A Computationally Efficient Crack Detection Approach Based on Deep Learning Assisted by Stockwell Transform and Linear Discriminant Analysis

Andy Nguyen[1], Canh Long Nguyen[1], Vahidreza Gharehbaghi[1], Ruveen Perera[1], Jason Brown[1], Yang Yu[2], Hashem Kalbkhani[3]

[1]*School of Engineering, University of Southern Queensland, Springfield, QLD 4300, Australia*

[2]*Centre for Infrastructure Engineering and Safety, University of New South Wales, Sydney, NSW 2052, Australia*

[3]*Department of Electrical Engineering, Urmia University of Technology, Iran*

Corresponding author: andy.nguyen@usq.edu.au (A. Nguyen)

Abstract

This paper presents SpeedyNet, a computationally efficient crack detection method. Rather than using a computationally demanding convolutional neural network (CNN), this approach made use of a simple neural network with a shallow architecture augmented by a 2D Stockwell transform for feature transformation and linear discriminant analysis for feature reduction. The approach was employed to classify images with minute cracks under three simulated noisy conditions. Using time-frequency image transformation, feature conditioning and a fast deep learning-based classifier, this method performed better in terms of speed, accuracy and robustness compared to other image classifiers. The performance of SpeedyNet was compared to that of two popular pre-trained CNN models, Xception and GoogleNet, and the results demonstrated that SpeedyNet was superior in both classification accuracy and computational speed. A synthetic efficiency index was then defined for further assessment. Compared to GoogleNet and the Xception models, SpeedyNet enhanced classification efficiency at least sevenfold. Furthermore, SpeedyNet's reliability was demonstrated by its robustness and stability when faced with network parameter and input image uncertainties including batch size, repeatability, data size and image dimensions.

Keywords: Crack Detection, Image Noise, Stockwell Transform, Linear Discriminant Analysis, SpeedyNet, Computational Efficiency

1 Introduction

Infrastructures are the primary assets of urban civilization and play numerous vital roles in modern civilization. Transportation corridors, housing projects, service facilities and power plants are all critical urban components that influence daily life and economic productivity. As a result, maintaining their functional capability and integrity is critical, propelling structural health monitoring (SHM) to prominence as a cutting-edge technical area. Numerous underlying problems, such as aging materials, concrete cracking, and steel corrosion, can impede structural functionality and degrade system performance. Timely and accurate condition assessment of these infrastructural systems is, therefore, of paramount importance.

Visual inspection performed by trained engineering technicians is one of the most common approaches to condition assessment. Another common approach is the use of non-destructive testing (NDT) techniques such as bulk wave ultrasonic testing, X-rays, thermal imaging, eddy current and guided waves to locate and measure damage or defects in structures (Abbas & Shafiee, 2018). A combination of experimental results and the engineer's opinion provide the groundwork for a pre-planned evaluation of the monitored structure and subsequent maintenance decisions. Despite their popularity, these traditional approaches are known to be time-consuming, laborious, increasingly expensive and, in many cases, dangerous (Gharehbaghi et al., 2021; Kim et al., 2019).

Integrating cutting-edge technologies such as artificial intelligence (AI) and modern image acquisition systems like remotely piloted aircrafts (aka drones) or laser scanners, offers an alternative solution to condition assessment. Unlike traditional visual inspection, this image-based assessment approach has the potential to monitor a large range of structures in an efficient, practical and error-free manner in harsh weather conditions. Image-based assessment is contactless, precise, long-range, immune to electromagnetic interference in multipoint measurements, and capable of measuring a wide range of parameters in large-scale civil infrastructures (Ye et al., 2016). In recent years, automated image-based identification of damage, such as cracks, has become very popular thanks to rapid advances in both software and hardware (Spencer Jr et al., 2019).

Automated image-based damage identification is carried out in different stages, including (i) damage detection via image classification, (ii) damage localization via image localization or object detection, and (iii) damage severity identification via image segmentation. Labelling images into different categories, finding specific regions (e.g. containing damage) in images, determining the vicinity or border of objects (i.e. damage) and separating different regions in images are the respective goals of these stages. Major advances in these areas have been fuelled by machine learning techniques such as artificial neural networks (ANNs) or, more recently, various deep neural networks (DNNs). The latter is part of a powerful subfield called deep learning.

