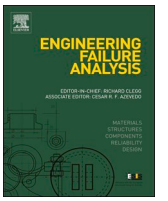
Engineering Failure Analysis 119 (2021) 104908

Contents lists available at ScienceDirect

Engineering Failure Analysis

journal homepage: www.elsevier.com/locate/engfailanal



Classification of fatigue crack damage in polycrystalline alloy structures using convolutional neural networks

Hassan Alqahtani[a,b,*], Skanda Bharadwaj[c], Asok Ray[a,d]

[a] Department of Mechanical Engineering, Pennsylvania State University, University Park, PA 16802, United States
[b] Department of Mechanical Engineering, Taibah University, Medina 42353, Saudi Arabia
[c] School of Electrical Engineering and Computer Science, Pennsylvania State University, University Park, PA 16802, United States
[d] Department of Mathematics, Pennsylvania State University, University Park, PA 16802, United States

ARTICLE INFO

Keywords:
Fatigue damage
Crack tip opening displacement
Convolutional neural networks
Image augmentation and classification

ABSTRACT

This paper proposes an autonomous method for detection and classification of fatigue crack damage and risk assessment in polycrystalline alloys. In this paper, the analytical and computational tools are developed based on convolutional neural networks (CNNs), where the execution time is much less than that for visual inspection, and the detection and classification process is expected to be significantly less error-prone. The underlying concept has been experimentally validated on a computer-instrumented and computer-controlled *MTS* fatigue testing apparatus, which is equipped with optical microscopes for generation of image data sets. The proposed CNN classifier is trained by using a combination of the original images and augmented images. The results of experimentation demonstrate that the proposed CNN classifier is able to identify the images into their respective classes with an accuracy greater than 90%.

1. Introduction

Fatigue damage is a major source of failures in mechanical structures; the main factors, which induce such failures, include: a large number of applied stress cycles, maximum tensile stress, and fluctuations in the applied stress. In general, the maximum stress accrues from discontinuities in both external surfaces (e.g., notches and fillets) and internal microstructures (e.g., inclusions and voids). These defects are often produced during the manufacturing process, and they are randomly distributed in polycrystalline alloy (e.g., metallic) structures.

Fatigue damage in ductile materials (e.g., polycrystalline alloys) can be classified into three consecutive stages [1]. The first stage is the crack initiation, where the actual crack growth rate may not be significant (e.g., ~1 nm per cycle); therefore, the fatigue damage risk is low. The second stage is the crack propagation, and it starts after the first stage and propagates in a direction approximately normal to the applied load. Here, the crack growth rate is low at the beginning and becomes high at the end of the stage. Thus, the fatigue damage risk is considered to be low at the beginning and gradually becomes high at the end of the stage. The third stage is defined when the crack becomes sufficiently long, where the remaining cross-section area othe stressed component may not be able to sustain the applied load, and the fatigue damage risk is very high [2,1,3].

Damage estimation in mechanical structures belongs to the class of engineering maintenance. Two approaches provide damage measurements in metallic structures. The first approach is to identify the maximum applied load that would prevent the structure from failing immediately, where the crack geometry (e.g., length, location, and orientation) is known in addition to boundary

* Corresponding author.
E-mail address: hhq1408@gmail.com (H. Alqahtani).

https://doi.org/10.1016/j.engfailanal.2020.104908
Received 3 June 2020; Received in revised form 21 July 2020; Accepted 11 September 2020
Available online 14 October 2020
1350-6307/

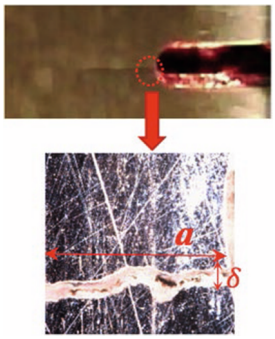


Fig. 1. The location of the investigated area, the crack tip opening displacement (CTOD) δ, and the crack length (CL) a.

conditions. The second approach is to determine the maximum length of a crack that would not cause the structure to fail immediately. The second approach is defined as damage tolerance. From the perspectives of damage tolerance, once a crack length is determined, the damaged component must be periodically inspected with appropriate non-destructive testing (NDT). However, when a crack length becomes critical, the structural part must be fixed or replaced where further crack growth could cause catastrophic failure of the fatigued component [4].

This paper defines the damage tolerance based on the crack tip opening displacement (CTOD) δ, instead of the crack length (CL) a. Fig. 1 shows how CTOD is dependent on CL that increases with expanding CTOD. Therefore, in this paper, CTOD has been used as the source of information for determining the damage risk of the fatigued component [5].

In this paper, the defined damage risk of the fatigued component may belong to two classes. The first class is the non-risk class, where the component surface is apparently crack-free, as shown in Fig. 2*a*. The second class is the risk class, and it is divided into two subclasses. Fig. 2*b* represents the first subclass, which is the low-risk damage class. The crack growth rate in the low-risk class is slow, and the crack tip opening displacement is small. Furthermore, the shape of the cracked surface, after the failure of a fatigued component, is a noticeable "thumbnail," as illustrated in Fig. 3. The second subclass is defined as high-risk damage class, where the crack growth rate is high and the crack opening size is significant, as shown in Fig. 2*c*. The region of high-risk damage starts after "thumbnail" as illustrated in Fig. 3 [5]

According to some investigators (e.g., [6]), more than two-thirds of all inspections in aviation industry are essentially made by visual inspection. Although the visual inspection is capable of detecting a variety of defects (e.g., fatigue cracks) in machinery components, damage detection accuracy may vary from person to person, depending on the vision of the inspector, in addition to various other factors, such as poor access, lighting, geometry, and size of the area being inspected. Furthermore, determining the damage-risk of the fatigued component depends on the inspector's experience; in most of the cases, it is a time-consuming task. Thus, automated crack detection is deemed very necessary.

Recently, much research on image-based algorithms has been reported around the theme of crack detection [7–9]. From these perspective, tools of digital image processing (e.g., edge detection and thresholding) are capable of crack detection in images, where the properties of crack images are analyzed based on photometric and geometric hypotheses [10]. For example, the location of a crack can be defined based on the photometric property, where crack pixels have distinct values from other pixels. Thus, a threshold value can be defined to separate cracks from the background [10–12]. One of the major disadvantages of these methods is that the noise on the image could affect the accuracy of crack detection. To overcome this problem, mathematically techniques (e.g., discrete wavelet transforms) are used to remove noise from the images. The discrete wavelet transform can distinguish between crack regions and crack free regions based on their scales, where scales of the crack regions be different from those of crack-free regions [13]. Although the above-mentioned techniques can detect cracks on material structure, they do not have the capability to classify crack images.

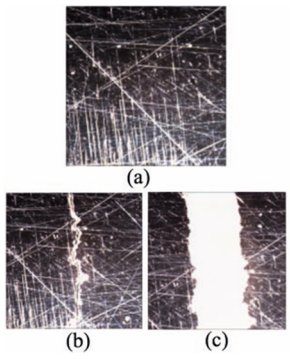


Fig. 2. (a) Crack free surface, (b) small-crack, and (c) large crack.

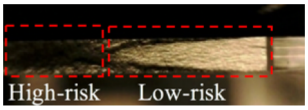


Fig. 3. Low risk region (thumbnail), and high-risk region.

