Deep Transfer Learning Enabled Estimation of Health State of Cutting Tools

M. Marei, S. El Zaataria, and W. D. Li

1 Introduction

In manufacturing industries, unplanned downtime is known to negatively impact profitability, and will be a barrier to implementing lean and zero-defect manufacturing. Also, operational safety could be compromised through unexpected failures, particularly when human operators are involved [1]. To tackle the challenge, predictive maintenance based on Prognostics and Health Management (PHM) has been developed to predict the failure points of working components (such as bearings and cutting tools) [2–6]. Based on that, a component in a manufacturing system can be replaced just before it fails. Thus, component lifetime can be maximized, system downtime can be minimized, and therefore optimal productivity and production quality can be achieved. In Computerized Numerical Control (CNC) machining processes, cutting tool wear leads to various manufacturing problems, ranging from stoppage downtime for redressing and tool replacement, to scraps and reworks of machined components due to degradation in surface quality [7]. Therefore, accurate prediction of the Remaining Useful Life (RUL) for a cutting tool is essential to mitigate such failures [8].

In the aspect, physics-based approaches on empirical models have been developed, such as the Taylor, Extended Taylor, Colding, and Coromant Turning model (a more detailed review can be found in [9]). However, these approaches are sensitive to variations in machining parameters (e.g., cutting speed, feed rate, cutting depth, cutting tool properties such as the number of teeth), which vary depending

M. Marei · S. E. Zaataria · W. D. Li (✉)
Faculty of Engineering, Environment and Computing, Coventry University, Coventry, UK
e-mail: weidong.li@coventry.ac.uk

W. D. Li
School of Logistics Engineering, Wuhan University of Technology, Wuhan, China

W. Li et al. (eds.), *Data Driven Smart Manufacturing Technologies and Applications*, Springer Series in Advanced Manufacturing,
https://doi.org/10.1007/978-3-030-66849-5_7

on component materials and machining set-up. Moreover, profound expert knowledge of machining processes is also expected to conduct effective and accurate RUL prediction. In contrast to physics-based approaches, data-driven approaches have been developed to leverage historical and real-time data to support decision-making. Deep learning algorithms (e.g., Convolutional Neural Networks (CNNs)) have been explored to facilitate data-driven approaches (a related review can refer to [10]). For instance, to attain a wide scope of image features from a variety of applications, CNNs models, such as ImageNet [11] (and the subsequent ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [12]), CIFAR-10 and CIFAR-100 [13], can be trained on millions of images of natural and synthetic objects. These approaches excel at extracting discriminative features and learning hidden relationships between problem parameters (i.e., feature learning), opposed to feature engineering approaches where human experts specify the features to learn [14]. However, the accuracy and reliability of deep learning enabled data-driven approaches may be reduced significantly when data are scarce or insufficiently descriptive of the problem. To address the issue, in recent years, transfer learning enabled approaches have been developed to improve pre-trained deep learning models to perform prediction on new problems with limited available data [15, 16]. The transfer learning strategy is to retain and reuse domain-invariant features learn from a task in one domain (called the source domain), to enhance the performance of another task in its corresponding domain (called the target domain). On the other hand, though transfer learning has shown its great potentials in different applications, there are still limited research works reported on manufacturing applications, especially for estimating the RUL of cutting tools.

In this chapter, it is aimed to develop a transfer learning enabled CNNs approach for RUL estimation of cutting tools and further establish PHM based on analyzing the images of healthy and worn tools. The problem of limited data available during machining processes is tackled by integrating transfer learning into CNNs. The main characteristics of this research are summarized as follows:

1. Developing a transfer learning enabled CNNs approach with gradient descent optimization algorithms based on normalized Maximum Mean Discrepancy (MMD) and Mean Square Error (MSE) for algorithm training with learnable weights;
2. Benchmarking the performance of the approach through evaluating several CNNs architectures and transfer learning for tool RUL prediction and PHM;
3. Providing recommendations for training techniques and data augmentation based on benchmarking and analysis results.

2 Related Works

A brief review on prior works in deep learning for PHM applications is presented here with a focus on implementations in the manufacturing domain. Relevant literature aligning with CNNs and transfer learning are also highlighted.

2.1 Deep Learning for PHM

In recent years, there have been increasing reviews on investigating the development of deep learning approaches for PHM applications [10]. Most deep learning approaches for PHM exploit an inflow of continuous real-time data, such as vibration signals [5], acoustic emission sensor data [8], force measurements [7, 17], temperature [8], power/current, etc. Alternatively, other approaches were developed and tested with an existing RUL dataset at the validation stage (some examples of these datasets are mentioned in [9]). The C-MAPSS (aero-engine health degradation simulation) tool [17] was used to create datasets extensively studied in prior works on PHM, with varying research perspectives [18]. In addition, the PHM society holds an annual challenge for PHM techniques based on data that it provides, in which the 2010 dataset focuses on high-speed CNC machining [19]. While numerous architectures of deep learning were implemented for PHM, several primary models can be summarized into the following types:

- CNNs and their variants: these approaches use a series of shallow or deep convolutional filters that extract spatial features from images or image-like data. These approaches are particularly efficient at learning meaningful representations of the data from individual and grouped clusters of filters across several depth channels [20]. Deeper CNNs layers are typically capable of extracting distinct visual features like parts of the general shape of the image, whereas shallower layers typically extract image intensity or color variation across a limited image window [21]. While predominantly used for failure classification in PHM [22], several researchers successfully used CNNs for regression-related feature extraction for RUL prediction [23]. A few approaches were developed to perform both classification and regression [23]. However, few approaches utilized images as input for predicting the health state of cutting tools.
- Recurrent Neural Networks (RNNs and their variants: these models learn from data sequences with explicit time dependencies between input samples (i.e., sequence prediction problems), due to series of gates within the architecture which retain the outputs from previous inputs. The output from one neuron is fed forward into the adjacent neuron, so that the hidden representation or the output of the neuron is influenced by past inputs [11]. Long Short-Term Memory (LSTM) networks [24] are similar to RNNs but can retain memory over a longer time horizon due to a more complex gate structure that preserves long-term dependencies. A combination of gates with different activation functions determine what portion of the cell state to retain or transform for input into the subsequent layer. LSTM and their variants have achieved widespread success at time series-based prediction problems, therefore are especially popular for PHM applications [21].
- Auto-Encoders (AEs) and their variants: AEs are feedforward neural networks comprising an encoder and a decoder. The encoder is trained to find the hidden representation of the input data via a nonlinear mapping of the weights and biases. The decoder attempts to find an output to the inverse function of the hidden representation to the data, i.e., a nonlinear mapping between the encoder function

and the hidden representation. To avoid human-assisted feature identification for Tool Condition Monitoring (TCM), Shi et al. used a novel Feature Multi-Space Sparse Auto-Encoder (FMSSAE) to classify four different wear states from TCM data in the time domain (TD), frequency domain (FD) and Wavelet Domain (WD) [25]. Their methodology reported to achieve 96% accuracy in classification.