For decades, ANNs have been a popular choice in data-based engineering and assessment (Jayasundara et al., 2020a; Jayasundara et al., 2020b; Liu et al., 2022). However, as computations advanced and graphical processing units (GPUs) became available, deep learning made its way into the world of engineering. Deep learning as a subfield of machine learning has accelerated AI computations in various tasks such as object detection (Wu et al., 2020; Zhao et al., 2019), pattern recognition (Busia et al., 2019; Yu et al., 2019), motion tracking (Chen et al., 2014; Jain et al., 2014), and semantic segmentation (Jain et al., 2014; Lateef & Ruichek, 2019, Wu & Liu, 2021). Deep learning models include convolutional neural networks (CNNs), recurrent neural networks (RNNs), long short-term memory (LSTM), deep belief networks (DBNs), and denoising auto encoders DAEs (Mosavi et al., 2019).

As one of the most popular deep learning architectures, CNNs play a key role in image classification as supervised learning models on variable and fixed-length data (Mohan & Subashini, 2019). These models are widely used for concrete crack detection (Dung, 2019; Vignesh et al., 2021), road and pavement crack classification (Li et al., 2020; Zhang et al., 2016), and many other areas such as medical image classification (Toaçar et al., 2020; Wegmayr et al., 2018; Rao et al., 2016). With many millions of learnable parameters to estimate, most CNNs are, however, well-known for being computationally expensive (Yamashita et al.,2018). This highlights a great need for computationally efficient deep learning algorithms for crack detection as well as other engineering applications.

In recent years, deep learning has been increasingly utilised for image-based SHM applications, however there are several issues demanding further research. First, most studies focus solely on the classification of images under ideal conditions, and are yet to consider the existence of noise (Bhowmick et al., 2020; Gharehbaghi et al., 2022; Panella et al., 2018). Second, our literature review shows that few research works compare the efficiency of different image classification approaches, and fewer still evaluate the impact of image noise in civil engineering contexts such as on the classification of concrete cracks (da Costa et al., 2016; Nguyen et al., 2022). This highlights the need for better noise-resistant deep learning approaches to crack detection, and more comprehensive evaluation and characterisation studies.

To address the above research gaps, this study developed a robust and computationally efficient method for classifying images containing concrete cracks under adverse imaging conditions. To this end, we examined the capabilities of image transformations and feature reduction algorithms in the image based SHM context. We then integrated the algorithms and a simple DNN to develop a novel crack detection method called SpeedyNet that is capable of accomplishing the following goals:

- Higher performance in concrete crack classification compared to conventional pre-trained deep learning models
- High robustness in adverse imaging conditions
- Robust to uncertainties of the model parameters
- Robust to the variations in the number of training images and image sizes
- Speedy enough to reduce the computational cost of image-based SHM.

The rest of the paper is organised as follows. The methodology section begins with a description of the image dataset. The three essential computing phases of SpeedyNet are then outlined. Beyond that, a brief discussion of the theories underpinning the suggested method is provided. In the results and discussion section, we compare the results of crack image classification against those obtained using two popular CNNs, which is followed by a comprehensive evaluation of the robustness and stability of SpeedyNet under various types of data and computing uncertainties. Finally, the conclusion section discusses the key research findings and broader implications.

2 Methodology

When dealing with challenging field conditions, the robustness and stability of image-based SHM techniques are crucial. The need for robust and computationally efficient crack detection methods are evidenced as discussed above, and SpeedyNet was ultimately developed for this purpose. To assess the robustness and efficiency of SpeedyNet, a dataset of cracked and uncracked photographs of concrete surfaces was created. Different digital filters were used to simulate environmental conditions. Following that, the proposed approach was sequenced through three key computing phases which are detailed in Subsections 2.2.1, 2.2.2 and 2.2.3, respectively. Next, two well-known CNN models were used as reference classification methods to evaluate the performance of SpeedyNet in different operational conditions. In the final step, the performance indices were established for use in this study.

2.1 Data Preparation

When images are taken, their quality may be compromised in a variety of ways. One common issue is the presence of salt and speckled pepper noise caused by incorrect digital ISO settings. Another common issue is the creation of motion blur noise caused by relative movement between image capture equipment and focal locations. To simulate these adverse conditions, salt and pepper noise (denoted as SP), motion blur noise (MB), and a combination of these two (Comb) were applied to the normal 3000 image dataset (Nrm) as seen in Figure 1. For SP and MB simulations, a noise density of 6% was applied, the motion length was set to 20 pixels, and the angle of motion was set to 11 degrees. While three of these datasets (i.e., Nrm, SP and MB) have been previously used (Nguyen et al., 2022), the fourth one (Comb) was purposed to create a new challenge to assess SpeedyNet. Regardless of the noise patterns applied, all images of these four datasets have the same size at 256×256 pixels. Readers can access these datasets via the online data library in the Kaggle website (Nguyen et al., 2021).