Recent years have witnessed a growing academic interest in detecting fatigue cracks using machine learning methods [7]. In addition to the capability of crack detection, the machine learning methods are able to classify the images to their proper classes. Some investigations have proved that machine learning methods are able to classify non-crack images and crack images, and the performance of these methods was excellent, their accuracy exceeded 90% [14]. However, most of these investigations have not used the image pixel data directly for classification, nor they have classified the crack images into further sub-classes. For example, one study used the histogram of the image as the input data for classification instead of the original image pixels data [14]. Another study used a convolutional neural network (CNN) to detect the crack based on the classification of the image, which means that this study has only two classes, crack-free images and cracked images [10]. Along this line, this paper proposes the concept of CNN to classify the status of fatigued surfaces into their proper classes by using image features.

Convolutional Neural Networks (CNNs) belong to a special class of artificial neural networks that are used for object detection and classification in images [15,16]. They are also referred to as deep neural networks due to the presence of a large number of hidden layers. Therefore, a CNN can be viewed as a deep learning network that takes an input image and classifies the image into a particular class. In any artificial neural network, layers are made up of a fundamental unit called the neuron. A neuron in a CNN is simply a kernel, or in other words a filter with which the input image is going to be convolved with. These kernels are stacked in a cascaded way to form a layer called the convolutional layer. Multiple such layers form a deep network. Since the input image is convolved with the filters to produce feature maps, such a network is called a Convolutional Neural Network. It is noted that CNNs are capable of classifying a large number of classes [16], and they may also be very efficient for image recognition [17,18]. Owing to the high performance of CNNs, there are now being used not only for image classification, but also for object detection. Apparently Girshick et al. [19] first proposed the use of CNNs for object detection within images. Later, efficient and fast performing models for visual object detection were developed [20–22]; CNNs are also capable of sub-pixel accurate segmentation of objects [23].

In this paper, all images have been generated using an optical metrology device. A major concern here is the low number of images, where operations of deep convolutional neural networks have to be performed well on a big data set by alleviating the problem of overfitting. Image augmentation is a possible solution to this problem, where the generated images need to be augmented by applying image processing techniques such as image translation, image rotation, image scaling, and image reflection [24]. The total of 255 images have been selected from those that were generated by an optical metrology device (Alicona). These images are then split into a training set and a testing set. Moreover, a part of the training set is further split into two subsets, i.e., training subset and cross-validation subset. The training set is used to generate parameters to build a problem model. In contrast, the cross-validation subset is used to evaluate the performance of the developed model from the training subset; thus, the entire learning phase consists of these two subsets. When the learning phase is finalized, the test set is used to compute the accuracy of the learning phase. Data from several studies suggest that the distribution for limited data should be as follows: 60% of the available data for training; 20% for cross-validation set; and the remaining 20% is for testing [25].

The novelty of the paper includes: (i) development of an autonomous method for detection and classification of fatigue crack damage in polycrystalline alloy structures (e.g., machinery components); and (ii) experimental validation of the associated risk assessment algorithms that are applicable to remaining life prediction in various types of machinery. Although the underlying algorithms are validated on a test apparatus for a polycrystalline alloy, which is a ductile material (in the room temperature), they are expected to be functional for other types of materials (e.g., brittle) when trained with appropriate data sets with the same or a similar CNN architecture. This is suggested as topic of future research in Section 5.

From the above perspectives, the main contributions of the work reported in this paper with respect to those in the existing literature are delineated below.

1. *Damage classification*: The damage tolerance of a fatigued component is classified using convolutional neural networks (CNNs) into three categories, namely, non-risk, low-risk, and high-risk.
2. *Data augmentation*: The ensemble of original input data is augmented using digital image processing methods such as translation, rotation, and reflection.
3. *Model performance evaluation*: The performance of the CNN model is evaluated using the Soft-Max algorithm and loss function.

The paper is organized into five sections including the present section. Section 2 describes the methodology adopted in thsi paper, including the information on convolutional neural networks (CNNs) that form the backbone of the work presented in this paper; this section consists of the following three subsections: descriptions of CNN methodologies; concepts of CNN architectures; and demonstration of data bank augmentation. Section 3 describes the experimental procedure on a test apparatus and the associated instrumentation. Section 4 discusses how the proposed CNN architecture evaluates test images and illustrates the accuracy of the proposed CNN architecture. Section 5 summarizes and concludes the paper with recommendations for future research.

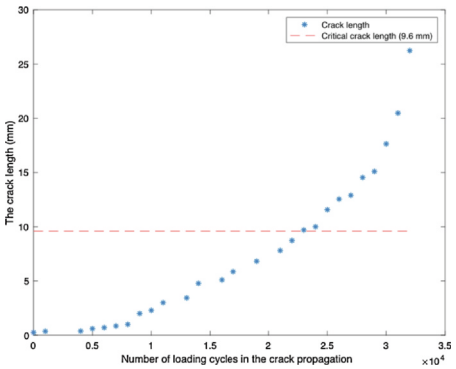


Fig. 4. Crack length in the propagation regime.

2. Methodology

This section develops the methodology for determination of different levels of risk resulting from fatigue crack failures in machinery. Three essential steps are considered to determine the risk level of the fatigued component for structural damage, as described below.

1. The first step is determination of the critical crack length (CCL), which can be calculated using the following equation:

$$a_c = \frac{K_{IC}^2}{\pi \sigma_{max}^2 Y_1^2} \tag{1}$$

where a_c is the critical crack length (CCL); K_{IC} is the fracture toughness (for aluminium 7075, K_{IC} =20.0 MPa-$m^{0.5}$); σ_{max} is the maximum applied stress;and the dimensionless parameter Y_1 is a function of the crack length a and width w of the tested specimen. For edge cracks, loaded in tension, Zahavi et al. [26] proposed the following equation to calculate Y_1.

$$Y_1 = 1.12 - 0.231\left(\frac{a}{w}\right) + 10.55\left(\frac{a}{w}\right)^2 - 21.72\left(\frac{a}{w}\right)^3 + 30.39\left(\frac{a}{w}\right)^4$$

where the CCL in all specimens (see Eq. (1)) is a_c≈9.6 mm. As shown in Figs. 4 and 5, the crack growth behavior before CCL (red dashed line) is slow and almost stable while, after CCL, the crack growth rate is higher and unstable. Therefore, all measurements before CCL are considered to be at the low-risk level while, after CCL, any measured value is considered to be at the high-risk level.

2. The second step is to measure the crack tip opening displacement (CTOD) δ, synchronized with the crack length (CL) a, because CTOD is illustrated in all generated images of the cracked tested specimen. Fig. 6 represents the relationship between CL and CTOD in the crack propagation regime. As Fig. 6 indicates, the relationship between the crack tip opening displacement and the crack length is strongly correlated. This correlation is determined using the following equation:

$$r_{(a,\delta)} = \frac{\sum_{i=1}^{n}\left[\left(a_i - \bar{a}\right)\left(\delta_i - \bar{\delta}\right)\right]}{\sqrt{\sum_{i=1}^{n}(a_i - \bar{a})^2 \sum_{i=1}^{n}(\delta_i - \bar{\delta})^2}} \tag{2}$$

where $\bar{a}$ and $\bar{\delta}$ are the average CL and the average CTOD, respectively.