- Hybrid approaches: which combine and adapt several architecture designs to conduct feature extraction and time-based prediction, e.g., Zhao et al. used a Convolutional Bi-directional Long Short-Term Memory (CBLSTM) to extract local and informative sequence features, followed by a bi-directional LSTM to identify long-term time dependencies, leading to linear regression for target value prediction [6]. The approach was demonstrated to predict tool wear based on raw sensory data.

2.2 Transfer Learning Enabled Deep Learning

In the reviewed literature, an ongoing theme is to develop approaches that are trained or validated on publicly available datasets or those collected in individual experiments. Consequently, the replicability of these studies that leverage closed-source data may be called into question [11]. An additional limitation to the data being generated for PHM studies relates to the classic imbalance problem (whereby healthy data samples are much more prominent than faulty data samples) [11]. Meanwhile, the accuracy and reliability of the approaches are significantly hindered by insufficient manufacturing data: a key limitation for most deep learning approaches is their reliance on large quantities of data, typically in the order of ~1000 samples per class (for classification type problems).

Transfer learning has presented its potential to address the above problems [15, 16, 26]. With transfer learning, knowledge acquired from one domain might be retained and reused in a new domain. In general, the methodologies of transfer learning can be classified into the following four categories according to what and how the knowledge is transferred: (1) Instance based transfer learning—the labelled datasets from the source domain are reused in the target domain; (2) Feature based transfer learning—the features in the source domain are reused in the target domain if the features of the source domain match those in the target domain; (3) Parameter based transfer learning—the setting parameters of a machine learning algorithm in the source domain are re-used in the target domain; (4) Relational knowledge-based transfer learning—the relationship between the data from the source domain and the target domain is established, which is the base for knowledge transfer between the two domains.

In essence, transfer learning models repurpose the weights of deep learning approaches learned in classification tasks, corresponding to features in a similar or different domain (e.g., general-purpose image classification challenge datasets like ILSVRC [8] and Places [27]), to perform predictions for a new task. Such models typically achieved remarkable success, leading to a new research direction in exploring

this generalizability by evaluating the classification or regression performance in new tasks (i.e., domain adaptation [27]). In particular, for transfer learning, many approaches work well under a pre-requisite condition: the cross-domain datasets are drawn under the same feature distribution. When the feature distribution changes, most deep learning based approaches need to be re-trained from scratch.

In recent years, various research works have been conducted to integrate transfer learning into deep learning algorithms, i.e., deep transfer learning. For instance, Lu et al. developed a deep transfer learning algorithm based on deep neural network and feature based transfer learning for domain adaptation [28]. In the research, the distribution difference of the features between the datasets from the source domain and target domain was evaluated based on the Maximum Mean Discrepancy (MMD) in a Reproducing Kernel Hilbert Space (RKHS). Weight regularization was carried out by an optimization algorithm to minimize the difference of the MMD for the two domains in implementing knowledge transfer. Xiao et al. designed another deep transfer learning algorithm for motor fault diagnostics [29]. In the algorithm, a feature based transfer learning approach was developed to facilitate knowledge learnt from labelled data under invariant working conditions to the unlabeled data under constantly changing conditions. MMD was incorporated into the training process to impose constraints on the parameters of deep learning to minimize the distribution mismatch of features between two domains. Wen et al. proposed a novel deep transfer learning algorithm for fault diagnosis. A cost function of weighted fault classification loss and MMD was used to fine-tune the model weights [30].

2.3 Images in PHM Applications

Traditionally, images are used within PHM for visual inspection when assessing the condition of the damaged component or machine. However, similar image data could be used as a viable tool to predict (or localize) faults within a machine component, particularly if the frequency of such image measurements is sufficiently large. Despite their recent success in several time–frequency applications for PHM, applications of CNNs to process image data in a PHM context are still not frequent. Most approaches use either 2D representations of time- or frequency-domain data (e.g., engine sensor data in [31], vibration signals in bearings in [5]). Comparatively few examples exist where the failure mode of a machine was classified by a pre-trained CNNs model based on visual data. In particular, Janssens et al. utilized pre-trained versions of the VGG-16 CNNs model [32, 33]. The model was re-trained on thermal video images to classify the failure type, firstly by the image intensity, and secondly based on the degree of imbalance of the rotating machine. Their approach combines a CNNs trained to extract spatial features from thermal images, and another trained to extract temporal information from differenced images to represent the motion due to imbalances. In their first case study, 12 different health conditions were successfully classified with 91.67% accuracy using a deep learning approach, versus conventional approach that classified the 12 health conditions with 55.00% accuracy. Additionally,

they implemented the same methodology to perform binary classification on another system, reporting an accuracy of 86.67% with the feature learning vs 80% with the conventional approach. The work provides a promising first step towards intelligent PHM for real-time fault prediction.

The above research predominantly relies on a large number of images to perform RUL prediction or failure classification for diagnosis. While demonstrating widespread success on their individual dataset, these methodologies are ineffective for the cases of limited images available. The methodology in this research leverages transfer learning enabled CNNs for PHM applications, through which the extensibility to other problems within PHM are also demonstrated.

3 Methodology

An overview of the developed methodology is described here, first by introducing the overall workflow and then by detailing its constituent components.

3.1 Problem Definition and Overall Methodology

The overall methodology is shown in Fig. 1. The objective of this research is to predict the RUL of a cutting tool, given the image of the tool as an input and the corresponding normalized tool wear measurement as a prediction target. In other words, the objective is to determine:

$$\widetilde{\boldsymbol{y}} = \boldsymbol{w}^T \boldsymbol{x} + \boldsymbol{b} \tag{1}$$

where $\widetilde{\boldsymbol{y}}$ is the matrix for the predicted regression output, $\boldsymbol{w}$ and $\boldsymbol{b}$ are the trainable weights and biases of a CNNs model respectively, and $\boldsymbol{x}$ is the input matrix for the images.

A pre-trained CNNs model is deployed for predicting the RUL of a cutting tool. The weights of the CNNs model are then adjusted through adaptively learning the images of cutting tools based on a transfer learning process. Meanwhile, to facilitate the CNNs model for the prediction, the end layers, which consist of the loss classifier and the classification output layer, are replaced with a designed regression layer stack as a regression problem.

