre-proofs. 2020;.
4. Quattoni A, Caruana CM. Transfer learning for image classification with sparse prototype representations. IEEE Conf on Computer Vision and Pattern Recognition. 2008;p. 1–8.
5. Nttuan8; 2024. Available from: https://nttuan8.com/bai-6-convolutional-neural-network/#Fully_connected_layer.
6. Theckedath D, Dhananjay RR; 2020.
7. Yosinski J, Yosinski CJ; 2008.
8. Akram WM, Akram LG. A CNN based automatic detection of photovoltaic cell defects in electroluminescence images. Energy. 2019;p. 189–189.
9. Digitalocean OPB, Olu-Ipinlaye; 2024. Available from: https://blog.paperspace.com/pooling-in-convolutional-neural-networks/.
10. Savyakhosla; 2024. Available from: https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/.
11. Tammina S. Transfer learning using VGG-16 with Deep Convolutional Neural Network for classifying images. Int J Sci Res Publ. 2019;.
12. Mathworks; 2024. Available from: https://www.mathworks.com/discovery/convolutional-neural-network.html.

VNUHCM Journal of Engineering and Technology 2026, 9(2): 2849-2858

Open Access Full Text Article

Bài báo nghiên cứu

Ứng dụng mô hình học sâu trong dự đoán độ nhám bề mặt gia công

Tô Tấn Tài , Võ Quang Bảo , Thân Trọng Khánh Đạt *

Use your smartphone to scan this QR code and download this article

TÓM TẮT

Trong lĩnh vực công nghiệp hiện đại, độ chính xác gia công đã có những bước tiến vượt bậc, đặt việc đo lường độ nhám bề mặt trở thành tiêu chí quan trọng trong đánh giá chất lượng gia công. Tuy nhiên, các phương pháp đo lường hiện tại thường tốn nhiều thời gian và công sức, đồng thời có nguy cơ gây sai lệch do yêu cầu tháo mẫu khỏi thiết bị gia công. Để giải quyết những thách thức này, nghiên cứu của chúng tôi hướng đến việc tối ưu hóa quy trình kiểm tra sau gia công bằng cách sử dụng hình ảnh bề mặt gia công để đánh giá độ nhám. Bài báo này trình bày ứng dụng các mô hình học sâu, được hỗ trợ bởi phần mềm MATLAB, để dự đoán độ nhám bề mặt một cách chính xác và hiệu quả. Chúng tôi cũng đề xuất một phương pháp đo lường toàn diện, nhằm định hướng phát triển các giai đoạn tiếp theo của nghiên cứu. Sử dụng các phương pháp đo lường không tiếp xúc, nghiên cứu không chỉ đảm bảo độ chính xác cao mà còn phù hợp với quy trình gia công, giúp tăng cường năng suất, giảm chi phí và thời gian sản xuất, tối ưu hóa lợi nhuận về lâu dài. Kết quả nghiên cứu cho thấy tiềm năng lớn trong việc dự đoán và đánh giá chất lượng bề mặt, với khả năng mở rộng quy mô và nâng cao độ chính xác của mô hình AI thông qua việc bổ sung dữ liệu và tăng cường mẫu đo. Điều này hứa hẹn mang lại những lợi ích kinh tế đáng kể và ứng dụng rộng rãi trong thực tiễn công nghiệp.

Từ khoá: Convolutional neural network (CNN), Deep learning, Prediction, Surface roughness

Khoa Cơ khí, Trường Đại học Bách khoa Thành phố Hồ Chí Minh, Đại học Quốc gia Thành phố Hồ Chí Minh

Liên hệ

Thân Trọng Khánh Đạt , Khoa Cơ khí, Trường Đại học Bách khoa Thành phố Hồ Chí Minh, Đại học Quốc gia Thành phố Hồ Chí Minh

Email: ttkdat@hcmut.edu.vn

Lịch sử

- Ngày nhận: 22-08-2025
- Ngày sửa đổi: 07-04-2026
- Ngày chấp nhận: 13-04-2026
- Ngày đăng: 28-05-2026

DOI : https://doi.org/10.32508/vnuhcmj-et.v9i2.1451

Check for updates

Trích dẫn bài báo này: T T T, V Q B, T T K D. **Ứng dụng mô hình học sâu trong dự đoán độ nhám bề mặt gia công**. *VNUHCM J. Eng. Technol.* 2026; 9(2): 2849-2858.