



Prediction of micro-hardness in thread rolling of St37 by convolutional neural networks and transfer learning

Mehdi Soleymani¹ · Mohammad Khoshnevisan² · Behnam Davoodi³

Received: 11 July 2022 / Accepted: 21 October 2022 / Published online: 10 November 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract

This study introduces a non-destructive method by applying convolutional neural networks (CNN) to predict the micro-hardness of the thread-rolled steel. Material microstructure images were collected for our research, and micro-hardness tests were conducted to label the extracted microstructure images. In recent years, researchers have used machine learning (ML) and deep learning (DL) models to predict material properties for forming, machining, additive manufacturing, and other processes. However, they encountered industrial limitations primarily because of the absence of historical information on new and unknown materials, which are necessary to predict material properties by DL models. These problems can be solved by employing CNN models. In our work, we used a CNN model with two convolutional layers and visual geometry group (VGG19) as transfer learning (TL). We predicted four classes of micro-hardness of the St37 rolled threads. The prediction results of the micro-hardness test images by our proposed CNN model and pre-trained VGG19 model are comparable. Our proposed model has produced the same precision and recall scores as VGG19 for class B and class C hardness. VGG19 performed slightly better than our model for precision in class A and recall in class D. We observed that the training time of our proposed model using the CPU (central processing unit) was approximately nine times faster than the VGG19 model. Our proposed CNN and VGG19 have direct applications in advanced manufacturing (AM). They can automatically predict the micro-hardness in the thread rolling of St37. Our proposed model requires less memory and computational power and can be deployed more efficiently than the VGG19 model.

Keywords Micro-hardness · Metallography · Convolutional neural networks · Deep learning · Machine learning · Mechanical properties · Microstructure

1 Introduction

Rotational forming is used to form cylindrical bars, pipes, and metallic sheets by compressive rotation. This process is the most common method for producing bolts and nuts

and creating gears, balls, rings, cylindrical and conical hollow parts, shafts, and other axial symmetrical cross sections. Generally, threads are produced by different processes, such as machining, grinding, and thread rolling processes. The thread rolling process has some advantages, like good mechanical properties, and can make threads on small and big diameters. This cold working process does not produce chips [1, 2]. Deform software has simulated the strain distribution and material flow in the thread rolling process. The effect of friction coefficient on strain and hardening has been investigated in ACME and unified threads [3, 4]. Stress distribution was studied on metallic-glass threads rolled by the thread rolling process. The results showed that the cold work due to the thread rolling process improved the mechanical properties by creating residual stress [5, 6]. Another study reported increasing tensile strength in specimens after the thread rolling process [7]. The thread rolling process increases the precision of the thread, improves

✉ Mohammad Khoshnevisan
m.khoshnevisan@northeastern.edu

Mehdi Soleymani
soleimani.mehdi@ut.ac.ir

Behnam Davoodi
bdavoodi@iust.ac.ir

¹ School of Mechanical Engineering, College of Engineering, University of Tehran, Tehran, Iran

² Physics Department, College of Science, Northeastern University, Boston, MA 02115, USA

³ School of Mechanical Engineering, Iran University of Science and Technology, Tehran, Iran

surface roughness, and increases the strength of threaded joints [8]. Roll forming and machining of threads on carbon steel revealed a noticeable increase in hardness in the roll-formed threads. This is due to the work-hardening phenomenon resulting from the rolling [9]. Machine learning has been employed in science and engineering [10]. Many processes in manufacturing screws depend on senior technicians and their experience. In recent years, the reliance on technicians' expertise has been reduced. Machine learning/deep learning techniques have been applied to manufacturing processes, including the traditional screw processes. An artificial neural network (ANN) was used to predict the physical properties of screws after heat treatments. A physical property estimation model was employed to determine the appropriate control parameters in the screw manufacturing process [11]. ANN has assisted in accurately predicting the mechanical properties of rolling materials [12]. Indeed, ANN has allowed researchers to create models to solve complex problems without having mathematical limitations [13]. The ANN was used for controlling the hot-rolling of BHP Steel Co [14]. Moreover, ANN has been used to model Al-Si-Cu alloys' mechanical performance for predicting mechanical properties such as hardness, yield strength, and elastic limit [15]. In another study, ANN (multi-layer perceptron) was used to obtain the hardness of AZ91 magnesium alloy. In this study, the aging temperatures and aging times were the inputs, and the material hardness was the computed output [16].

The application of ANN for predicting the hardness properties of Sn-9Zn-Cu solder alloy was studied. The training and validation data in the ANN were obtained from experimental results [17]. An ANN was used to predict residual stress and hardness from ball burnishing [18]. A total of 24 ball burnishing tests were conducted under different pressures, yield stress, residual stress, and hardness. In another work, two ANN and nonlinear regression methods were compared to estimate material tensile strength, elongation, and hardness of equal channel angular rolled Al-5083 sheets. The errors of the ANN model were significantly lower than the nonlinear regression model [19]. Other researchers used the ANN to predict the mechanical and metallurgical properties such as surface roughness, hardness, and residual stress of steel 316L [20]. In [21], the authors used an ANN prediction model of residual stress and hardness for ultrasonic nanocrystal surface modification (UNSM) parameter optimization on nickel alloys. The surface hardness parameters of Shot Peening on AISI 1017, 1045, 1050, 1060, and 1070 were studied experimentally. Microstructural studies were conducted with optical and transmission electron microscopes in this research. Furthermore, the micro-hardness investigation was implemented to record the surface hardness. Also, the surface hardness of the treated materials by the Shot Peening process was analyzed by ANN [22].

It should be noted that microstructure forms of elements, like shape and their distribution, are often significant parameters in modeling micromechanical behavior [23]. Machine learning has been able to recognize and classify microstructure images. This research used a support vector machine (SVM) to extract and predict microstructure specifications and morphological features [24]. A convolutional neural network (CNN) for predicting the mechanical properties of hot-rolled steel using chemical composition and process parameters was implemented [25]. This research aimed to introduce the CNN prediction for the steel properties after transforming the production data into two-dimensional images. In recent years, CNN has been attractive in industrial applications. There is no need to extract the features in CNN manually. It has some advantages in comparison with ANNs. CNN reduces the number of trainable parameters and provides faster convergence. CNN also reduces dimensionality by down-sampling [26]. Indeed, CNNs can be trained efficiently because they have fewer connections and parameters than regular neural networks [27]. In thread rolling, the hardness of the threads is changed due to work hardening. The work hardening is accompanied by changes in material microstructure and microstructure and can be extracted from microstructure images. Traditional machine learning methods have been successfully applied to recognize and predict mechanical properties. The ANN models have opened new doors to predict material properties expeditiously, precisely, and non-destructively. It is important to note that micro-hardness values of material severely vary at different material positions due to the heterogeneous distribution of material phases. In recent years, machine learning algorithms have been used to recognize micro-hardness by defining the chemical percentage of elements, previous process conditions, and their equivalent micro-hardness in training data [15–19, 21, 22]. However, from the industrial point of view, the previous process conditions may be unknown when examining a new material. Thus, this solution may not be useable. Moreover, defining the percentage of elements is too expensive and time-consuming, especially when obtaining micro-hardness values for different points in the material is necessary. Furthermore, training an ANN model with various material parameters such as element percentages, previous conditions, and the micro-hardness of any position is also costly and time-consuming. Furthermore, obtaining micro-hardness needs an expert to analyze the results. They may not be accessible or unable to analyze large datasets accurately and promptly. However, CNN can overcome this problem and automate the micro-hardness test procedure. In our work, we propose a CNN model to predict the micro-hardness of the threads rolled. We also compare our results with a pre-trained VGG19 model. Our models were trained with the material microstructure images obtained from metallography.