More details of the developed approach are in the following steps: (1) forward computation of the CNNs is carried out by using the datasets of both the source domain (datasets for pre-training the CNNs) and the target domain (the images for tool heath state) as input for tool RUL prediction; (2) back propagation computation in the CNNs is performed to minimize the feature distribution discrepancy for the two domains and the prediction errors of the tool RUL. Gradient descent optimization

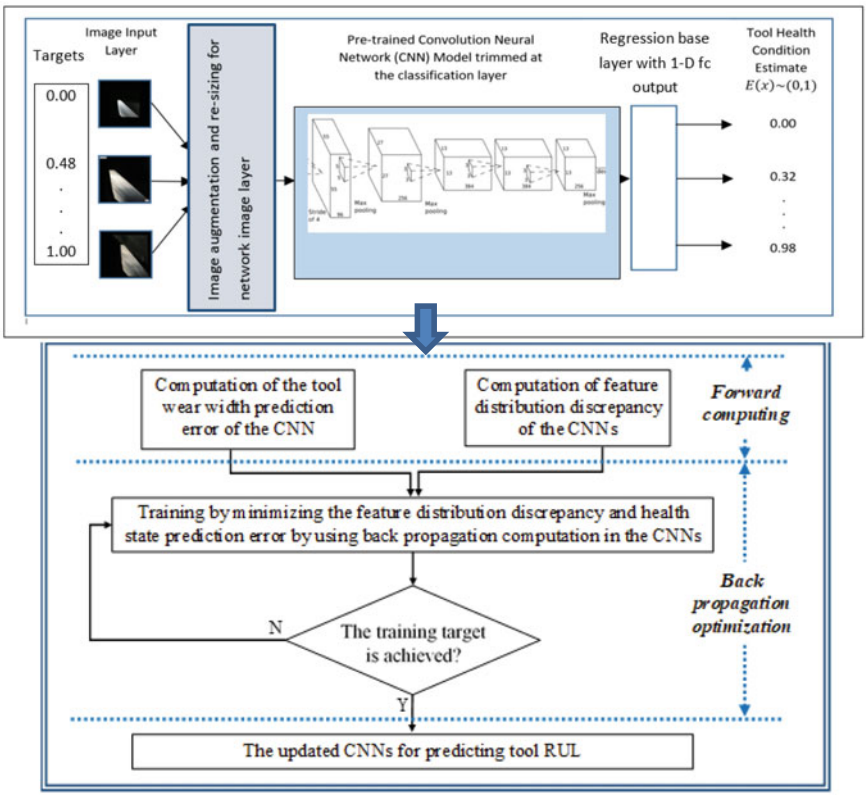

Fig. 1 The detailed procedure of the approach for tool RUL prediction

algorithms are evaluated and used for the minimization process, and the updated CNNs is deployed for tool RUL prediction. illustrates the above steps. More details on constituent components are given in the following sub-sections.

3.2 The Input Data for Transfer Learning

The dataset used for transfer learning comprises microscope images of the carbide cutting tool flank, collected at set intervals throughout the experiment, along with recorded flank tool wear width v in mm. The cutting experiments were conducted under varying conditions of cutting speed v_c, feed rate f_d, and cutting width a_e. In total, 25 experiments were used to vary these parameters following a Taguchi Design of Experiments (DoE) approach, with 3 factors (v_c, f_d, and a_e), 5 levels per factor, and 1 response variable (i.e., the measured flank wear width, v). The purpose is to analyze the effects of varying these parameters on the longevity of the tools.

In addition to the cutting tool images, the flank wear width was measured for each tool with a wear threshold of $v_b = 0.4$ mm indicating the tool has failed. Several image views of the cutting tool were recorded, but the analysis was focused on one particular variant of image views, at a magnification factor of 100 and a full frontal view of the tool.

Figure 2 illustrates the tool life trend of the 25 cutting tools within this experiment. In Fig. 3, the images of the post-processed cutting tools were captured for Experiment #1. In addition, some tool wear measurements (in particular those corresponding to early failure events, e.g., Experiment #15) had a final values v < 0.4 mm; others had much larger values (e.g., Experiment #21) with v ~ = 1 mm. In total, 327 images of cutting tools with appropriate tool wear width were used, split into 195 images (59.94%) for training and 132 (40.06%) for validation. To simplify benchmarking, the same pre-shuffled training and validation data were used. For original images, the function of image batch processing was implemented to perform a boundary cropping operation to remove the excess image backgrounds, yielding 800 × 800 pixel images. To avoid discarding additional data that could be relevant, the training and validation data were normalized between 0 (indicating a healthy tool) and 1 (for a fully worn tool).

3.3 CNNs Models and Regression Stack

For the approach developed in this paper, transfer learning allows the knowledge obtained from original tasks to be repurposed for different tasks. This means that the weights and biases of pre-trained CNNs models could be adjusted or fine-tuned with new training data. While the earlier feature pool layers of CNNs typically extract general image features that are considered safely transferable, specialized features are typically extracted in deeper layers. The degree of specialization often leads to some levels of pre-training bias, where the models retain features learned from the pre-training phase even despite being trained for extensive durations. It is intuitive to select models with a good performance in general classification to be further fine-tuned via transfer learning. This is because such a model would have been trained to accurately recognize features belonging to a multitude of different classes. Feature transferability is addressed using minimization optimization procedures for MMD and RUL prediction errors, described in Sect. 4 later on.

When selecting pre-trained CNNs models for transfer learning, another important consideration is the computational complexity of the models. While it is often observed that deeper models tend to outperform shallower ones at certain tasks, it is not always the case. The SqueezeNet architecture, for instance, was able to attain AlexNet-level accuracy on the ImageNet data challenge with nearly 50 times fewer parameters [34]. Therefore, comparing a variety of CNNs models quantitatively is useful to help evaluate their merits and appropriateness of this research. The CNNs models were chosen based on their classification performances in general-purpose