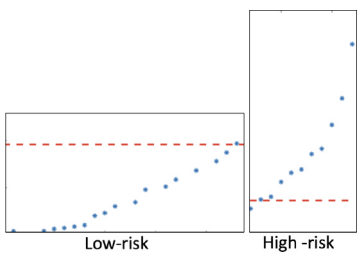


Fig. 5. The crack growth behavior of low-risk and high-risk.

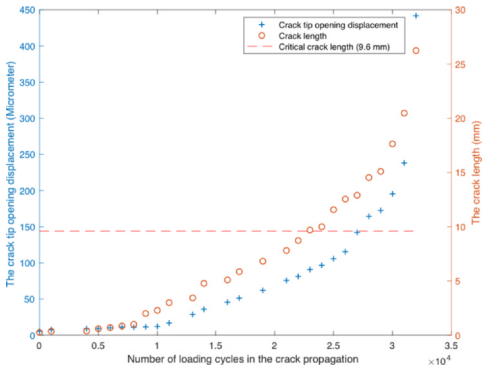


Fig. 6. The relationship between the crack length and the crack tip opening displacement.

$$\bar{a} = \frac{1}{n}\sum_{i=1}^{n}(a_i) \quad \bar{\delta} = \frac{1}{n}\sum_{i=1}^{n}(\delta_i) \tag{3}$$

The correlation coefficient $r = 0.96$ using Eq. (2), which means that CL and CTOD have a very strong positive correlation. Therefore, the decision on risk category determination of CL can be applied to CTOD, where CTOD before critical crack length (CCL) belongs to a low-risk category, and it belongs to a high-risk category after the CCL.

Since CTOD is illustrated in all generated images of cracked component, these images can be evaluated to determine the damage risk category.

3. The third step is data preparation. The generated images are split into three categories based on damage risk. The non-risky category contains crack-free images; the low-risk category represents generated images before the critical crack length; and, the high-risk category includes generated images after the critical crack length. The classification hierarchy of this section is illustrated in Fig. 7.

In this study, the essential condition that must be considered during the experiment is that the fatigue load must be at the maximum capacity (e.g., 11,000 N) for all images because the applied load has a significant effect on the crack (e.g., Eq. 1) and the crack length and CTOD have a positive correlation. Hence, the maximum applied load causes the largest crack tip opening displacement. For example, Fig. 8 and Fig. 9 represent two different classification cases. In the first case, the crack tip opening displacement is 80.54 μm at 29,602 loading cycles, which is considered to be a high-risk class because it is after the critical crack length. On the other hand, in the second case, at 30,096 loading cycles, the crack tip opening displacement is 44.39 μm which could be considered to be a low-risk class while in fact, it is a high-risk class because it is measured after the critical crack length. In the second case, CTOD should be greater than 80.54 μm. The reason for the miss classification between Fig. 8 and Fig. 9 cases is that during generating the image, the applied load in the first case was more significant than the applied load in the second case.

2.1. Data augmentation

The size of the generated image by Alicona is $2040 \times 2040 \times 3$, which means each image contains $4,161,600 pixels \times 3$. As a result, the input data to CNNs classifier is 12,484,800, which means the computation process for classifying each image becomes complicated. Thus, each image should be resized such that the main image features (e.g., cracks and scratches) are still illustrated in the image, and the time for the computation process is acceptable.

In this study, three techniques were applied to all images before using the CNN method. The first technique is reducing the channel number of the generated images where all images are converted from RGB scale to grayscale. As shown in Fig. 10*a*, all

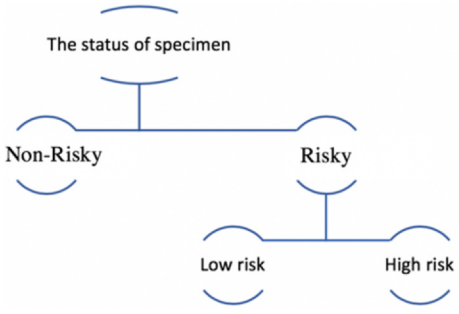


Fig. 7. The classification hierarchy.

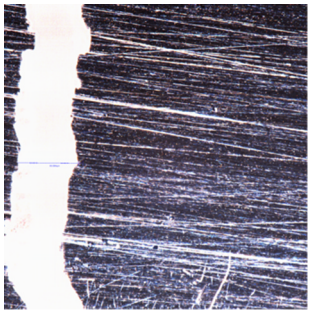


Fig. 8. The crack tip opining displacement at 29,602 loading cycles.

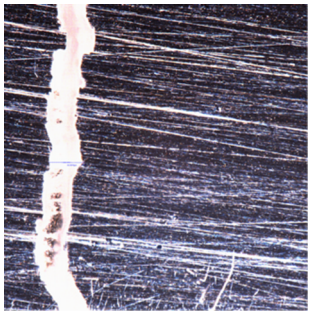


Fig. 9. The crack tip opining displacement at 30,096 loading cycles.

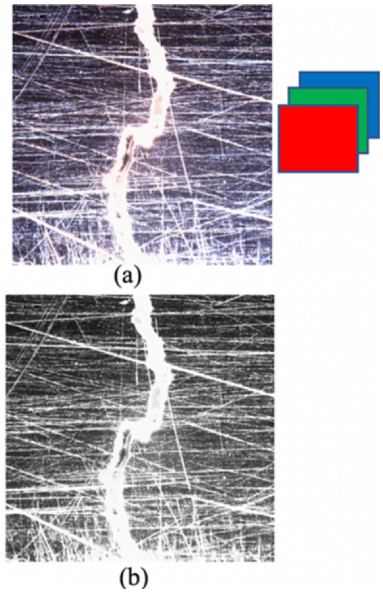


Fig. 10. (a) RGB image, (b) grayscale image.

generated images are RGB (red, green, and blue) images, which means that all images have three channels, while images in grayscale have only one channel, as shown in Fig. 10*b*.

The reason for reducing channel number is to minimize the image data from 12,484,800 pixels to 4,161,600 pixels. In this study, the unique property of crack images is those crack pixels are brighter than the surrounding pixels. As shown in Fig. 10*b*, the crack path is noticeable because crack pixels intensity values are higher than the intensity values of the surrounding area. The second step is reducing image size. Determining the optimal image size is a difficult task, and it is not part of this study. However, our objective for the second step is to minimize the image data by reducing image pixels resolutions where the main features, such as the cracks, are still visually illustrated in the image because we need to distribute images into different classes. As shown in Fig. 11*a* and Fig. 11*c*, it is difficult to distinguish between the tiny crack, which is shown in the bottom of the image and the scratches, but small cracks could

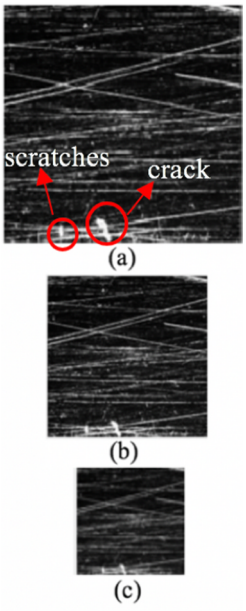


Fig. 11. (a) Image size is 150 × 150, (b) image size is 128 × 128, (c) image size is 64 × 64.

be seen when image size is 150 × 150. Thus, all grayscale images are resized from 2040 × 2040 to 150 × 150 (see Fig. 11*b*). Fig. 12 presents the original image.