2 Materials and methods

2.1 Thread rolling process and micro-hardness

Figure 1 shows the thread rolling machine. Figure 2 shows a sample of threads after the thread rolling process on St37 steel before metallography tests. The thread rolling process conditions are presented in Table 1.

In order to take images of material microstructure by optical microscope, polishing and etching were initially conducted on the threaded samples. Then, the microstructure images were obtained by metallographic microscope and saved in a png format in high resolution. Finally, the micro-hardness was recorded by a micro-hardness tester. The results are illustrated in Fig. 3.

2.2 Convolutional neural network model

Convolutional neural network (CNN) is a deep learning neural network (DLNN) commonly employed for various computer vision tasks. CNNs can classify, cluster images, and recognize objects within an image. CNNs are widely used in physics, engineering, and medical applications. In CNN, the pooling layers decrease the dimensionality of the input data and the computational scale [28]. A CNN model

contains two parts, feature extraction and prediction parts. The predicted micro-hardness labels are the output of the last layer in the present model. Figure 4 is a schematic of a CNN model. The microstructure images of the threads will be the input layer. A set of convolutional filters is used for the convolution operation. In the convolution procedure, each convolution kernel provides a feature map. Parameter sharing is an essential characteristic of CNN. It decreases the number of parameters and extracts features efficiently [28, 29].

The output nodes of the fully connected layer equal the number of classes [30]. Adam and categorical cross-entropy have been applied in our model as the optimizer and loss function, respectively. We built our CNN model using the python programming language (version 3.9.7) [31] and the TensorFlow platform (version 2.7.0) [32]. In our research, 160 images were used for training and 26 for testing. We employed two convolutional layers + (Rectified Linear Unit) ReLU activation function, two max-pooling layers, two dense layers + ReLU and one dropout, and a Softmax activation function. The ReLU activation function is mathematically defined as

$$f(x) \equiv \max(0, x) \equiv \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \quad (1)$$

Fig. 1 Thread rolling machine and dies

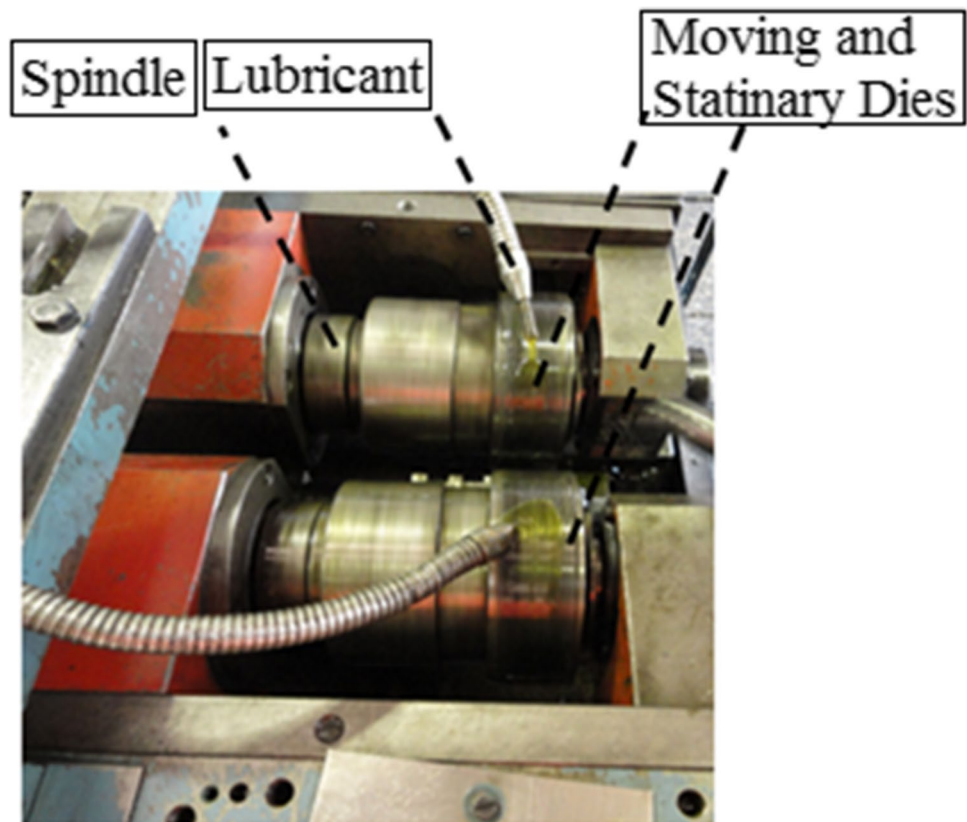


Fig. 2 Cross section of the threads produced by thread rolling and metallography images