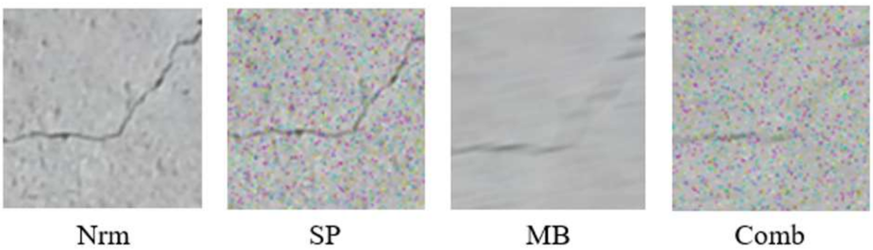


Figure 1. A Normal Image and Its Compromised Variants

2.2 SpeedyNet Architecture

This section explains the flow of the innovative crack identification solution under adverse conditions, as shown in Figure 2. Using a simplified deep neural network, the SpeedyNet combined the advantages of image modification and feature conditioning for crack identification. This novel combination is depicted below in three sequential phases.

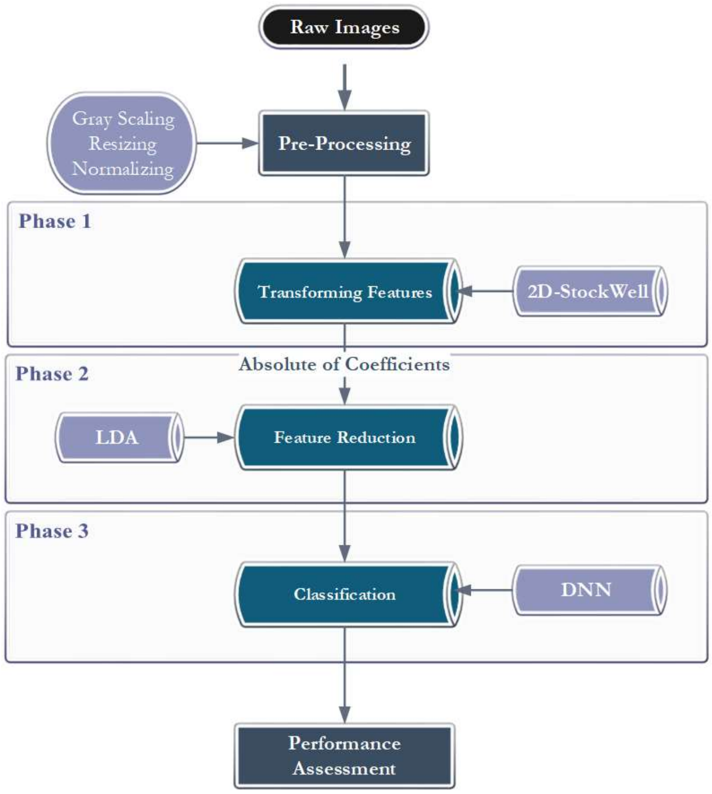


Figure 2. Flow Diagram of SpeedyNet Architecture

2.2.1 Phase 1: Transforming Features using 2D-Stockwell

In the first phase, the images were transformed into a new feature space. Several methods have been proposed and used for time-frequency analysis. Stockwell transform (ST), introduced in 1996 (Stockwell et al., 1996), shares similarities with short-time Fourier transform (STFT), and both techniques can reveal the time distribution of various frequency bands. Since STFT inherits a fixed window width, it cannot be implemented primarily for non-stationary signals (Kalbkhani & Shayesteh, 2017).

By contrast, ST benefits from a variable window as a frequency function, emulating the benefits of continuous wavelet transform (CWT) in terms of multi-resolution decomposition. Thus, higher frequency bands enable efficient processing via a fine-scaled window that yields fine resolution at high frequencies and coarse resolution at low frequencies (Kalbkhani & Shayesteh, 2017).

ST has been utilized for anomaly detection in different studies, however limited research has been undertaken for civil engineering applications like SHM. As a case in point, Singh and Shaik (2019) located and classified the bearing faults within an induction motor using ST and support vector machines (SVMs). Ditommaso et al. (2015) studied the capabilities of modal curvature evaluation and ST for the damage assessment of framed structures under seismic excitations. Sartipi et al. (2020) applied ST to obtain detailed information about the time-series of the region of interest

(ROI) in functional magnetic resonance imaging (fMRI). Time-frequency features partitioned into sub-matrices were combined with fuzzy entropies to diagnose a kind of disease from patients' recorded data.