The third step is the image augmentation. Four different methods augmented all images. The first method is the rotation; every image produced 37 different images by rotation, where each image has its unique rotation angle. Fig. 13 represents the rotational method where the locations of original image pixels were rotated by almost 90 °. The second method is the transition; every image was augmented 21 times by transition, where the original image was shifted by a distinct distance. As shown in Fig. 14, the locations of original image pixels were moved into the right by a distance equivalent to 14 pixels. Fig. 15 represents the third method, which is the reflection, where every image generated its reflected image. The fourth method is called the scaling; this method creates 31 different images. As shown in 16, the original image was scaled by 6%. In the scaling method, the output image must be lie within its class. For all augmented images, main image features (e.g., crack) must be illustrated.

2.2. The Convolutional Neural Network (CNN)

This subsection briefly introduces the underlying concept of convolutional neural network (CNN) that is comprised of various layers, such as convolution layer (CL), pooling layer (PL), activation layer (AL), fully connected layer (FCL) and Softmax layer (SML). Different CNN networks can be created by several combinations of these layers.

Fig. 17 represents a CNN with five layers, which takes an input image of size 128 × 128 × 1, representing height, width, and channel, respectively; here, channel = 1 implies that the considered image is of grayscale. As the input image passes through different layers of the CNN before classification, the spatial dimension of the image reduces. In this example, a 128 × 128 × 1 reduces to 11 × 11 × 1 image. Due to the presence of a number of filters several such low resolution images called feature maps are formed before passing through the fully connected layer (FCL). These images are “flattened” to form a 1*D* column vector as an input to the FCL. The output scoring for each particular class which is finally converted to probability scores by the softmax layer, which represents final out of the network. The details of individual layers are explained in the next subsections.

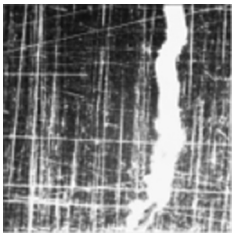


Fig. 12. The original image.

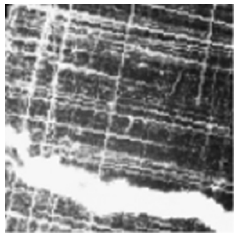


Fig. 13. The rotated image.

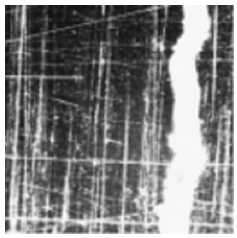


Fig. 14. The translated image.

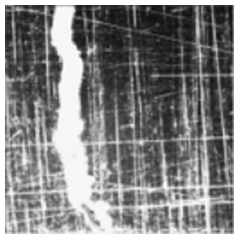


Fig. 15. The reflected image.

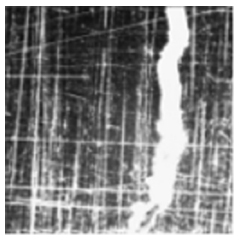


Fig. 16. The scaled image.

2.2.1. Convolution Layer (CL)

The CL is the most important layer in the CNN, which is also the most computation-intensive. A CL consists of several learnable filters that are spatially small of size $h \times w \times n$ where h and w represent the height and width of the filter and n is the channel number of in the input image. Typical choices of (h, w) for filters are 3×3 or 5×5. The forward pass convolves the image with the filter to create a $2D$ activation map. This $2D$ convolution of two signals is defined as:

$$\begin{aligned} &\mathrm{h(m,\ n)} = \mathrm{g(m,\ n)} \star f(m,\ n) \\ &= \sum_{i=-k}^{k} \sum_{j=-k}^{k} f\left(i, j\right) g\left(m - i, n - i\right) \end{aligned}$$

where the operator ☆ represents the convolution product of two functions; $h(m, n)$ is the convolved; $g(m, n)$ is the input image; and $f(m, n)$ is the filter. Fig. 18 exemplifies the convolution operation. The filter f slides over the entire image g to create the convolved output h. The corner cases are handled by padding zeros to the borders along the height and width; a stride determines how much does the filter slides during a step of the convolution operation. The stride is one of the hyperparameters, which can be used to fine tune the network. The initial values of the filters can be chosen arbitrarily and the filter values are updated during the training phase [16]. The network is trained using stochastic gradient descent method. As shown in Fig. 17the proposed network architecture consists

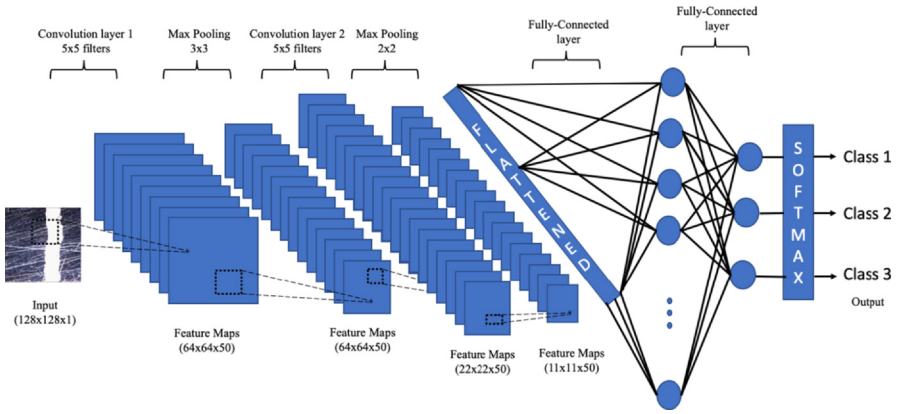


Fig. 17. Convolution Neural Network architecture used for classification. It consists of 2 convolutional layers with corresponding ReLU activation layers, 2 max pooling layers, 2 fully-connected layers and a softmax output layer.

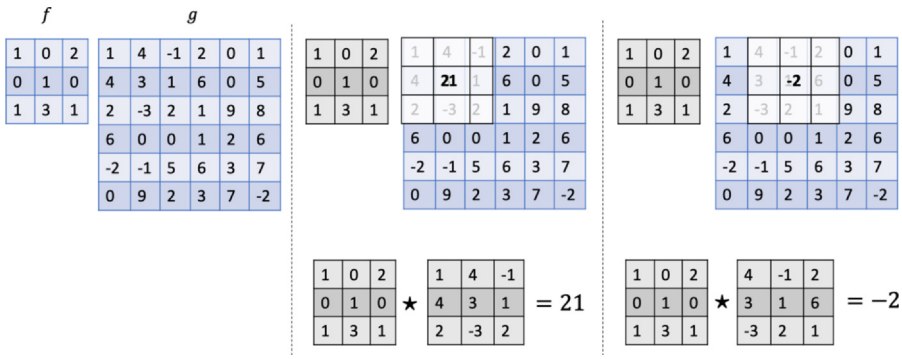


Fig. 18. The above figure represents an example of convolution. The kernel is slid over the entire image to produce a feature map. The stride length is a hyper parameter which can be used to tune the network. In this example the stride length is 1.

of 2 convolution layers.