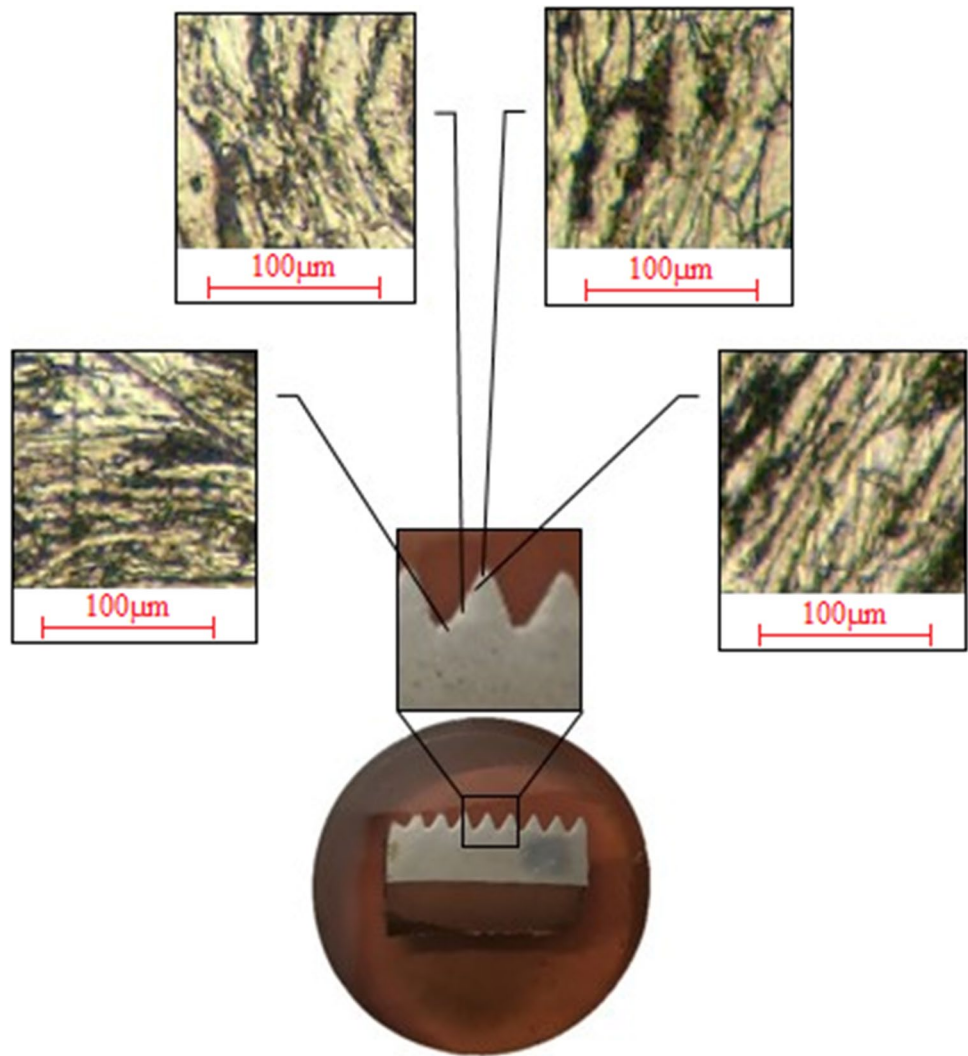


Table 1 Process and material conditions

Work-piece material	Work-piece diameter (mm)	Thread pitch (mm)	Rotational speed (rpm)	Lubricant	Die material
St37	46.1	2	22	Oil 10	D2 Steel

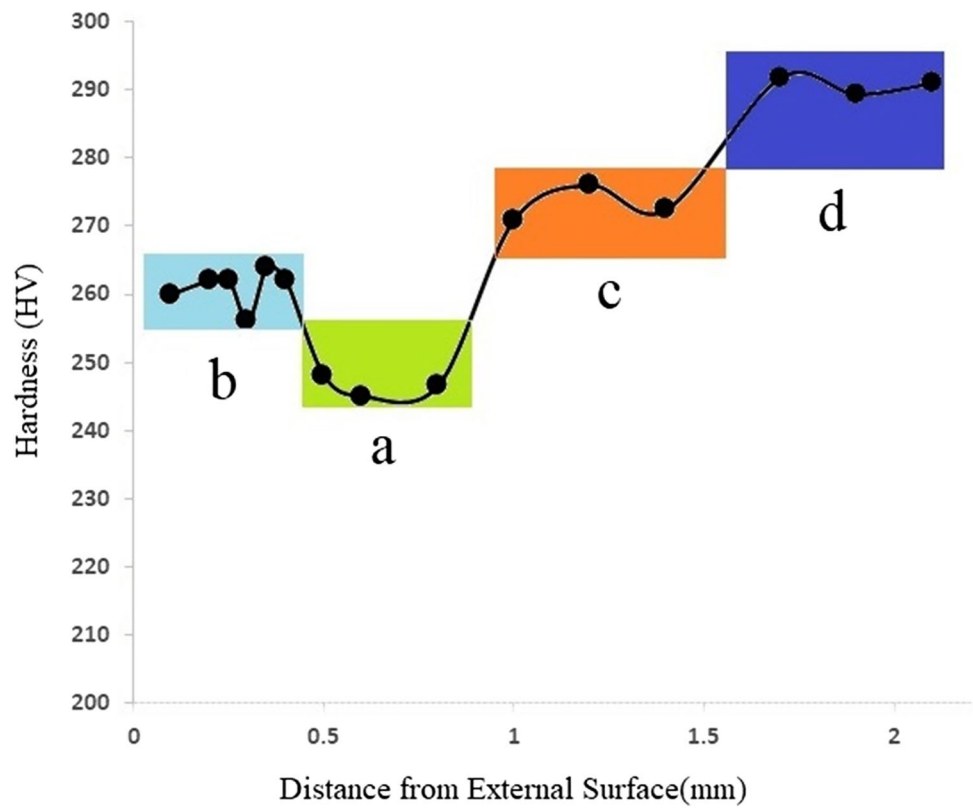
Figure 5 elucidates the workings of the convolutional neural network with a simple example. In the first step, this example applies a filter size of (3×3) to the image. Then, elementwise multiplication between the image’s pixel values and the filter is performed. Finally, all the values are summed. This process will continue for all the cells. In this example, we assumed $\text{stride} = 1$. Stride is the number of pixels the kernel window will slide at each convolution step. Below is the computation for two cells.

$$\begin{aligned}
 &(0 \times 0 + 0 \times 0 + 1 \times 1) \\
 &+ (0 \times 1 + 0 \times 1 + 0 \times 0) \\
 &+ (0 \times 0 + 1 \times 1 + 1 \times 0) \equiv 2
 \end{aligned} \tag{2}$$

$$\begin{aligned}
 &(0 \times 0 + 0 \times 0 + 1 \times 1) \\
 &+ (0 \times 1 + 1 \times 1 + 1 \times 0) \\
 &+ (0 \times 0 + 1 \times 1 + 1 \times 0) = 3
 \end{aligned} \tag{3}$$

Pooling layers are constructed to reduce the size of the image across layers. This operation selects the maximum value in each window and creates a pooled feature map. In the final step, we flatten the output of the pooled feature map into a column vector and connect it to the final classification model with the number of classes. It should be noted that a typical kernel/filter size in most CNN applications is a 3×3 filter [33].

Fig. 3 Micro-hardness results and labels of regions (a, b, c, and d)



2.3 Input data

After micro-hardness testing, the micro-hardness values were classified into four classes (a, b, c, and d), as shown in Fig. 3. Then, each class was labeled from zero to three based on its average micro-hardness. The microstructure images with 100*100 pixels were extracted from primary images taken from an optical microscope with 2560*2048 pixels. They were labeled according to their micro-hardness and used as our training set. Table 2 summarizes their labels, numbers, and their equivalent micro-hardness.

3 Results and discussion

3.1 Metric for deciding our CNN model

Accuracy is a metric used in classification problems. It is obtained by dividing the number of correct predictions by the total number of samples. In the best-case scenario, the accuracy for training and validation datasets should converge to 1. Since we had limited images in our work, we used image augmentation (IA) to improve our validation metric. The animation of learning curves at the New York Institute and Laboratory for Artificial Intelligence illustrates the evolution of training accuracy and validation accuracy [34].