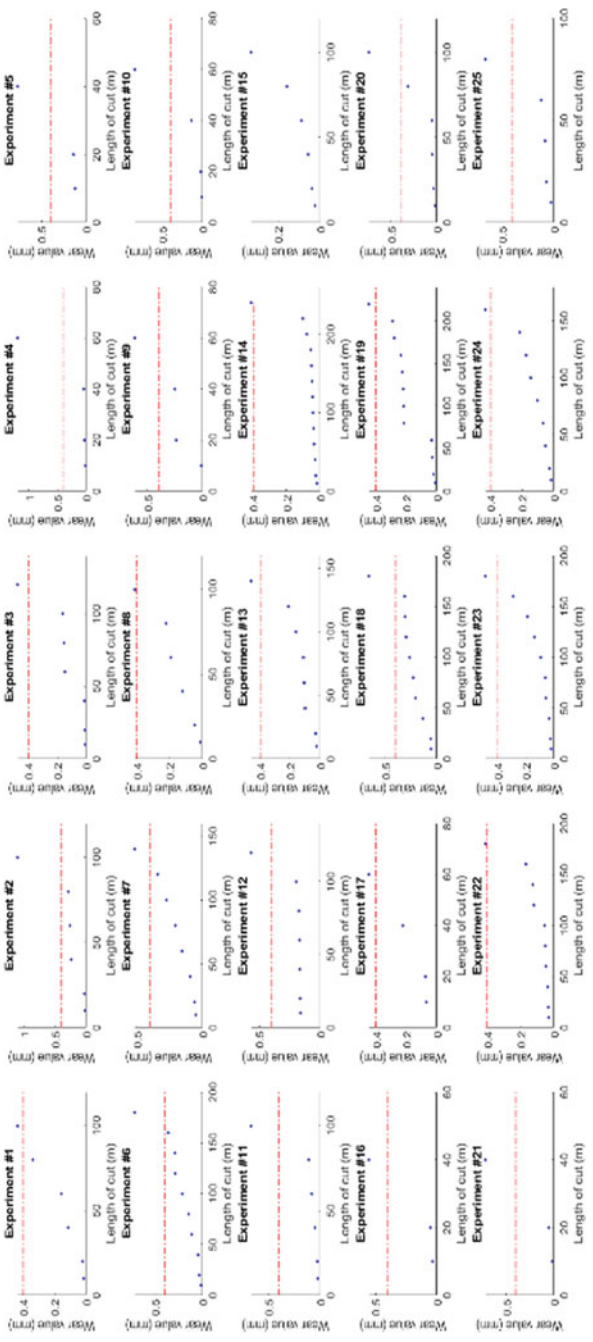

Fig. 2 Tool life trends of 25 cutting tool experiments

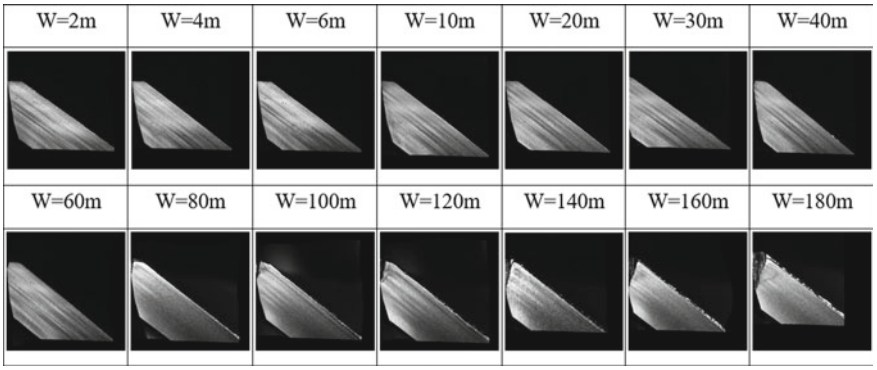

Fig. 3 Experiment #1: pre-processed tool wear images recorded at W m cutting intervals. The cutting interval in earlier measurements was kept small to capture the early wear trend

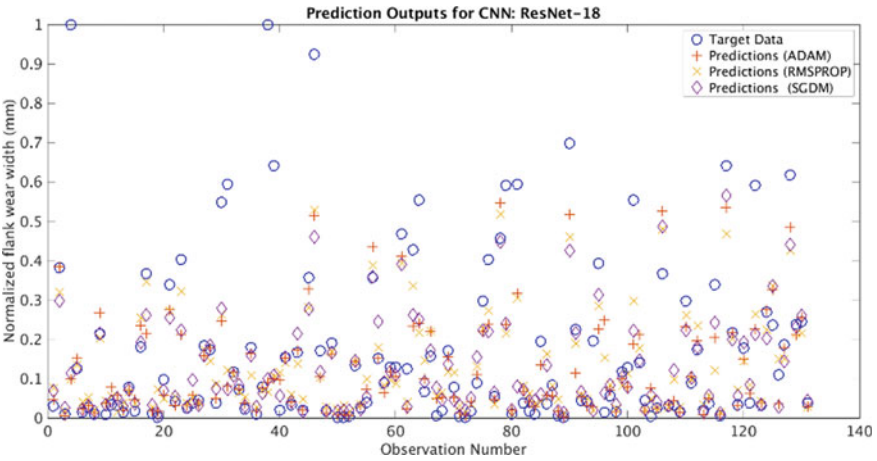

Fig. 4 ResNet-18 predictions versus targets using three optimizer variants

Table 1 Pre-trained CNNs models investigated in this study, with performance reported in terms of top-1 and top-5 percentage accuracy when trained on ILSVRC 2012

Network	Top-1 Accuracy (%)	Top-5 Accuracy (%)	Parameters (Millions)	Input Size
AlexNet [21]	63.3	84.6	61.0	227 × 227
ResNet-18	71.78	90.58	11.7	224 × 224
ResNet-50	77.15	93.29	25.6	224 × 224
ResNet-101	78.25	93.95	44.6	224 × 224
SqueezeNet	60.4	82.50	1.24	227 × 227
InceptionV3	78.95	94.49	23.9	299 × 299

classification tasks. Table 1 highlights the model size, input image size, and results from classification challenges for the CNNs models.

The regression stack is a collection of layers that progressively adapt the outputs of the pre-trained CNNs to make them more suitable for regression-based prediction. It comprises the following layers:

- A 4096-channel fully-connected layer, which function is to further down-sample the outputs of the previous pooling layer (which is a common design choice for CNNs models);
- A batch normalization layer, responsible for normalizing the inputs of the previous layer;
- Rectified Linear Unit (ReLU), which applies a non-linear transformation to the prior layer outputs;
- A 410 fully-connected layer, which down-samples the previous layer inputs;
- A sigmoid layer, which transforms the outputs of the previous layer to the range (0, 1) via the sigmoidal activation function;
- A regression output layer, which computes the loss of the prediction $\hat{\boldsymbol{y}}_i$.

As opposed to a classification layer that computes the probability of an image belonging to a given image class, the regression output layer computes the loss as the MSE (Mean Square Error) of the prediction $\hat{\boldsymbol{y}}_i$ given the target $\boldsymbol{y}_i$. MSE is defined as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i)^2} \tag{2}$$

where N is the node number of the layer.

3.4 Transfer Learning Process

The CNNs model with the regression stack is re-trained on the training dataset of 195 images through the procedure illustrated in Fig. 1. In order to transfer the knowledge from the source domain (pre-trained CNNs) to the target domain (trained CNNs for tool RUL prediction), the developed approach should be subject to the condition that features are in similar distributions between domains. To address feature distribution mismatch during transfer learning, an error minimization optimization strategy is applied through back propagation computing on the pre-trained CNNs. In the previous literature, MMD (Maximum Mean Discrepancy) was popularly used to measure the distance metric for probability distribution between two domains. That is, the datasets in the source domain and the target domain are represented as $D_S = \{X_S, P(x_s)\}$ and $D_T = \{X_T, P(x_T)\}$ respectively. Meanwhile, $X_S = \prod_{i=1}^{ns}\{x_s^i, y_s^i\}$ with ns samples, and $X_T = \prod_{i=1}^{n_t}\{x_T^i\}$ with n_t samples respectively. Their MMDs are defined below:

$$Mean_H(X_S) = \frac{1}{n_s}\sum_{i=1}^{n_s} H\left(x_s^i\right) \tag{3}$$

$$Mean_H(X_T) = \frac{1}{n_t}\sum_{j=1}^{n_t} H\left(x_T^j\right) \tag{4}$$

$$MMD_H(X_S, X_T) = \sup[Mean_H(X_S) - Mean_H(X_T)] \tag{5}$$

where sup(·) represents the supremum of the aggregate; $H(\cdot)$is a RKHS (Reproducing Kernel Hilbert Space).

In this research, the above concept of MMD is adopted for measuring the feature distribution difference of domain invariant features. To achieve similar distributions from two domains, $MMD_H(X_S, X_T)$ is considered as the optimization objective to regularize the weights of the CNNs. Meanwhile, during the re-weighting process on the CNNs, the prediction error should be minimized as well. Thus, the prediction error is considered as another optimization objective.