2.2.2. Pooling Layer (PL)

The PLs are commonly placed between successive layers of a CNN. It plays a key role to progressively reduce the spatial size of the respective input images, i.e., the spatial size as the number of parameters is reduced, and therefore, the computational cost going ahead in the network. This is also referred to as downsampling. By performing pooling overfitting of the network can be controlled. Two major pooling techniques are mean pooling and max pooling. Given an $m \times n$ matrix, the max pooling takes the maximum of all the values, whereas the mean pooling takes the average of the matrix. Mean pooling and max pooling are explained in Fig. 19. In this example, the a stride of 2 is considered. That is, max is considered over matrices of size 2×2. It can be seen that a 4×4 matrix is now downsampled to a matrix of 2×2. However, the depth of the output remains unchanged. In [27] Scherer et al. proved that max pooling outperforms mean pooling for image datasets. In our architecture, after each convolution layer a max pooling layer is used. The first pooling layer has a stride of 3×3 and the second layer has a stride of 2×2.

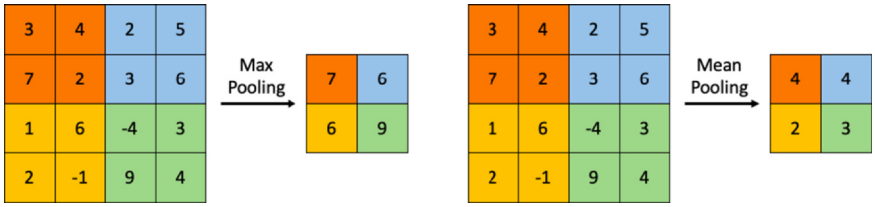


Fig. 19. In the above figure, the pooling example on the left represents max pooling and the pooling example to the right represents mean pooling. It can be seen that the pooling process downsamples the image to reduce the number of parameters and therefore the number of computations going ahead in the network.

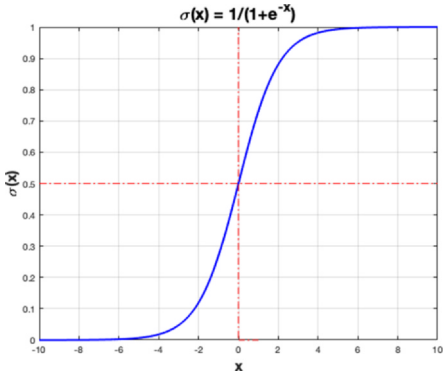


Fig. 20. Sigmoid function.

2.2.3. Activation layer

Activation layers are used to bring non-linearity into the network since the data that is being classified can have a non-linear distribution. The most common activation layer used is the sigmoid function. A sigmoid function is defined as

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

It squashes the real numbers into a range between [0, 1]. Fig. 20 represents the sigmoid function. But the major drawbacks of the sigmoid function are that they kill gradients by saturating and their outputs are not zero-centered. The other option is to use the $tanh()$ function. But this is also known to kill gradients.

Nair et al. [28] introduced Rectified Linear Unit (ReLU) as a non-linear activation function. ReLu is defined as

$$f(x) = \{0 \text{ if } x \leqslant 0 \; x \text{ if } x > 0 \tag{5}$$

Or, in other words, $f(x) = max(x, 0)$. It can be seen that ReLU has no bounded output on the positive side and also that the gradients of ReLU are always zeros and ones. Fig. 21 represents ReLU activation function. Activations layers are usually placed adjacent to convolutional layer. In our architecture, ReLU is used for both the convolutional layers.

2.2.4. Fully connected layer

A fully connected layer consists of neurons, also called perceptrons. Fig. 22 represents a perceptron. A perceptron takes several inputs and produces a single output. As shown in the figure, x_1, x_2, x_3 are inputs and Y is the output of the single perceptron. In general it could have more or fewer inputs. Each input is associated with real number called the *weight*, that signifies the importance of the respective input to the output. The neuron's output, 0 or 1, is determined by whether the weighted sum $\sum w_i x_i$ is less than or greater than some threshold value. These perceptrons are stacked to form the fully-connected layer. All the features maps obtained at the last convolutional layer are flattened and are connected to the fully-connected layer.

2.2.5. Softmax layer

Softmax layer is usually the last layer in the CNN architecture. This layer is used to convert the output of the fully connected layer into normalized class probabilities. The softmax function is defined by the equation given below

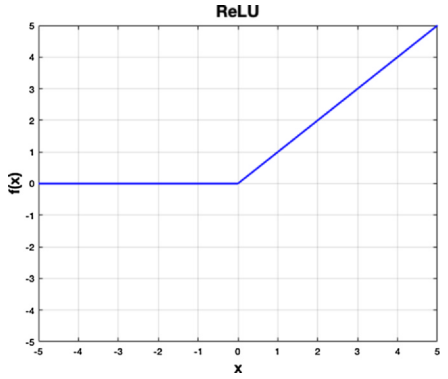


Fig. 21. ReLU.

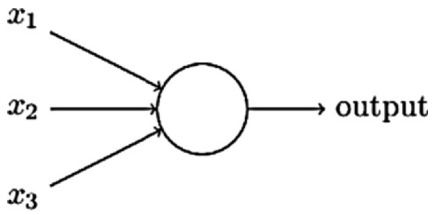


Fig. 22. Neuron (Perceptron).

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \tag{6}$$

It takes in a vector with real-values and outputs a vector with elements in the range [0, 1], which sum to one. In other words, a softmax layer outputs the probabilities of each output class.

2.3. Backpropagation: Cross-entropy loss function and mini-batch gradient descent

In the training phase, the initial values of the filters are randomly chosen and hence the predictions usually tend to deviate from the desired output. In order to calculate the amount of deviation, we define a loss function. The cross-entropy loss function of our architecture is described in the below equation

$$L = -\sum_i y_i log(p_i) \tag{7}$$

where, L is the loss function, y_i is the desired output and p_i is the probability of the i^{th} class. It is generally used when the output of the network is a probability distribution. In other words, when the output layer is a softmax layer. The intention behind training the network is to minimize the loss function. In order to minimize the loss function with respect to the input, the loss function back-propagates through the network. For this, we need to consider taking derivatives or gradients in each layer and update the weights in the filters. There are several algorithms that can be considered for updating the weights in the filters. The gradient descent using back-propagation is considered the most efficient and simplest way to minimize the deviations [29]. As explained earlier, gradient descent using backpropagation involves finding gradients in each layer and passing it on to the preceding layer. In gradient descent algorithm, the weights are updated after the entire dataset is shown to the network. A variant of the gradient descent called the stochastic gradient descent (SGD) can be used for faster convergence of the loss function. Here, weights are updated after ever sample is shown to the network. In order to speed up the process, weights are updated only after a batch of images are shown to the network. For instance, if the dataset has about 1000 images, weights are updated after a batch of 100 images are shown to the network. These batches are referred to as mini-batches and this variation is commonly known as the mini-batch gradient descent. Processing one mini-batch through forward pass and backpropagation is called an iteration. An epoch is when all the mini batches are processed. That is, once all images are processed one time. Our architecture uses the mini-batch mode of gradient descent algorithm. By taking the gradient of the loss function (∇L) with respect to the weights, we obtain the update equation.

$$\nabla L = p_i\left(y_i + \sum_{k\neq 1} y_k\right) - y_i \tag{8}$$

where, y is one-hot encoded vector for the labels, so $\sum_k y_k = 1$ and $y_i + \sum_{k\neq 1} y_k = 1$. And therefore,

$$\nabla L = p_i - y_i \tag{9}$$

The weights os the network are updates as shown below,

$$W_{ij} = W_{ij} - \alpha \nabla L \tag{10}$$

where, W_{ij} are the filter weights and α is the learning rate.