Loss is the sum of the errors for each sample in training or validation datasets. A decrease in a loss function implies that the validation and training errors are getting smaller. Hence, there is improvement in the predictability of our CNN model. In an ideal situation, the training loss and validation loss should ultimately converge to zero. However, in many real-world applications, this may not be possible. If our model validation loss increases while our training loss decreases, we have an overfitting problem. The loss function for our model is categorical cross-entropy (CCE) since we have a multi-class classification problem (four classes). CCE is the combination of Softmax activation and cross-entropy loss and is defined as:

$$CCE = - \sum_{i=1}^C t_i \log (f(s)_i) \tag{4}$$

$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \text{ is a Softmax activation function} \tag{5}$$

where,

f is a Softmax function

s_i is the raw output i

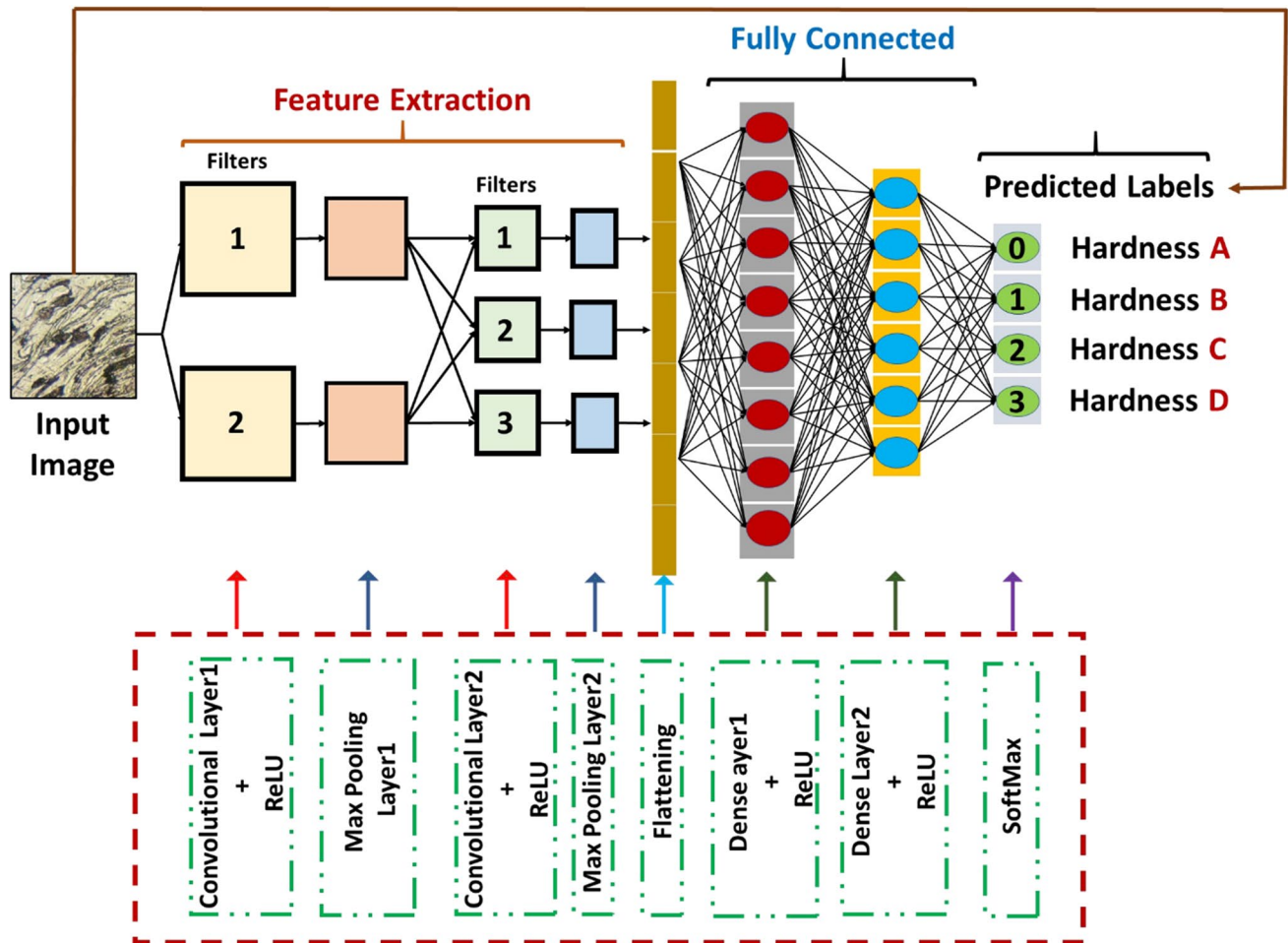


Fig. 4 Schematic of a CNN model

s_j is the raw output j

c is the total number of classes

t_i is the true label class

If the predicted probability of our sample deviates from the true label, then the cross-entropy loss will increase. For instance, if our model predicts a probability of 0.1 when the true label is 1, then the cross increases. On the other hand, if our model predicts a probability of 0.8 when the true label is 1, the cross-entropy loss decreases.

3.2 Convolutional layer, image augmentation, filter number, and dropout rate effects

CNN models with several layers can be trained to detect complex features. More convolutional layers can detect complicated features from input images. However, this does

not necessarily mean that more convolutional layers will improve prediction accuracy. Indeed, more convolutional layers have more trainable parameters, requiring more computational resources. Image augmentation (IA) is a technique employed in computer vision tasks. In many cases, providing big input data is not possible. IA applies transformations to all the images and then uses the original and transformed images for training. For instance, we can use shift, rotation, and horizontal flips. IA assists us in increasing the overall accuracy of our model and minimizing overfitting, particularly when we have limited examples for our training data. Keras provides the [ImageDataGenerator] class [35, 36]. We applied transformations as we trained our model. The dropout technique turns off some neurons during training, which means the model does not consider these neurons. We used dropout in our first dense layer. Dropout randomly puts a fraction of neurons in a particular layer to zero. Dropout is not used when our model makes an inference for the test images. Convolutional filters are used to extract information from an input image. In many CNN applications, the number

Fig. 5 Convolutional neural network—simple example

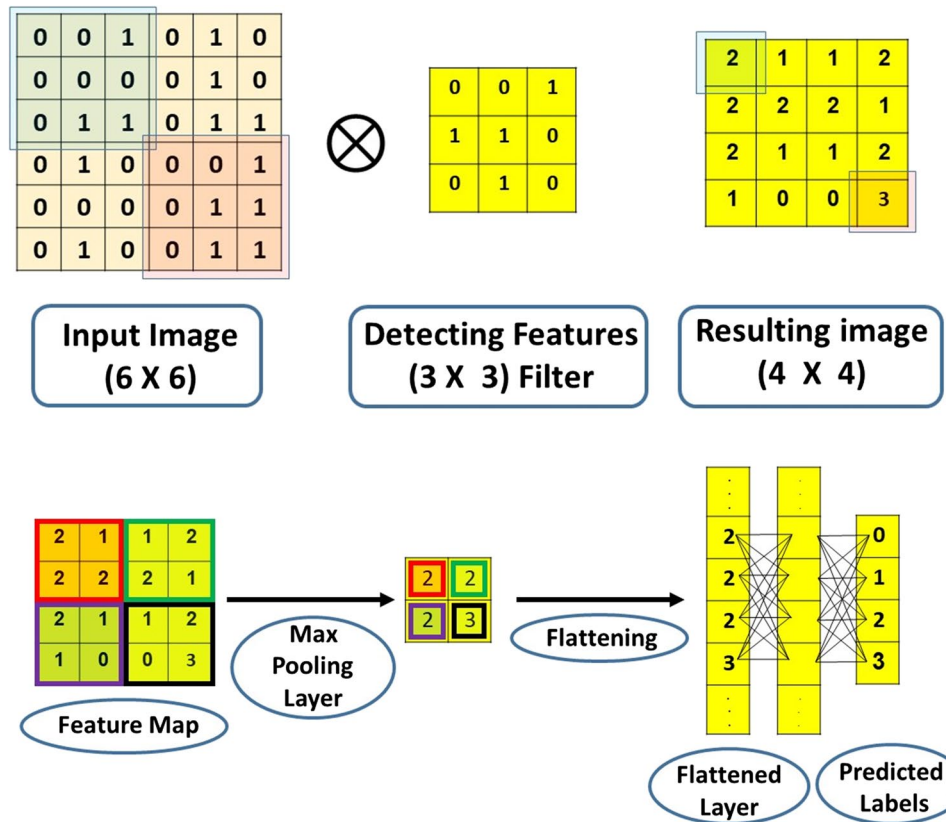


Table 2 Micro-hardness range and the equivalent labels

Region according to Fig. 3	Micro-hardness (HV)	Number	Equivalent label
Region “a”	245–255	40	0
Region “b”	255–265	40	1
Region “c”	265–275	40	2
Region “d”	275–295	40	3

of convolutional kernels will also increase when the network depth increases. Convolutional kernels are learned during training [37].