As discussed earlier, the total loss function $Loss$ can be calculated based on $MMD_H(X_S, X_T)$and MSE. Since $MMD_H(X_S, X_T)$ and MSE are in different value ranges, normalization is required. In this research, Nadir and Utopia points are utilized to normalize the above three objectives in an effective means [35]. The Utopia point ${z_i}^U$ provides the lower bound of No. i objective obtained by minimizing the objective as below:

$${z_i}^U = minf(i) \tag{6}$$

The Nadir point ${z_i}^N$ provides the upper bound of No. i objective by maximizing the objectives:

$${z_i}^N = \max_{1 \ll j < I} f(j) \tag{7}$$

where I is the total number of the objective functions.

According to Eqs. (5) and (6), the normalized $MMD_H(X_S, X_T)$and MSE can be calculated below:

$$NMMD_H = (MMD_{H1}(X_S, X_T) - {z_1}^u)/({z_1}^N - {z_1}^u) \tag{8}$$

$$Nmse = (MSE - {z_2}^u)/({z_2}^N - {z_2}^u) \tag{9}$$

where $NMMD_H$ and $Nmse$ are the normalized $MMD_H(X_S, X_T)$and MSE respectively.

The total loss function $Loss$ can be calculated based on the weighted sum of the two normalized objectives:

Table 2 Fine-tuning the transfer learning enabled CNNs using different optimizers

1. Imagesize = size of image input layer
2. Identify the last (tune-able) layers in the network
3. Set learning rate to 0 for other layers
4. For each optimizer
5. If optimizer is ADAM or RMSProp
6. Set initial learning rate to 4e-5
7. Else if optimizer is SGDM
8. Set initial learning rate to 2e-2
9. Pre-initialize augmenter which augments the input images by resizing and performing random transformations
10. Train the network on augmented images
11. End

$$Loss = w_1 \cdot NMMD_H + w_2 \cdot Nmse \tag{10}$$

where $w_1 - w_2$ are the weights of the two objectives, and $\sum_{i=1}^{2} w_i = 1$.

Based on the above process, three variants of training optimization algorithm were investigated and compared, including Stochastic Gradient Descent with Momentum (SGDM), Root Mean Square Propagation (RMSPROP) and Adaptive Moments (ADAM) [36]. SGDM has been a popular choice for training ANNs since its inception in 1999, and its subsequent resurgence when used in AlexNet. RMSProp is another popular algorithm for gradient descent training to eliminate the need for learning rate adjustment. ADAM combined the heuristics of both Momentum and RMSProp to achieve faster convergence.

The CNNs models were trained according to the procedure illustrated in Table 2, after being converted to a *layerGraph* (the MATLAB structure that retains the CNNs configuration). Firstly, a helper function searches the *layerGraph* of the CNNs model for a Softmax layer, its preceding layer (a fully-connected layer with 1000 outputs) and subsequent classification layer. The function returns the names and indices of these layers in the *layerGraph*, after which they are removed using a helper function, and replaced with the *baseLayers* configuration described in the regression stack.

The models were trained for 750 epochs in 12 iterations per epoch (9000 iterations total), with a mini-batch size of 16 images. The training function was set to shuffle the mini-batch every epoch. To speed up training, validation was done every 40 iterations. During training, image augmentation operations were implemented on each training mini-batch, to vary the input images by introducing some aspects of visual variation to the images (random translations, rotations, and scaling). This has the effect of inflating the dataset, allowing the CNNs model to consider more examples of the data than are available.

4 Experimental Results and Discussions

4.1 Experiment Data Collection

Table 3 details the benchmarking results for fine-tuning the 6 CNNs models by the transfer learning process, where the 3-run average of each model variant's output was recorded. To evaluate the quality of prediction, the models were assessed with the following performance criteria:

(1) Training time (in seconds).
(2) Mean Absolute Error (MAE), which is defined below:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i)^2 \tag{11}$$

(3) Mean Square Error (MSE), which has been defined in Formula (2).
(4) $acc_{10,20,30}$, the accuracies of all predictions are below 10%, 20% or 30% error thresholds from the targets. The threshold of the T percentage accuracy is:

$$acc_T = \frac{1}{N}\sum_{i=1}^{N} 1_T\left((\hat{\boldsymbol{y}}_i\right) \tag{12}$$

$$1_T\left(\bar{\boldsymbol{y}}_i\right) := \begin{cases} 1, (\hat{\boldsymbol{y}}_i \geq T \times \left|\max\left(\boldsymbol{y}_i\right)\right| \\ 0 otherwise \end{cases} \tag{13}$$

where the range of T is determined by $T \in [0.1, 0.2, 0.3]$.

The threshold of 10% accuracy indicates the percentage of prediction that falls within 10% of this error in either direction. In most cases, due to the proportion of healthy samples compared to faulty or worn tool images, the prediction errors are on the lower end of the range, i.e., ~10% below the target value. The performance measures of 20% and 30% accuracy are considered additional qualitative metrics indicating whether predictions can be accepted.

4.2 Result Analysis and Observations

In Table 3, comparing these results, ResNet-18 (trained with ADAM) can offer the best performance in terms of average prediction error and acc10, with a reasonable training time considering the number of iterations attempted. ResNet-50 and ResNet-101 variants are both longer to train and generally less accurate based on MAE and MSE. Meanwhile, despite being much deeper than other models, the InceptionV3

Table 3 Benchmarking results for fine-tuning the 6 CNNs models

Pre-trained Model	Training Details			Model Performance					
	Optimizer	Learning Rate	Batch Size	MAE	MSE	acc10 (%)	acc20 (%)	acc30 (%)	Training Time (s)
AlexNet [34]	adam	4e-5	16	0.0829	0.1684	81.68	**90.84**	91.60	2124.8
	sgdm	2e-2	16	0.0868	0.1723	79.39	90.08	91.60	1940.1
	rmsprop	4e-5	16	0.0903	0.1726	77.86	90.08	90.84	2027.6
ResNet-18 [31]	adam	4e-5	16	**0.0773**	0.1654	**83.97**	**90.84**	92.37	3358.4
	sgdm	2e-2	16	0.0820	0.1591	78.63	90.08	92.37	2649.0
	rmsprop	4e-5	16	0.0791	0.1594	80.15	90.08	92.37	2853.5
ResNet-50 [31]	adam	4e-5	16	0.0868	0.1764	80.92	86.26	90.84	14,790
	sgdm	2e-2	16	0.1124	0.1967	74.05	84.73	90.08	9184.2
	rmsprop	4e-5	16	0.1050	0.1954	76.34	83.97	90.08	12,689
ResNet-101 [31]	adam	4e-5	16	0.0833	0.1657	74.81	89.31	92.37	17,750
	sgdm	2e-2	16	0.0992	**0.1565**	74.05	89.31	**93.89**	15,122
	rmsprop	4e-5	16	0.0882	0.1740	77.86	86.26	90.84	16,654
SqueezeNet [37]	adam	4e-5	16	0.0891	0.1732	79.39	88.55	91.60	2151.8
	sgdm	2e-2	16	0.0868	0.1784	79.39	89.31	91.60	**1763.5**
	rmsprop	4e-5	16	0.0882	0.1710	77.68	88.55	91.60	1878.0
InceptionV3 [32]	adam	4e-5	16	0.0886	0.1784	79.39	85.50	91.60	20,334
	sgdm	2e-2	16	0.1040	0.1931	77.10	85.50	90.08	14,653
	rmsprop	4e-5	16	0.0916	0.1728	79.39	87.02	91.60	18,780