3. The experimental apparatus and procedure

This section describes the experimental procedure on a fatigue testing apparatus that is computer-instrumented and computer-controlled. The apparatus in Fig. 24, has been used to evaluate the damage risk of the cracked surface. Specimens were subjected to tension–tension load cycles at 60 Hz, and the load was generated by MTS test machine. The target setpoint of the load was 8,000 N. The applied loads fluctuated between 11,000 N and 6,000 N, where the amplitude load was 4,000 N. Five specimens of 7075-T6 aluminum alloy were used in this study. The dimensions of these specimens are 3 mm thickness, 50 mm width, and (1 mm ∗3.5 mm) slot cut at the edge as shown in Fig. 23.

Infinite-Focus is the trade name of the confocal microscop manufacturer Alicona, which is an optical metrology device that is used to provide 3D surface measurements, as shown in Fig. 25. The principle idea of the Focus-Variation system is that the topographical

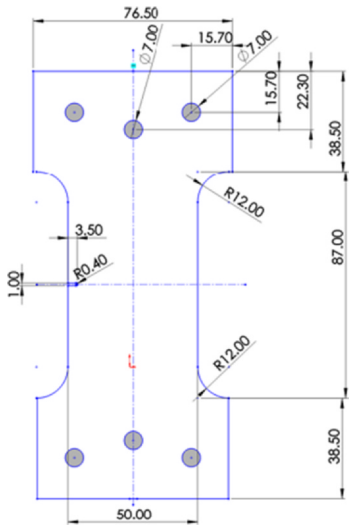


Fig. 23. Specimen geometry used for fatigue testing.

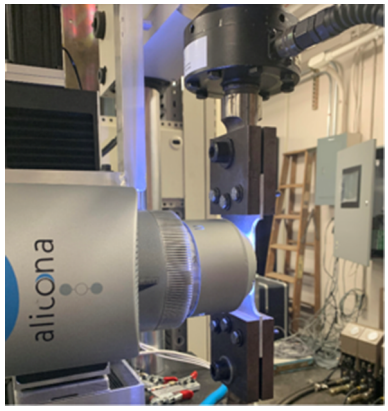


Fig. 24. The experimental apparatus used in this investigation.

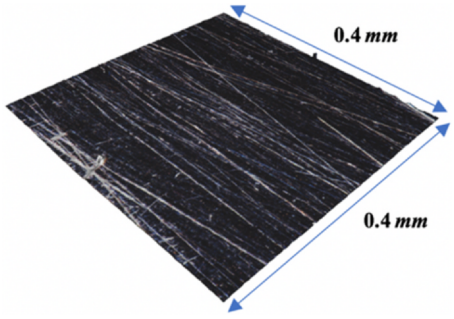


Fig. 25. 3D surface generated using Infinite-Focus deceive.

and color information are generated due to the combination between vertical scanning and the small depth of focus of an optical system. The vertical resolution of the Infinite-Focus system can be as low as 20 nm, and that improves the accuracy of surface measurement data. Furthermore, Infinite-Focus can provide huge data for a minimal surface area. In this study, as shown in the figure, the investigated area is 0.4 mm × 0.4 mm. This tiny area has 4,161,600 pixels. [30,31]. Table 1 shows technical specifications of Infinite- Focus device features.

The investigated area of this study is the area behind the notch, as shown in Fig. 26. Looking at Fig. 27 illustrates that the middle of the bottom frame line represents the location of the notch tip. The reason for not displaying the notch tip in the image is that the notch tip covers a portion of the image, as shown in Fig. 26b, and this portion is worthless.. The optical metrology device is located

Table 1
Technical specifications of Alicona.

Technical feature	Specifications
Vertical resolution	can be as low as 10 nm
Range of the vertical scan	3.2 mm - 22 mm
Measured area	0.14 mm × 0.1 mm – 5 mm × 4 mm

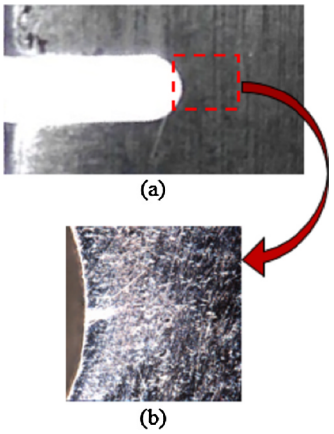


Fig. 26. (a) the location of the investigated area, (b) an image produced by an optical metrology device.

Fig. 27. The location of the notch tip on the image.

perpendicular to the investigated area, as shown in Fig. 28, and it measures CTOD by determining the distance between image pixels, as shown in Figs. 8, 9. The investigated area was chosen based on the location of the maximum stress on the specimen. In this study, the maximum stress exists on the notch of the specimen. The finite element analysis (FEA) using Soildwork was used to show the location of the maximum stress on the specimen. As shown in Fig. 29, the maximum stress clearly appears on the notch tip. Furthermore, the crack always propagates in the direction which is perpendicular to applied loads. Therefore, the investigated area of all

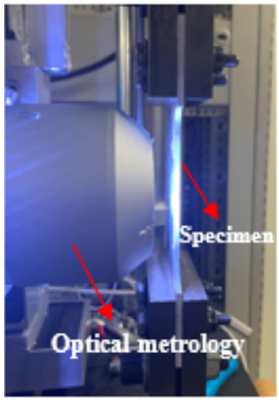


Fig. 28. Positions of the optical metrology device and the specimen.

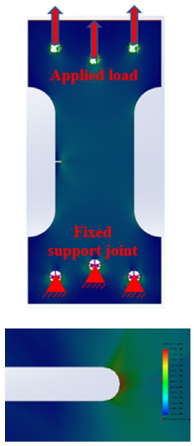


Fig. 29. The stresses analysis using Soliwork for investigated specimen.

crack images is almost fixed. It is thus suggested that damage tolerance of a mechanical structure can be defined for real-world applications in the following steps:

1. Creating a 3D model of the mechanical structure using computer aided design (CAD) software (i.e., Soildwork, Ansys, Abaqus).
2. Applying finite element analysis (FEA) on the created 3D model using one of CAD software.
3. Determining the critical locations (the maximum stress locations) on the 3D model.
4. Monitoring the condition of the critical locations in the real structure using an appropriate NDT method.
5. Asking a maintenance expert to evaluate the NDT results and to make a plan for the maintenance schedule.

4. Results and discussion

4.1. Data augmentation

From Table 2 and Table 3, the total number of original images is 255, while the total number of augmented images is 23,205 images, which means that each image has 91 distinct images.