In our study, we built three CNN models with two convolutional layers using width shift parameter = 0.05, 0.4, and 0.6 and number of filters = 16–64, 16–32, and 32–32 and dropout rate = 0.05, 0.1, and 0.06. We also tested our CNN model with three convolutional layers using width shift parameter = 0.05, number of filters = 16–64–64, and dropout rate = 0.05. Our model with three convolutional layers did not yield better results. Hence, we used two convolutional layers. Through many trials, we obtained the best width shift parameter, the number of filters, and the dropout rate. We then computed the validation metric for all the models. Our results are illustrated in Fig. 6

and Table 3. We chose model A with two convolution layers using width shift parameter = 0.05, number of filters = 16–64, and the dropout rate = 0.05. Model A’s validation accuracy is more than other models (B, C, and D) and has the lowest validation loss.

We observed fluctuations during our training time because we had a small dataset with 160 images for training. Indeed, the oscillations can be minimized, and the validation accuracy of our CNN model can be improved by using more training images.

4 Transfer learning and VGG model

Training a model from scratch is often difficult because it warrants many training examples. Transfer learning is a technique that assists us in training our model with a few training examples by utilizing a pre-trained model such as NASNet, ResNet, VGG, EfficientNet, and Inception. In this process, we first chose a pre-trained model, which in our case is VGG19.

The model of the Visual Geometry Group at Oxford University, also known as the VGG model, is a deep-CNN architecture with 16 and 19 layers [38]. The VGG researchers examined the effect of the convolutional network depth on the ImageNet Challenge dataset in 2014. The ImageNet

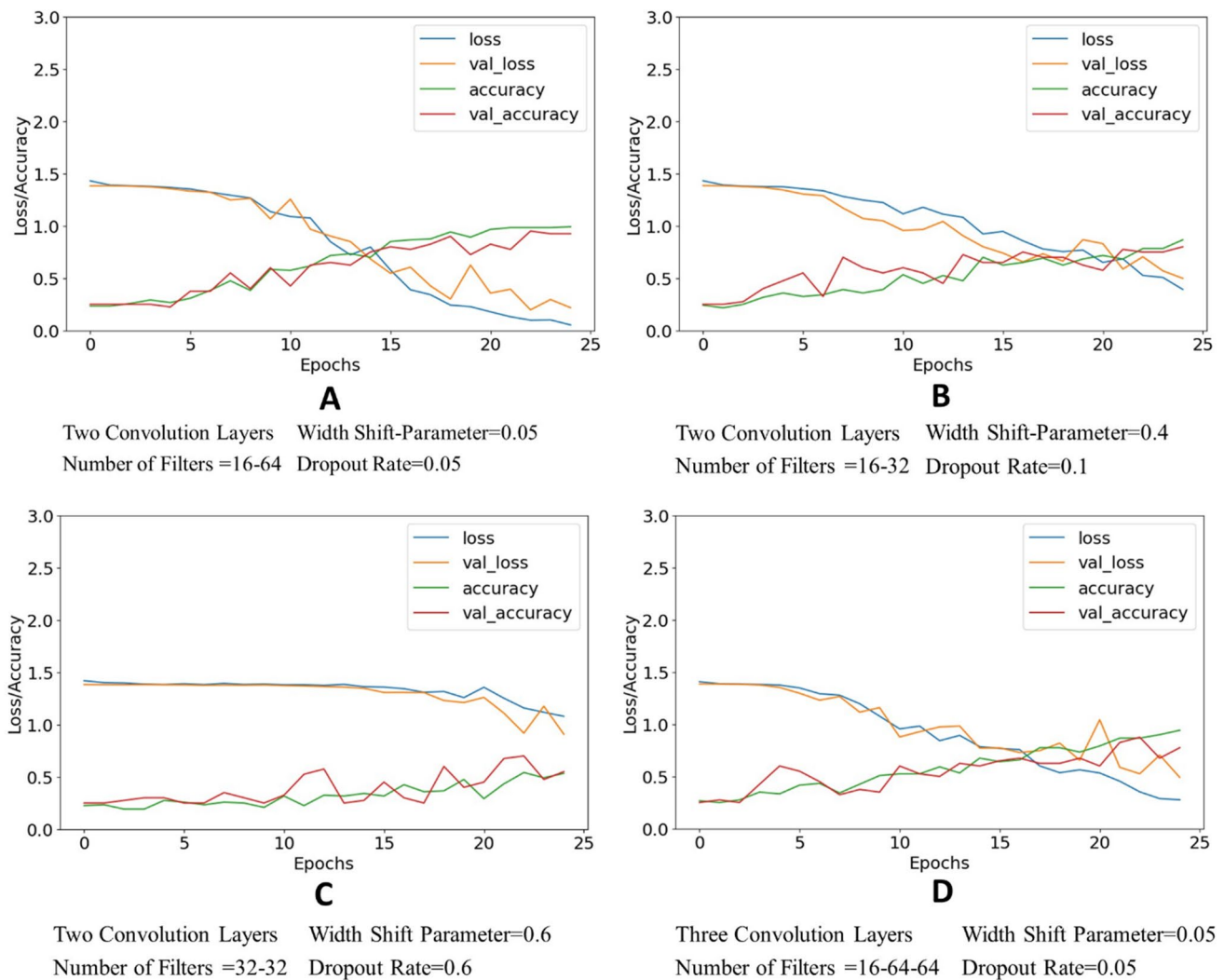


Fig. 6 Convolutional layers with image augmentation, filter number, and dropout rate

Large Scale Visual Recognition Challenge (ILSVRC) was established in 2010 to improve object detection and image classification tasks on a large scale [39]. VGG19 is a pre-trained model accessible in Keras with Theano and TensorFlow backends. We employed the pre-trained model (VGG19) as a feature extractor. We then froze all the layers of the pre-trained model and included our classifier on top of the pre-trained model. We finally trained our model with

our micro-hardness dataset with four classes and obtained predictions [40]. The input size image for the VGG model is 224×224 . Figure 7 shows the architecture and configuration of VGG19. Figure 8 is the loss and accuracy plots for our training and validation sets using VGG19.