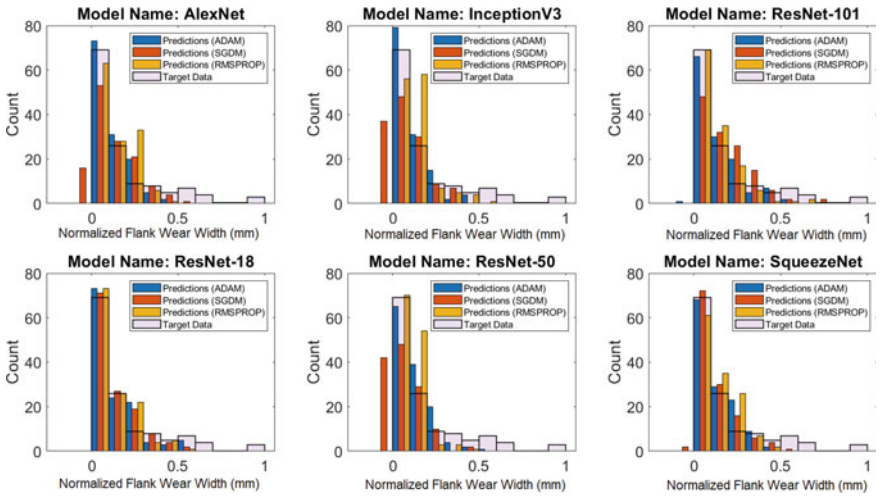

Fig. 5 Prediction histogram comparison between six benchmarked CNNs models

variants were amongst the worst performing models considering MAE, MSE and the accuracy thresholds. Furthermore, the increase in model depth from ResNet-18 to ResNet-101 has increased training time fivefold, without an improvement in performance. This emphasizes a key observation that increase of the model depth does not mean increase in prediction accuracy accordingly. Sample prediction outputs using the three optimizers chosen (ADAM, RMSPROP and SGDM) are illustrated in Fig. 4.

Figure 5 shows a histogram plot of the prediction outputs of the six CNNs. Some further observations can be made regarding the performance of these models:

- Overall best fit: It shows that ResNet-18 produced the closest prediction output distribution to the validation target data, across the three optimizer training variants.
- Overfitting: All models over-fit the results significantly in the "healthy" categories, with the performance of ResNet-18 (ADAM) being the best out of the compared model variants in terms of overfitting, where the less the model over-fits, the better its performance.
- Generalization performance: Comparatively, ResNet-50 produced the worst general fit results, indicated by its comparatively higher MAE and MSE as well as lower accuracy across all thresholds. This might indicate that the model has a tendency to over-fit the data more strongly than other models. In fact, the generalization performance of SqueezeNet, which is close to 20 times smaller in parameters, is markedly better consider the relative difference in model size.
- Anomalous predictions: With the exception of ResNet-18, all models trained with SGDM have a tendency to produce negative outputs, despite the sigmoid layer (whose function is to force its outputs to be between 0 and 1) being the last layer

prior to the regression output layer. This is a property of SGDM which enables it to generalize better than the other training algorithms. However, in doing so the SGDM variants predict results in the reverse direction of what is desired. This contrasts to tool wear width values, which must always be indicated by a positive value.

- Training duration: It is also worth mentioning that increasing the number of epochs to 9000 did not have a profound impact on the accuracy. Some initial trials with fewer iterations (i.e., 150 instead of 750) yielded similar results for most of the models. It is common to select a short training duration for the fine-tuning process.

Figures 6, 7, 8, 9 compare the results (accuracy, log(training time), MAE, and MSE) from all the model variants (ADAM, SGDM and RMSPROP). ResNet-18 is clearly shown to have the highest average accuracy and lowest MAE, despite being slightly longer to train than AlexNet in training time. ResNet-18 is also amongst the best performing models for MSE, bested only by ResNet-101 trained with SGDM. It therefore concludes that ResNet-18 is the best performed CNNs at learning a new task (regression output of normalized tool wear state) from images of tools using transfer learning.

From the above analysis, the prediction workflow of tool health state based on transfer learning enabled ResNet-18 variants can be more effective in early stages (i.e., good tool health). However, data imbalance and overfitting have considerable negative impacts on prediction accuracy, where classes are not uniformly distributed across the dataset. This is evident in the collected data in this research, where many more examples of healthy tools (i.e., $v < 0.4$) are available than those of less healthy tools ($v \geq 0.4$). This is made further apparent in Fig. 10 that shows the distribution of the normalized training and testing dataset targets; there are much fewer values