To have a brief definition of this algorithm, the ST of a continuous signal $x(t)$ is calculated as (Sartipi et al., 2020):

$$S(T,f) = \frac{|f|}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} x(t)\exp(\frac{(T-t)^2}{2\sigma^2})\exp(^{-j2\pi ft})dt \\ = A(T,f)\exp(j\theta(T,f)) \;\&\; j=\sqrt{-1} \tag{1}$$

where t and T are time variables and $j=\sqrt{-1}$, f shows the frequency and $\sigma = \frac{1}{|f|}$ is the scale factor. The magnitude and phase of ST are represented by $A(T,f)$ and $\exp(j\theta(T,f)$ respectively. The output of ST is a complex-valued matrix in which the row and columns denote the time and frequency, respectively. ST has some advantages compared to other transformations such as (Stockwell, 2007):

- It can combine frequency-dependent resolution with an absolute reference phase
- It measures the local amplitude spectrum coupled with the phase spectrum, while WT provides the local amplitude/power spectrum
- Unlike most WT approaches, ST can be implemented on complex time series.

Concerning two-dimensional space, 2D-ST is applied to images. Let $f(x,y)$ represent an image, then the 2D-ST is formulated as (Soleimani et al., 2020):

$$S(u,v,f_u,f_v) = \frac{|f_u||f_v|}{2\pi} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\exp\left(\frac{(u-x)^2(v-y)^2}{2}\right)\exp(-j2\pi(f_u x + f_v y))dxdy \tag{2}$$

The Gaussian window's location on the x-axis and y-axis is adjusted by shifting parameters u and v, respectively. f_u and f_v are the corresponding frequencies regarding the scale parameters in each direction and control the window's expansion in each axis. $S(u,v,f_u,f_v)$ is a 4D matrix with complex values. For each point (u,v) in the image, the ST coefficients are a 2D matrix shown as $S_{u,v}(f_u,f_v)$.

In this study, the absolute values of the ST coefficients were chosen as the features of the concrete surface images. Figure 3 presents 2D-ST of non-cracked and cracked images for Nrm samples. As demonstrated, the feature map shows higher coefficient values for non-crack images compared to cracked ones.

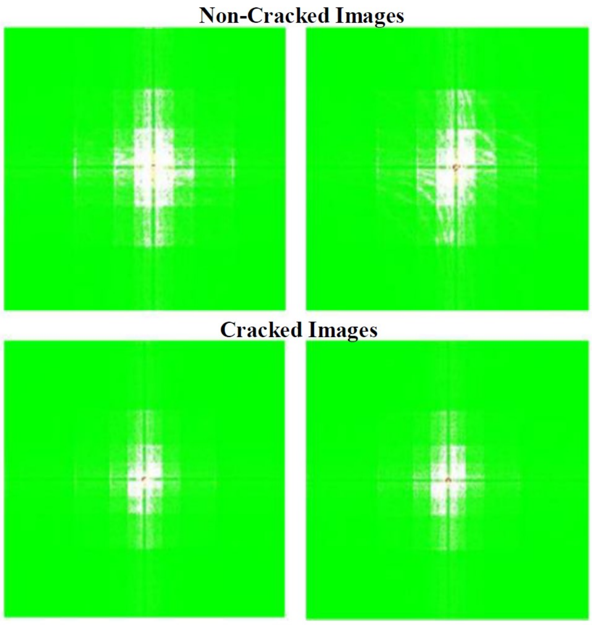


Figure 3. 2D-ST of Non-Cracked Images and Cracked Images

2.2.2 Phase 2: Feature Reduction using LDA

In the next phase, features were reduced as illustrated here. Reducing the number of features is crucial when dealing with machine learning and big data concerns. Although linear discriminant analysis (LDA) is mostly used to classify patterns into two groups, it can also be used to sort many patterns (Balakrishnama & Ganapathiraju, 1998). According to LDA, all groups are linearly separable, and to discriminate between them, several linear discrimination functions representing various hyperplanes in the feature space are constructed. On the condition that there are two groups, the LDA draws one hyperplane and projects the data onto this hyperplane to maximize the separation of the two classes (Balas et al., 2020). This hyperplane is built based on the two criteria considered concurrently:

- Maximizing the distance between the average of two groups
- Minimizing the variation between each class.