5 Classification report

The authors evaluated the precision and recall scores for each class and compared the performance of VGG16, ResNet50, and SE-ResNet50 on the test dataset [41]. The classification report measures the model's performance on the test data by estimating the model's precision, recall, F1 score, and accuracy. Precision is the fraction of correctly predicted positive images among all images the model predicted positive. It is mathematically defined by $TP / (true\ positives + false\ positives)$. Recall is the fraction of correctly predicted positive

Table 3 Accuracy and loss metrics

	Validation accuracy	Validation loss
A	0.925	0.2181
B	0.8	0.4981
C	0.55	0.9077
D	0.775	0.775

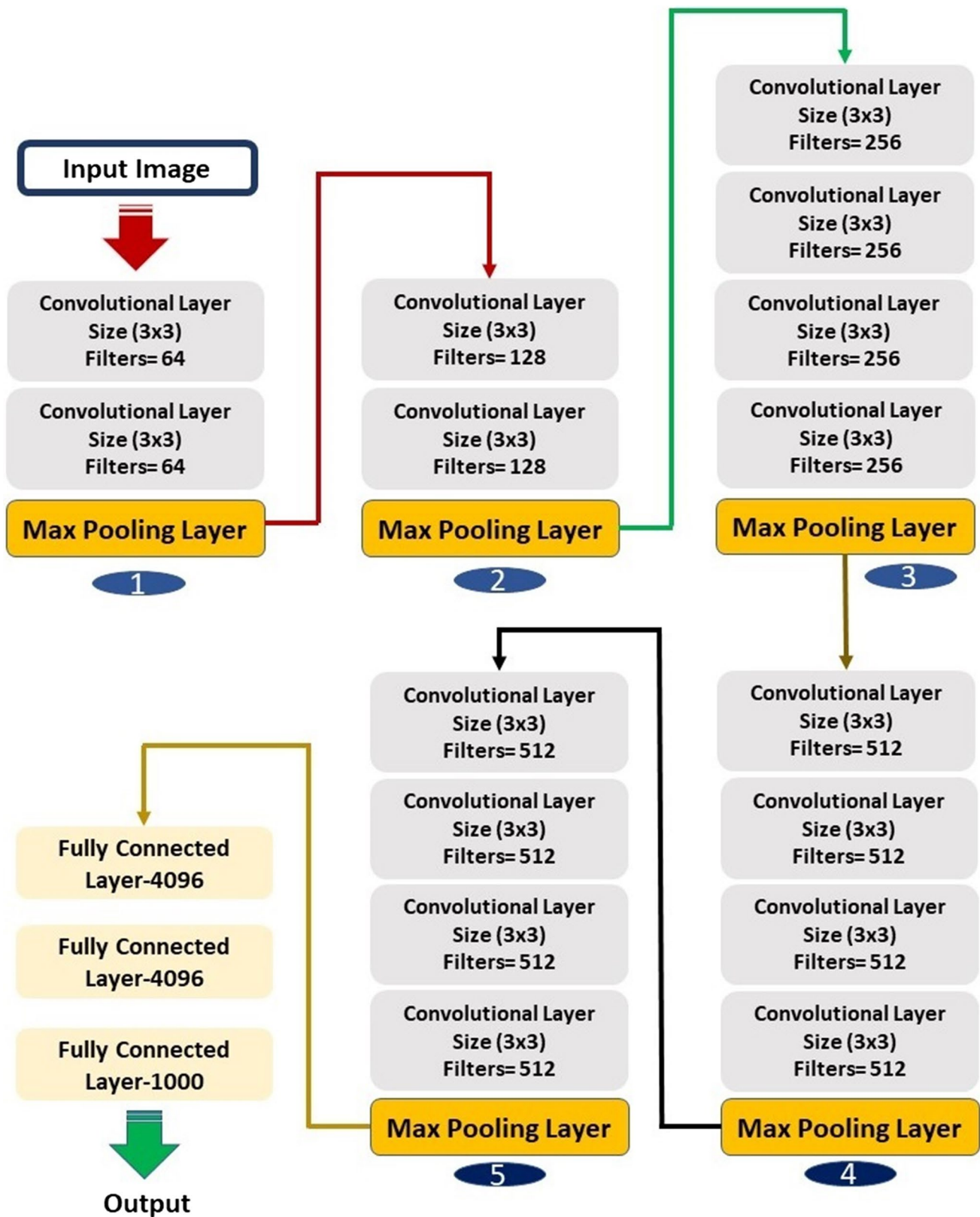
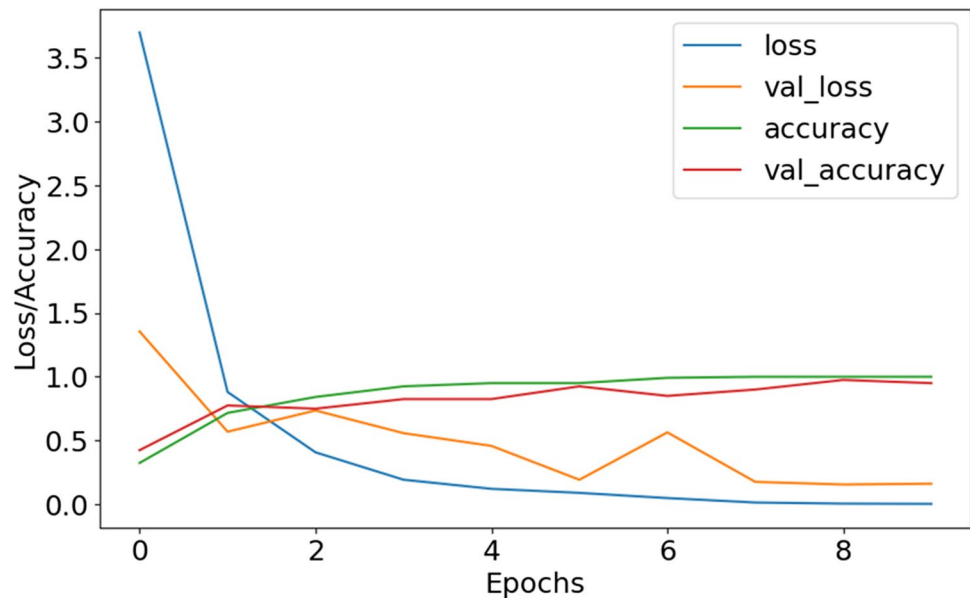


Fig. 7 The VGG19 architecture

Fig. 8 The loss and accuracy plots using VGG19



images among images that are positive and mathematically is defined by true positives/(true positives + false negatives). The recall and precision values are from 0 (lowest value) to 1 (highest). The F1 score is the weighted harmonic mean of the precision and recall [42]. Support is the number of test samples in each class. Accuracy computes the sum of correct classifications and divides it by the total number of classifications. In our model, the first, second, third, and fourth rows indicate the precision, recall, and F1 scores for the first, second, third, and fourth classes, respectively. The results of the model's prediction for our test images consisting of 26 images are provided in Fig. 9. The predictions by our proposed CNN model with two convolutional layers are comparable with the pre-trained VGG19 model. The results in Fig. 9 show that our proposed model for class B and C hardness has produced the highest precision and recall score of 1, the same as the precision and recall scores (1) for VGG19. Also, the recall score for class A hardness in our model and VGG19 is 1. Furthermore, our model and VGG19 have produced the same precision score (1) for class D. VGG19 performed slightly better than our model for precision in class A and recall in class D. The validation accuracy for our model and VGG19 was 0.925 and 0.950, respectively. In VGG19, the precision score for class A is 1, which is higher than our proposed model (0.86). For class D hardness, the recall score for VGG19 is 1, which is also higher than our proposed model (0.86). Our proposed model has two advantages. First, our model's training time using the CPU was approximately nine times faster than VGG19. VGG19 architecture is more complex than our convolutional neural network with two layers. Second, our model can be deployed more efficiently via Flask and requires less computational power. We had a small dataset and used 160 images

to train our model. Our model performance can be improved with more training images.