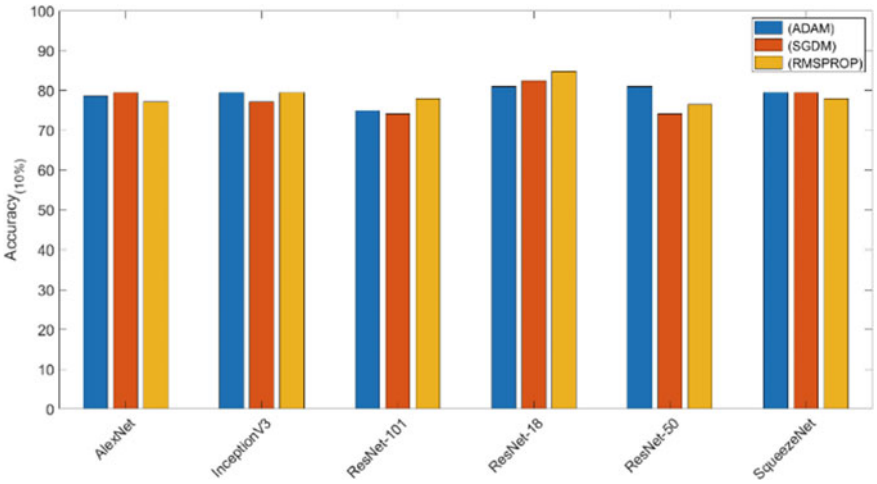

Fig. 6 CNNs models' accuracy for all training variants

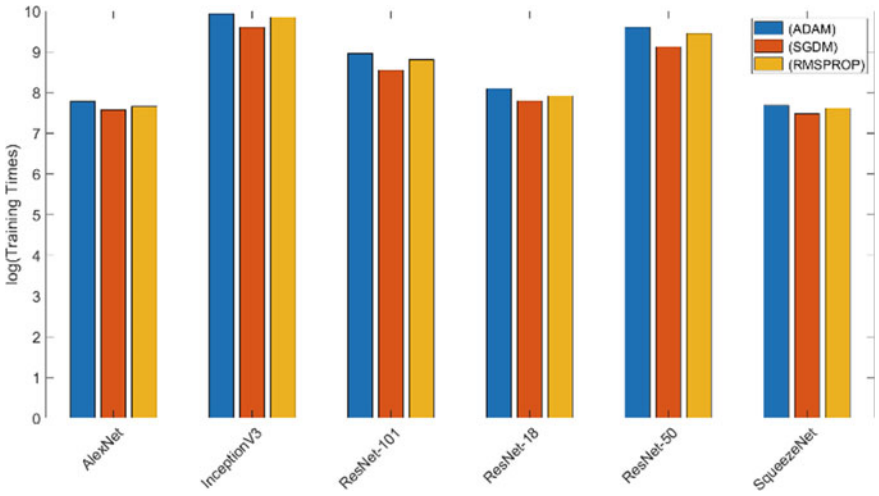

Fig. 7 CNNs models' log (training times) for all training variants

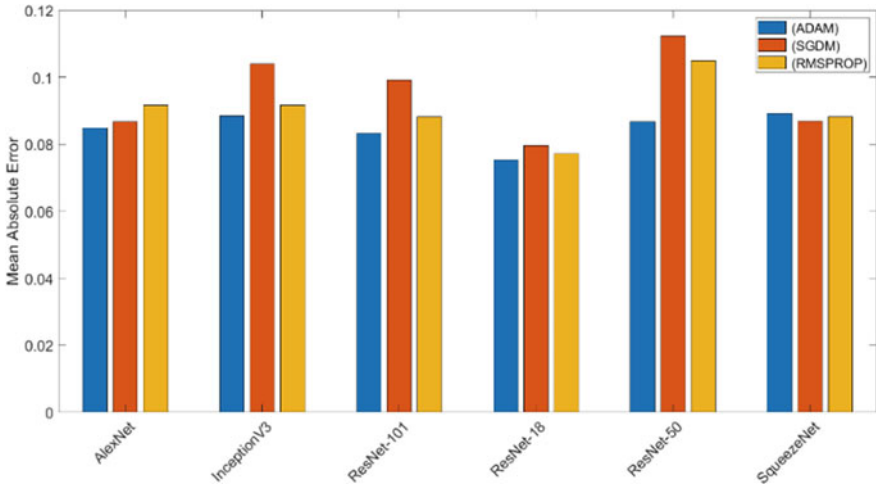

Fig. 8 CNNs models' Mean Absolute Error (MAE) for all training variants

close to 1 in the normalized scale, corresponding to v values close to 0.4. Therefore, further investigations should be made:

- To address the imbalance between healthy and faulty tool states, classification-then-regression methods could be further explored, where weights are assigned based on class probability. Alternatively, cumulative attribute-based methods could help improve accuracy by reformulating the regression problem in a manner similar to classification. Another alternative could be explored based on parameter

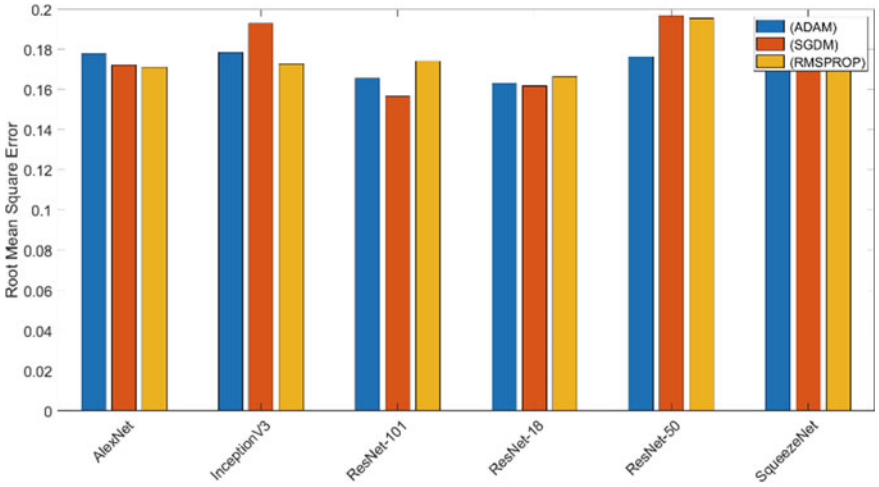

Fig. 9 CNNs models' Mean Square Error (MSE) for all training variants

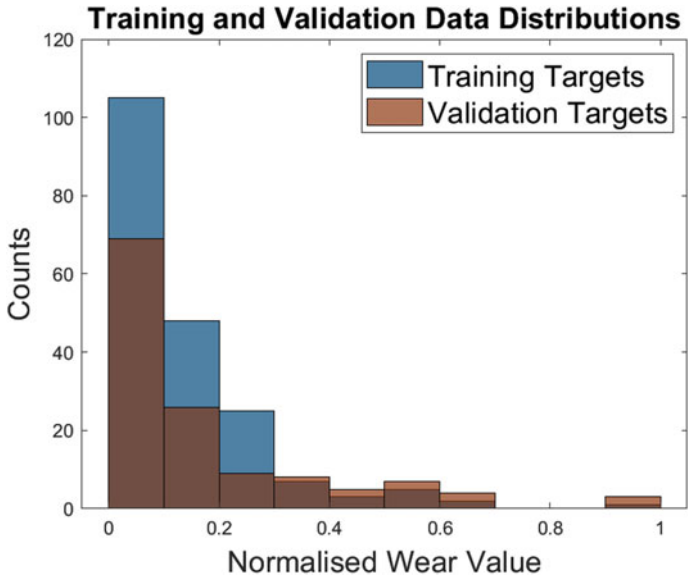

Fig. 10 The data distribution of the training and validation target data

transfer approaches, where a source task (and its corresponding source domain data) is used to pre-train the model.

- Additional works are required to improve the accuracy of prediction across increasing wear levels (i.e., where the normalized wear value exceeds 0.5). Some additional pre-manipulations of the data need to be implemented, by adding extra

safety margins to the hand-measured wear values, for example. Increasing the cost parameters for the regression layer, for example by increasing regularization L2-norm penalties, could reduce overfitting.

- Investigating maximum likelihood estimation methods for regression could help with improving predictions across the full range of expected outputs, thereby reducing prediction bias.
- As a supplement or alternative to these aforementioned approaches, some techniques to re-balance the classes of datasets, or implement class penalties using target RUL functions, could enhance the accuracy of the CNNs-based regression workflow.
- An end-to-end approach to incorporate additional model inputs such as machining parameters or categorical labels, combined with CNNs tool wear estimation, could be developed.

5 Conclusions

Deep learning algorithms have been increasingly applied for PHM due to their great potentials in the applications. Nevertheless, they are still ineffective in practical manufacturing applications as sufficient amounts of training dataset are not usually available. Seeking to overcome these limitations, in this chapter, a transfer learning enabled CNNs approach is developed to effectively predict tool RUL in CNC machining processes based on a limited number of the images of cutting tools. Quantitative benchmarks and analysis are conducted on the performance of the developed approach using several typical CNNs models and training optimization techniques. Experimental results indicate that the transfer learning approach, particularly using ResNet-18, can predict the health state of the cutting tool (as a normalized value between 0 and 1) with up to 84% accuracy and with a prediction mean absolute error of 0.0773. Based on these results, it demonstrates that the developed approach can achieve effective predictions on the health state of cutting tool in the early stages of tool wear.