To cite a few studies on LDA for anomaly detection, Luo et al. (2021) substantiated the potential of spatial frequency domain imaging (SFDI) and LDA for detecting three types of damage in pear fruit. Yañez Borjas et al. (2019) employed a statistical feature extraction, LDA, and a neural network classifier to evaluate three levels of corrosion in a bridge. Gharehbaghi et al. (2022), employed kernel-based principal component analysis (PCA) and LDA to modify features of variational mode decomposition (VMD) and generalized autoregressive conditional heteroscedasticity (GARCH) for detecting linear and nonlinear damage in two building models.

In LDA, classes are expected to be normally distributed, and LDA also can be utilized for dimension reduction and classification. In a two-class dataset, a priori probabilities for Class 1 and Class 2 are $p1$ and $p2$, respectively; the class means and overall mean are µ1, µ2, and µ, respectively; and the class variances are $cov1$ and $cov2$, respectively.

$$\mu = p_1 \times \mu_1 + p_2 \times \mu_2 \tag{3}$$

Then, within-class and between-class scatter are used to depict the required criteria for class individually. The scatter measures for a multiclass situation are calculated as:

$$S_w = \sum_{j=1} p_j \times (\mathrm{cov}_j) \tag{4}$$

where C refers to the number of classes and:

$$cov_j = (x_j - \mu_j)(x_j - \mu_j)^T \tag{5}$$

Finally, the between-class scatter is calculated as:

$$S_b = \frac{1}{C} \sum_j (\mu_j - \mu) \times (\mu j - \mu)^T \tag{6}$$

The aim is to find a discriminant plane to maximize the between-class to within-class scatters (variances) ratio:

$$J_{LDA} = \frac{W S_b W^T}{W S_w W^T} \tag{7}$$

In this study, the absolute values of 2D-ST were presented in this new feature space provided by LDA to have maximum variances between features extracted from the input images.

The output of LDA was the reduced features that are different from the original features and obtained considering the original features and transformation matrix.

2.2.3 Phase 3: Classification using DNN

In the last phase, the reduced features were used as an input to a DNN-based classifier. One of the benefits of DNNs is their widespread use in image recognition and their exceptional accuracy when compared to regular ANNs. However, establishing dense architectural networks demands high-performance computing resources and a significant time effort. To avoid using a high-computational and low-speed classifier, the authors used a simple, fast deep learning classifier by reducing the number of features and increasing discrimination using LDA.

A simple DNN was then used to classify the images based on the LDA-modified input attributes. Classification was accomplished in this network using three fully connected layers (FC) connected to batch normalization layers for faster learning and improved performance, rectified linear unit (ReLU), and dropout layers to avoid overfitting and maximize generalization potential, as well as a softmax layer connected to the final FC. The architecture of this DNN is shown in Figure 4.