Tables 4, 5, and 6 show a sample of prediction results for the VGG19 and our model. According to these results, the VGG19 model predicted 7 images correctly. Our model predicted 5 images correctly and misclassified two images. The model's performance can be increased, and the misclassification of images can be minimized if training examples are increased [43]. In our proposed CNN model, the reason for the misclassification of images is our small training samples (160 images). Indeed, with more images, we can minimize overfitting and decrease the number of misclassified images. It should be noted that if we are predicting a few images, the expert prediction can be as good as the CNN prediction or better. However, in real-world situations, we often deal with many images. Most likely, human experts will not be able to predict the correct class label of micro-hardness images as fast and as accurately as CNN. Therefore, many CNN models can outperform humans with sufficient training examples. For example, in 2021, researchers used a large dataset with 17,302 images of melanoma (dangerous skin cancer) and nevus. They compared the diagnosis (image classification) of their optimized deep-CNN model with 157 dermatologists from 12 university hospitals in Germany. The experimental results demonstrated that their deep-CNN model outperformed 157 dermatologists [44].

6 Conclusions

In our research, we studied a prediction method of material micro-hardness after threading by the rolling process. We used convolutional neural networks (CNN) and VGG19 as

Fig. 9 Confusion matrix and classification report: A: proposed model, and B: VGG19

Confusion Matrix				
	[[6 0 0 0]			
	[0 7 0 0]			
	[0 0 6 0]			
	[1 0 0 6]]			
Classification Report				
	precision	recall	f1-score	support
A Hardness	0.86	1.00	0.92	6
B Hardness	1.00	1.00	1.00	7
C Hardness	1.00	1.00	1.00	6
D Hardness	1.00	0.86	0.92	7
accuracy			0.96	26
macro avg	0.96	0.96	0.96	26
weighted avg	0.97	0.96	0.96	26

A. Proposed Model

Confusion Matrix				
	[[6 0 0 0]			
	[0 7 0 0]			
	[0 0 6 0]			
	[0 0 0 7]]			
Classification Report				
	precision	recall	f1-score	support
A Hardness	1.00	1.00	1.00	6
B Hardness	1.00	1.00	1.00	7
C Hardness	1.00	1.00	1.00	6
D Hardness	1.00	1.00	1.00	7
accuracy			1.00	26
macro avg	1.00	1.00	1.00	26
weighted avg	1.00	1.00	1.00	26

B. VGG19

transfer learning (TL). We predicted four classes of micro-hardness of the St37 rolled threads. The conclusions of our work are provided below:

1. Our study presents an automated approach using CNN and VGG19 for micro-hardness evaluation when it is necessary to detect the micro-hardness of new material with no historical information.
2. We used two convolutional layers in our final proposed CNN model. The number of filters = 16 (first convolutional layer)—64 (second convolutional layer), and the dropout rate (0.05) in the fully connected layer were the

Table 4 The correct prediction labels by VGG19 model








	Predicted Label: 0 Actual Label: 0	Predicted Label: 1 Actual Label: 1	Predicted Label: 2 Actual Label: 2	Predicted Label: 3 Actual Label: 3	Predicted Label: 0 Actual Label: 0
Sample Image					
	Predicted Label: 2 Actual Label: 2	Predicted Label: 3 Actual Label: 3			
					

Table 5 The correct prediction labels by our proposed CNN model


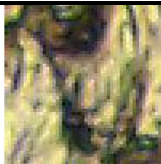



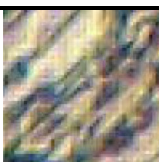
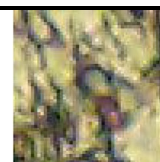
	Predicted Label: 0 Actual Label: 0	Predicted Label: 1 Actual Label: 1	Predicted Label: 2 Actual Label: 2	Predicted Label: 3 Actual Label: 3	Predicted Label: 0 Actual Label: 0
Sample Image					

Table 6 The incorrect prediction labels by our proposed model

	Predicted Label: 3 Actual Label: 0	Predicted Label: 2 Actual Label: 1
Sample Image		

best hyperparameters. The optimum width shift parameter was 0.05. These values were obtained through computational trials.

- The prediction results of the micro-hardness test images by our proposed CNN model and pre-trained VGG19 model are comparable. Our proposed model has produced the same precision and recall scores as VGG19 for class B and class C hardness. However, VGG19 performed slightly better than our model for precision in class A and recall in class D. More training examples can improve the performance of our proposed CNN model and minimize the fluctuations during our training time.
- Our proposed CNN and VGG19 both have direct applications in advanced manufacturing (AM) and can auto-

matically predict the micro-hardness in thread rolling of St37.

- The training time of our proposed model using the CPU (central processing unit) was approximately nine times faster than the VGG19 model. Our proposed model requires less memory and computational power and can be deployed more efficiently than the VGG19 model.

7 CNN performance and future work

In our future research, we will collect more images. We will employ EfficientNetV1, the current state-of-the-art for image recognition tasks. EfficientNetV1 was developed using inverted residual blocks of MobileNetV2 and Neural Architecture Search (MNAS). EfficientNetV1 includes the Swish Activation Function and SE (Squeeze and Excitation) blocks. We will also use Residual Neural Network (ResNet), a novel convolutional neural network architecture with skip connections and batch normalization. In standard CNN architecture, training and validation errors stop improving after a specific depth, and adding more layers will not improve the model’s performance. However, instead of stacking layers on top of each other to accomplish the desired accuracy, the network will fit a residual mapping in ResNet. We will improve our CNN model with

more images and compare its performance with Efficient-NetV1 and ResNet. We also intend to deploy our model with Flask and build a CNN web application similar to the AI web app that the second co-author of this paper has created with his colleagues for detecting COVID-19 using X-ray images and classifying galaxy images [45, 46]. The second co-author of this paper has already built a CNN web application for our proposed model with his research colleague [47]. We will improve their CNN application with more images in our future work. We have provided three videos in the supplementary section of our paper. These videos show the application of convolutional neural networks when deployed as a web application.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s00170-022-10355-4>.

Acknowledgements I have always immensely benefited from Professor Sergey Kravchenko's academic support at Northeastern University, Department of Physics in the United States of America, and I would like to express my sincere appreciation and most profound gratitude for his academic support and encouragement in particular in the application of Artificial Intelligence in Physics and Engineering. Affiliated Research Professor Mohammad Khoshnevisan, Physics Department, College of Science, Northeastern University, Boston, MA 02115, United States of America.