4.2. Convolutional neural networks

In this paper a custom CNN was built to classify the images into three different classes namely, non-risky, low-risk, and high risk. Table 4 summarizes our custom cnn. Each input image was of the size ($128 \times 128 \times 1$). A total of 255 images were acquired using Infinite-Focus device, Alicona. This database of 255 images were increased to a database of 23, 205 by image augmentation. Image augmentation is a technique used to artificially create training images through different ways of processing or combination of multiple processing, such as rotation, shifts and shear. Image augmentation is required to boost the performance of deep networks when the available database has fewer images. Out of the 23, 205 images, 18, 837 images were used for training and 4, 368 were used for testing the CNN. Among the training images, 9, 191 images were labelled as healthy, 4, 004 images were labelled as large and 5, 642 images were labelled as small. Similarly, among the testing images, 2, 093 images were labelled as healthy, 1, 092 images were labelled as large and 1, 183 images were labelled as small. Table 5 summarizes configurations of the network and the outputs of the experiment. Batch size represents the number of images shown to the network in a single iteration. If there are n images in the database and the m is the batch size, then m images are shown to the network every iteration. Therefore, there are a total of n/m iterations that constitutes 1 epoch.

Figs. 30 and 31 represent the confusion matrices for training and testing, respectively, where the rows correspond to the output class or the predicted class and the columns correspond to the ground truth labels. The diagonal cells correspond to observations that are correctly classified. The off-diagonal cells correspond to incorrectly classified observations. The right-most column of the matrix

Table 2
Number of images before augmentation.

Class type	Train folder	Test folder	Total (class)
Non-risky	101	23	124
Low risk	62	12	65
High risk	44	13	75
Total (folder)	207	48	255

Table 3
Number of images before augmentation.

Class type	Train folder	Test folder	Total (class)
Non-risky	9191	2093	11284
Low risk	5642	1183	6825
High risk	4004	1092	5096
Total (folder)	18837	4368	23205

Table 4
Architecture of our custom CNN.

Layer	Activation Shape	Activation Size	#Parameter
Input	(128, 128, 1)	16, 384	0
Conv1(f = 5, s = 2, p = 2)	(64, 64, 50)	819, 200	1300
ReLU	–	–	–
Max Pool (s = 2)	(64, 64, 50)	204, 800	0
Conv2(f = 5, s = 2, p = 2)	(22, 22, 50)	24, 200	1300
ReLU	–	–	–
Max Pool (s = 2)	(11, 11, 50)	6, 050	0
FC1	(25, 1)	25	151,250
FC2	(3, 1)	3	75

Table 5
Architecture of our custom CNN.

Parameter	Network
Learning Rate	1×10^{-4}
No. of epochs	10
Batch Size	250
Training Accuracy	92.54%
Validation Accuracy	91.34%
Testing Accuracy	89.38%

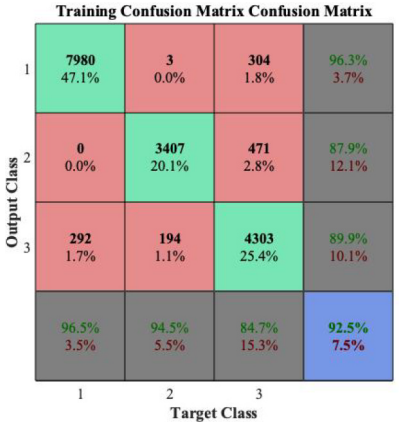


Fig. 30. Training Confusion Matrix.

shows the percentages of all the examples predicted to belong to each class that are correctly and incorrectly classified. These metrics are called the precision and false discovery rate, respectively. The last row of the matrix represents the percentages of all the examples belonging to each class that are correctly and incorrectly classified. The cell in the bottom right of the plot shows the overall accuracy. Fig. 32 represents the graphs of loss and accuracy vs number of epochs.

Closer inspection of Fig. 32 shows that all trained images are classified into their proper classes with an accuracy greater than ~90%. However, this level of accuracy may not be sufficient for some applications, where the presence of any data miss-classification has serious consequences. For example, the damage miss-classification in a nuclear reactor could cause a massive disaster. In order to improve the accuracy of the CNN model [32,33], the following points should be considered:

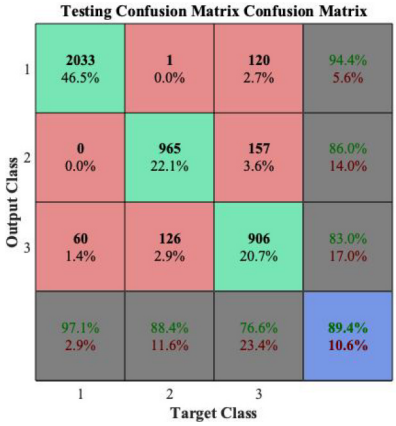


Fig. 31. Testing Confusion Matrix.

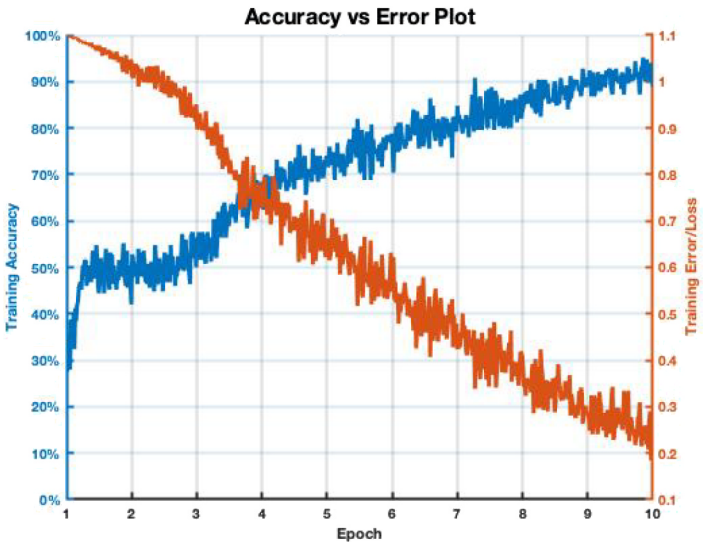


Fig. 32. Loss and accuracy vs number of epochs in the CNN-based damage classification.

1. Collect more useful data: mostly, deep learning algorithms have a better performance with more data.
2. Identify a suitable network structure: number of hidden layers and number of neurons can play a significant role in improving the performance of the model.
3. Tune the parameters: CNN model performance can be improved by tuning some parameters such as epochs. Number of epochs has a significant effect on the model performance, where for large number of epochs, the model performs better. However, deciding the number of epochs depends on the improvement in the training accuracy. Once the improvement in the training accuracy becomes almost stable, as shown in Fig. 32, the required number of epochs is achieved.

The results of this investigation provide important insights into damage measurements. Precisely, the damage tolerance. For instance, when the tested images are classified into low-risk class, the damaged component should be inspected and evaluated periodically. On the other hand, the damaged component must be fixed or replaced, when tested images are classified into high-risk class.

5. Summary, conclusions, and future work

Visual Inspection is one of the current state-of-the-art techniques for detection of fatigue crack damage in machinery components; however, visual inspection is known to be labor-intensive and is often error-prone for (quantitative) estimation of fatigue damage in structural materials. Therefore, analyzing the risk of fatigue damage still remains a challenging task in industry. To address this issue, the paper has proposed an autonomous method for detection and classification of fatigue crack damage and the associated risk assessment in machinery components that are often made of ductile materials (e.g., polycrystalline alloys). The underlying algorithms in the proposed method are built upon the concept of convolutional neural networks (CNNs), where the execution time is much less

than those for visual inspection, and the detection and classification process is expected to be significantly less error-prone than that in visual inspection.