A further research work is to integrate additional information to predict the tool RUL for increased accuracy (such as temperature, power dissipation, or current signals from the machine). The applicability of the methodology developed in this approach is not restricted to PHM alone; it could be used for other applications with only limited datasets in a target domain are available.

References

1. Compare M, Bellani L, Zio E (2017) Reliability model of a component equipped with PHM XE "PHM" capabilities. Reliab Eng Syst. Saf 168:4–11
2. Lu B, Zhou X (2017) Opportunistic preventive maintenance scheduling for serial-parallel multistage manufacturing systems with multiple streams of deterioration. Reliab Eng Syst Saf 168:116–127
3. Li X, Zhang W, Ding Q (2019) Deep learning XE "Deep learning" -based remaining useful life estimation of bearings using multi-scale feature extraction. Reliab Eng Syst Sa. 182:208–218
4. Oh JW, Jeong J (2019) Convolutional neural network and 2-D image based fault diagnosis of bearing without retraining. Proceedings of the 3rd international conference on compute and data analysis - ICCDA 2019, pp 134–138
5. Hoang DT, Kang HJ (2019) Rolling element bearing fault diagnosis using convolutional neural network and vibration image. Cogn Syst. Res 53:42–50
6. Zhao R, Yan R, Wang J, Mao K (2017) Learning to monitor machine health with convolutional Bi-directional LSTM XE "LSTM" networks. Sensors 17(2):273
7. Ghani JA, Rizal M, Nuawi MZ, Ghazali MJ, Haron CHC (2011) Monitoring online cutting tool wear using low-cost technique and user-friendly GUI. Wear 271(9–10):2619–2624
8. Lei Y, Li N, Guo L, Li N, Yan T, Lin J (2018) Machinery health prognostics: A systematic review from data acquisition to RUL XE "RUL" prediction. Mech Syst Signal Process 104:799–834
9. Johansson D, Hägglund S, Bushlya V, Ståhl JE (2017) Assessment of commonly used tool life models in metal cutting. Procedia Manuf. 11:602–609
10. Zhao R, Yan R, Chen Z, Mao K, Wang P, Gao RX (2019) Deep learning XE "Deep learning" and its applications to machine health monitoring. Mech Syst Signal Process 115:213–237
11. Deng J, Dong W, Socher R, Li L-J, Li K, Li FF (2010) ImageNet: A large-scale hierarchical image database. Proceedings of the 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255.
12. Russakovsky O et al (2015) ImageNet large scale visual recognition challenge. Int J Comput Vision 115:211–252
13. Krizhevsky A, Nair V, Hinton GE (2020) CIFAR-10 and CIFAR-100 datasets. https://www.cs.toronto.edu/~kriz/cifar.html. Accessed 10 Jan 2020
14. Janssens O, Van De Walle R, Loccufier M, Van Hoecke S (2018) Deep learning XE "Deep learning" for infrared thermal image based machine health monitoring. IEEE/ASME Trans Mechatronics 23(1):151–159
15. Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22:1345–1359
16. Weiss K., Khoshgoftaar T.M., Wang D.D., 2016. A survey of transfer learning. J. Big Data. 3(1).
17. Gouarir A, Martínez-Arellano G, Terrazas G, Benardos P, Ratchev S (2018) In-process tool wear prediction system based on machine learning techniques and force analysis. Procedia CIRP 77:501–504
18. Hsu C-SS, Jiang J-RR (2018) Remaining useful life estimation using long short-term memory deep learning. Proceedings of the 2018 IEEE international conference on applied system invention (ICASI), pp 58–61
19. PHM Society (2010) 2010 PHM society conference data challenge. https://www.phmsociety.org/competition/phm/10. Accessed 15 Jan 2020
20. Günther J, Pilarski PM, Helfrich G, Shen H, Diepold K (2014) First steps towards an intelligent laser welding architecture using deep neural networks and reinforcement learning. Procedia Technol 15:474–483
21. Saxena A, Goebel K, Simon D, Eklund N (2008) Damage propagation modelling for aircraft engine run-to-failure simulation. Proceedings of the 2008 international conference on prognostics and health management (PHM 2008), pp. 1–9.
22. Lu C, Wang Z, Zhou B (2017) Intelligent fault diagnosis of rolling bearing using hierarchical convolutional network based health state classification. Adv Eng Inform 32(139–151)

23. Li Z, Wang Y, Wang K (2019) A deep learning driven method for fault classification and degradation assessment in mechanical equipment. Comput Ind 104:1–10
24. Zhang J, Wang P, Yan R, Gao RX (2018) Learning improved system remaining life prediction. Procedia CIRP 72:1033–1038
25. Shi C, Panoutsos G, Luo B, Liu H, Li B, Lin X (2018) Using multiple feature spaces-based deep learning for tool condition monitoring in ultra-precision manufacturing. IEEE Trans Ind Electron 66:1–1
26. Liu X, Li Y, Chen G (2019. Multimode tool tip dynamics prediction based on transfer learning. Robotics and Computer Integrated Manufacturing. 57: 146–154.
27. Zhou B, Khosla A, Lapedriza A, Torralba A, Oliva A (2016) Places: an image database for deep scene understanding. https://arxiv.org/abs/1610.02055. Accessed 08 Jan 2019
28. Lu W, Liang B, Cheng Y, Meng D, Yang J, Zhang T (2017) Deep model based domain adaptation for fault diagnosis. IEEE Trans Ind Electron 64(3):2296–2305
29. Xiao D, Huang Y, Zhao L, Qin C, Shi H, Liu C (2019) Domain adaptive motor fault diagnosis using deep transfer learning. IEEE Access 7:80937–80949
30. Wen L, Gao L, Li X (2019) A new deep transfer learning based on sparse auto-encoder for fault diagnosis. IEEE Trans Syst, Man, Cyber: Syst 49:136–144
31. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. Proc IEEE Conf Comput Vis Pattern Recognit 770–778.
32. Szegedy C et al (2015) Going deeper with convolutions. Proceedings of the IEEE computer society conference on computer vision and pattern recognition, pp 1–9
33. Qian N (1999) On the momentum term in gradient descent learning algorithms. Neural Networks 12(1):145–151
34. Tieleman T, Hinton GE, Srivastava N, Swersky K (2012) Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA Neural Networks Mach. https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf. Accessed 10 Jan 2019
35. Mausser H (2019) Normalization and other topics in multi-objective optimization. Proceedings of the fields–MITACS industrial problems workshop, pp 59–101
36. Ruder S (2016) An overview of gradient descent optimization algorithms. https://arxiv.org/abs/1609.04747. Accessed 10 Jan 2019
37. Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K (2016) SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. Proc Int Conf Comput Vis Pattern Recognit