Author contribution All authors contributed to this work.

Data availability Not applicable.

Code availability Not applicable.

Declarations

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

Competing interests The authors declare no competing interests.

References

- Semiati SL (2005) ASM handbook, volume 14A: metalworking: bulk forming. ASM International
- Groover MP (2007) Fundamentals of modern manufacturing: materials processes, and systems. John Wiley & Sons
- Domblesky JP, Feng F (2002) Two-dimensional and three-dimensional finite element models of external thread rolling. *Proc Inst Mech Eng Part B J Eng Manuf* 216:507–517
- Domblesky JP, Feng F (2002) A parametric study of process parameters in external thread rolling. *J Mater Process Technol* 121:341–349
- Yamanaka S, Amiya K, Saotome Y, Inoue A (2011) Plastic working of metallic glass bolts by cold thread rolling. *Mater Trans* 52:243–249
- Yamanaka S, Amiya K, Saotome Y (2014) Effects of residual stress on elastic plastic behavior of metallic glass bolts formed by cold thread rolling. *J Mater Process Technol* 214:2593–2599
- Sağlam H, Kuş R (2011) Performance of internal thread rolling head and the mechanical properties of rolled thread. In: 6th International Advanced Technologies Symposium (IATS'11). 16–18
- Khostikoev MZ, Mnatsakanyan VU, Temnikov VA, Maung WP (2015) Quality control of rolled threads. *Russ Eng Res* 35:143–146
- Babei YI, Dutsyak ZG (1967) Effect of thread cutting techniques on the fatigue and corrosion-fatigue strength of steel. *Sov Mater Sci a transl Fiz mekhanika Mater Sci Ukr SSR* 3:277–283
- Brunton SL, Hemati MS, Taira K (2020) Special issue on machine learning and data-driven methods in fluid dynamics. *Theor. Comput. Fluid Dyn.* 1–5
- Lu NH, Huang H-C, Wu S-J, Hwang R-C (2021) Estimation of screw's physical properties using neural network. *Sensors Mater* 33:1859–1867
- Larkiola J, Myllykoski P, Korhonen AS, Cser L (1998) The role of neural networks in the optimisation of rolling processes. *J Mater Process Technol* 80:16–23
- Jenab A, Sarraf IS, Green DE et al (2016) The use of genetic algorithm and neural network to predict rate-dependent tensile flow behaviour of AA5182-O sheets. *Mater Des* 94:262–273
- Yao X (1996) Applications of artificial intelligence for quality control at hot strip mills, Doctor of Philosophy thesis, Department of Mechanical Engineering, University of Wollongong
- Dobrzański LA, Maniara R, Sokolowski JH, Krupiński M (2007) Modeling of mechanical properties of Al-Si-Cu cast alloys using the neural network. *J Achiev Mater Manuf Eng* 20:347–350
- El-Rehim A, Alaa F, Zahran HY et al (2020) Simulation and prediction of the Vickers hardness of AZ91 magnesium alloy using artificial neural network model. *Crystals* 10:290
- Abd El-Rehim AF, Habashy DM, Zahran HY, Soliman HN (2021) Mathematical modelling of Vickers hardness of Sn-9Zn-Cu solder alloys using an artificial neural network. *Met Mater Int* 1–13
- Magalhães FC, Ventura CEH, Abrão AM et al (2019) Prediction of surface residual stress and hardness induced by ball burnishing through neural networks. *Int J Manuf Res* 14:295–310
- Mahmoodi M, Naderi A (2016) Applicability of artificial neural network and nonlinear regression to predict mechanical properties of equal channel angular rolled Al5083 sheets. *Lat Am J Solids Struct* 13:1515–1525
- Maleki E, Unal O (2021) Shot peening process effects on metallurgical and mechanical properties of 316 L steel via: experimental and neural network modeling. *Met Mater Int* 27:262–276
- Sembiring J, Amanov A, Pyun YS (2020) Artificial neural network-based prediction model of residual stress and hardness of nickel-based alloys for UNSM parameters optimization. *Mater Today Commun* 25:101391
- Maleki E, Unal O (2021) Optimization of shot peening effective parameters on surface hardness improvement. *Met Mater Int* 27:3173–3185
- Gaudig W, Mellert R, Weber U, Schmauder S (2003) Self-consistent one-particle 3D unit cell model for simulation of the effect of graphite aspect ratio on Young's modulus of cast-iron. *Comput Mater Sci* 28:654–662
- Iacoviello F, Iacoviello D, Di Cocco V et al (2017) Classification of ductile cast iron specimens based on image analysis and support vector machine. *Procedia Struct Integr* 3:283–290
- Xu Z-W, Liu X-M, Zhang K (2019) Mechanical properties prediction for hot rolled alloy steel using convolutional neural network. *IEEE Access* 7:47068–47078

26. Li Z, Liu F, Yang W, et al (2021) A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Trans neural networks Learn Syst*
27. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. *Commun ACM* 60:84–90
28. Scherer D, Müller A, Behnke S (2010) Evaluation of pooling operations in convolutional architectures for object recognition. In: *International conference on artificial neural networks*. Springer. 92–101
29. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: *European conference on computer vision*. Springer. 818–833
30. Yamashita R, Nishio M, Do RKG, Togashi K (2018) Convolutional neural networks: an overview and application in radiology. *Insights Imaging* 9:611–629
31. <https://www.python.org/>
32. <https://www.tensorflow.org/>
33. Blanchard R (2020) Deep learning for computer vision with SAS: an introduction. SAS Institute
34. ABOUT NYILAI. <https://nyailabandinstitute.com/about-nyilai/>
35. Chollet F (2015) keras, GitHub. <https://github.com/fchollet/keras>
36. ImageDataGenerator. https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator
37. Milosevic N, Corchero M, Gad AF, Michelucci U (2020) Introduction to convolutional neural networks: with image classification using PyTorch. Springer
38. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. *arXiv Prepr arXiv14091556*
39. <https://image-net.org/>. <https://image-net.org/>
40. Sewak M, Karim MR, Pujari P (2018) Practical convolutional neural networks: implement advanced deep learning models using Python. Packt Publishing Ltd
41. Theckedath D, Sedamkar RR (2020) Detecting affect states using VGG16, ResNet50 and SE-ResNet50 networks. *SN Comput Sci* 1:1–7
42. Grigorev A (2017) Mastering Java for data science. Packt Publishing Ltd
43. Shrivastava VK, Pradhan MK, Minz S, Thakur MP (2019) Rice plant disease classification using transfer learning of deep convolution neural network. *Int Arch Photogramm Remote Sens Spat Inf Sci* 3:631–635
44. Pham T-C, Luong C-M, Hoang V-D, Doucet A (2021) AI outperformed every dermatologist in dermoscopic melanoma diagnosis, using an optimized deep-CNN architecture with custom mini-batch logic and loss function. *Sci Rep* 11:1–13
45. Khoshnevisan M, Afshari BM, Dehnad K <https://covid-19-nyilai.herokuapp.com/>
46. Kravchenko S, Khoshnevisan M, Afshari BM <https://galaxy-nyilai.herokuapp.com/>
47. Khoshnevisan M, Afshari BM <https://hardness.herokuapp.com/>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.