The above algorithms have been experimentally validated on a computer-instrumented and computer-controlled *MTS* fatigue testing apparatus, which is equipped with an optical microscope for generation of image data sets. The size of the generated images is 2, 040 × 2, 040 pixel resolutions. Since the image data is enormous, every generated image is modified by using gray-scale conversion and pixel resolutions reduction. The modified image size becomes 150 × 150 pixel resolutions instead of 2, 040 × 2, 040 pixel resolutions. Thus, the processed image has been classified based on the status of the fatigued side-notch surface. The free-crack image belongs to the non-risky class, while the image that exhibits a crack is classified as belonging to a risky class. Furthermore, the risky class is divided into two sub-classes: low-risk class and high-risk class. The low-risk class represented all images that were generated the critical crack length whereas generated images after the critical crack length are considered to be a high-risk class. The CNN architecture consists of two major parts. The first part is used for feature extraction, while the second part is used for classification. The feature extractor in this paper has four layers, of which there are two convolution layers and two pooling layers. The classifier has one layer called a fully connected layer. The performance of the generated CNN model was excellent, where the accuracy of the CNN model is 92.54% for training, 91.34% for validation, and 89.38% for testing. Therefore, it is concluded that the CNN-based algorithms are capable of quantitative evaluation of damage risk in critical components of the operating machinery, where the optical microscope generates the images of fatigued surfaces, as needed for training, validation, and testing.

While there are many areas of theoretical and experimental research, which must be investigated before the proposed methodology can be implemented in real-life applications, the following topics of future research are suggested.

- Modification of the CNN-based algorithms for guaranteed robustness to noise and uncertainties in both process variables (e.g., random loading of machinery components) and measurements (e.g., noise-corrupted ultrasonic signals and disturbance-polluted microscope images).
- Usage of other machine learning tools (e.g., hidden Markov modelling and Gaussian process modelling [34]) as alternatives to neural networks.
- Enhancement of the CNN-based algorithms to accommodate other types of materials. An example is reinforced concrete in civil infrastructures, which is a brittle material.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors are thankful for generous support of this research by the Saudi Arabian Cultural Mission (SACM). The authors also acknowledge the benefits of technical discussions with Dr. Najah Ghalyan. This research has been supported in part by U.S. Air Force Office of Scientific Research (AFOSR) under Grant No. FA9550-15-1-0400. Any opinions, findings, and conclusions in this paper are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

References

[1] T.L. Anderson, T.L. Anderson, Fracture Mechanics: Fundamentals and Applications, CRC Press, Boca Raton, FL, USA, 2005.
[2] F.C. Campbell, Elements of Metallurgy and Engineering Alloys, ASM International, 2008.
[3] M. Saggar, A. Nasr, C. Bouraoui, Initiation life prediction method for defective materials, Advances in Materials, Mechanics and Manufacturing, Springer, 2020, pp. 17–25.
[4] D. Roylance, Introduction to fracture mechanics. URL http://web.mit.edu/course/3/3.11/www/modules/frac.pdf.
[5] P. Kumar, K. Prashant, Elements of Fracture Mechanics, Tata McGraw-Hill Education, 2009.
[6] C.G. Drury, J. Watson, Good practices in visual inspection, Human factors in aviation maintenance-phase nine, progress report, FAA/Human Factors in Aviation Maintenance.@ URL: http://hfskyway.faa.gov.
[7] S. Li, X. Zhao, Convolutional neural networks-based crack detection for real concrete surface, in: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2018, vol. 10598, International Society for Optics and Photonics, 2018, p. 105983V.
[8] R. Asok, R. Patankar, Fatigue crack growth under variable-amplitude loading: Part I-Model formulation in state-space setting, Appl. Math. Modell. (2001) 979–994.
[9] R. Asok, R. Patankar, Fatigue crack growth under variable-amplitude loading: Part II-Code development and model validation, Appl. Math. Modell. (2001) 995–1013.
[10] Y.-J. Cha, W. Choi, O. Büyüköztürk, Deep learning-based crack damage detection using convolutional neural networks, Comput.-Aided Civil Infrastruct. Eng. 32 (5) (2017) 361–378.
[11] H. Oliveira, P.L. Correia, Automatic road crack segmentation using entropy and image dynamic thresholding, in: 2009 17th European Signal Processing Conference, IEEE, 2009, pp. 622–626.
[12] N. Tanaka, K. Uematsu, A crack detection method in road surface images using morphology, MVA 98 (1998) 17–19.
[13] P. Subirats, J. Dumoulin, V. Legeay, D. Barba, Automation of pavement surface crack detection using the continuous wavelet transform, 2006 International Conference on Image Processing, IEEE, 2006, pp. 3037–3040.
[14] N.F. Ghalyan, I.F. Ghalyan, A. Ray, Modeling of microscope images for early detection of fatigue cracks in structural materials, Int. J. Adv. Manuf. Technol. 104 (9–12) (2019) 3899–3913.
[15] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al., Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[16] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing

Systems, 2012, pp. 1097–1105.
[17] P.Y. Simard, D. Steinkraus, J.C. Platt, et al., Best practices for convolutional neural networks applied to visual document analysis., in: Icdar, vol. 3, 2003.
[18] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[19] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
[20] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
[21] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[22] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.
[23] K. He, G. Gkioxari, P. Dollár, R.B. Girshick, Mask r-cnn. corr abs/1703.06870 (2017), arXiv preprint arXiv:1703.06870.
[24] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (1) (2019) 60.
[25] A. Ahmed, A. Ali, S. Elkatatny, A. Abdulraheem, New artificial neural networks model for predicting rate of penetration in deep shale formation, Sustainability 11 (22) (2019) 6527.
[26] E. Zahavi, V. Torbilo, S. Press, Fatigue Design: Life Expectancy of Machine Parts, CRC Press, Boca Raton, FL, USA, 1996.
[27] D. Scherer, A. Müller, S. Behnke, Evaluation of pooling operations in convolutional architectures for object recognition, International Conference on Artificial Neural Networks, Springer, 2010, pp. 92–101.
[28] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in, Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 807–814.
[29] Y.A. LeCun, L. Bottou, G.B. Orr, K.-R. Müller, Efficient backprop, Neural Networks: Tricks of the Trade, Springer, 2012, pp. 9–48.
[30] R. Danzl, F. Helmli, S. Scherer, Comparison of roughness measurements between a contact stylus instrument and an optical measurement device based on a colour focus sensor, Proc. of the Nanotechnology Conference, 2006, pp. 284–287.
[31] R. Danzl, F. Helmli, S. Scherer, Focus variation–a robust technology for high resolution optical 3d surface metrology, Strojniški vestnik-J. Mech. Eng. 57 (3) (2011) 245–256.
[32] A. Khan, A. Sohail, U. Zahoora, A.S. Qureshi, A survey of the recent architectures of deep convolutional neural networks, Artif. Intell. Rev. (2020) 1–62.
[33] J. Wang, L. Perez, et al., The effectiveness of data augmentation in image classification using deep learning, Convolut. Neural Networks Vis. Recognit. (2017) 11.
[34] K. Murphy, Machine Learning: A Probabilistic Perspective, The MIT Press, Cambridge, MA, USA, 2012.