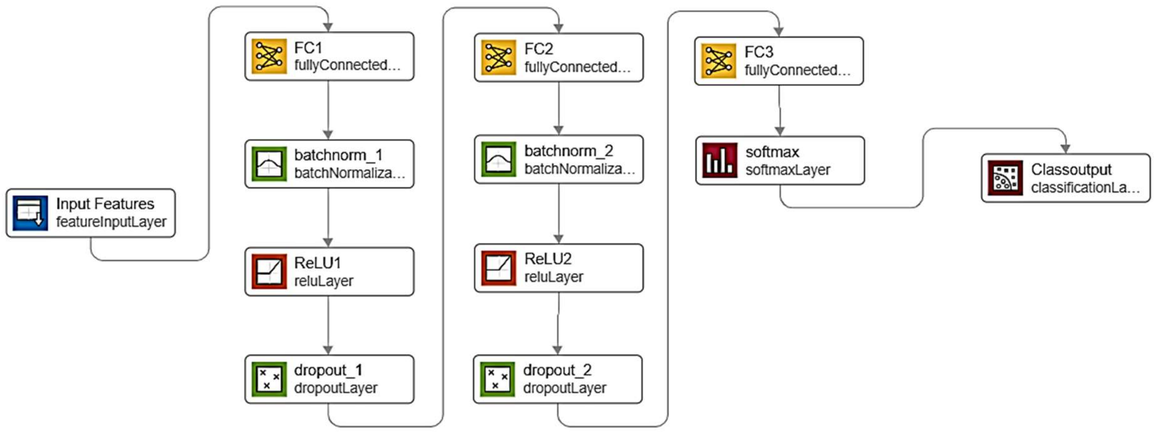


Figure 4. DNN Architecture Used in SpeedyNet

2.3 Reference Classification Methods

In AI, CNN is a feed-forward neural network known as a standard image recognition tool. A CNN extracts every part of an image, referred to as the receptive field and is made up of three forms of layers: (1) convolution, (2) pooling and (3) fully connected. CNNs can extract features from every part of an image based on the size of the receptive field. There are two ways of implementing a CNN: direct training from scratch and fine-tuning using pre-trained models (Nguyen et al., 2022). While the first approach requires a significant amount of data to train the network, the second is fast and straightforward and typical for engineering applications.

When dealing with pre-trained models, transfer learning is defined as the process of conveying knowledge that has already been learned by one neural network into another neural network (Olschofsky & Köhl, 2020). This process is often performed by imitating the learned weights and biases from layers of an entirely trained network to a second network. Transfer learning is capable of overcoming overfitting issues and accelerating the training procedure for a related role. Most of the pre-trained networks were previously trained with more than 1,000,000 images from the ImageNet database (Deng et al., 2009) to classify more than 1000 image classes. In the current study, the output classes were redefined as cracked and non-cracked.

In this research, to benchmark the performance of SpeedyNet, two pre-trained models with a relatively small and relatively large number of layers (i.e., GoogleNet and Xception, respectively) were selected as reference classification methods. These networks were considered among the best performing CNNs for crack detection when compared against the other six pre-trained models namely AlexNet, SqueezeNet, InceptionV3, ResNet-18, ResNet-50 and ResNet-101 (Nguyen et al., 2022). Their capacities for crack and damage detection have also been verified in other studies (Li & Zhao, 2019; Liu et al., 2020; Maeda et al., 2018). Table 1 provides a summary of key characteristics of these two models.

Table 1 Key Characteristics of Two Reference Classification Models

Model	No. Layers	Size on Hard Disk (MB)	No. of Original Parameters (Millions)	Image Input Size (Pixels)
GoogleNet	22	27	7.0	224 × 224
Xception	71	85	22.9	299 × 299

To avoid overfitting and performance degradation, deep learning optimization techniques were used. To optimize the network parameters and increase the network's overall performance in terms of overfitting, convergence, generalization and run time, we employed the stochastic gradient descent with momentum (SGDM), a non-adaptive learning rate technique. To avoid overfitting, L2 regularisation was used within the layers (Maeda-Gutierrez et al., 2020). The training iterations and epochs were set to 480 and 30, respectively. The associated values were obtained by trial and error to achieve the convergence of the training progress within the first few steps. The hyper-parameters denoted in Table 2 were utilized for a fair comparison between the pre-trained models utilized. Based on previous research (Nguyen et al., 2022), these parameters were determined by a grid search to achieve optimal performance with minimal overfitting.

Table 2 Hyper-parameters of Two Reference Classification Models

Parameter	Value
Optimization algorithm	SGDM
Initial learning rate	$1e^{-4}$
L2 Regularisation	$1e^{-3}$
Epochs	30
Total Iterations	480

2.4 Performance Indices

It is critical to evaluate the efficiency of machine learning predictions in classification challenges. Three of the most widely used metrics were used in this study to measure classification efficiency, as denoted by the following equations. Precision is known as the proportion of relevant findings, whereas recall is the percentage of total relevant results that have been correctly classified. Recall is also known as validation accuracy. F1-score combines precision and recall measures synthesized into a single and convenient measure for CNN performance comparisons (Saxena, 2018):

$$\boldsymbol{Precision} = \frac{True\ Positive}{Actual\ Results} \tag{8}$$

$$\boldsymbol{Recall} = \frac{True\ Positive}{Predicted\ Results} \tag{9}$$

$$\boldsymbol{F1-score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (10)$$

3 Results and Discussion

As detailed in this section, the classification performance of SpeedyNet was first evaluated and compared to that of the reference methods under different adverse conditions. Next, the computational efficiency was examined. After that, the function and impact of its components, including transformation and feature reduction, were evaluated. Finally, the robustness of SpeedyNet was assessed under various network architecture and input data uncertainties.

3.1 Comparing Performance

First, the performance of the SpeedyNet was assessed under different adverse conditions using the same numbers of epochs and total iterations (i.e., 30 and 480) as employed by GoogleNet and Xception. In this initial evaluation, the efficiency comparison was conducted in terms of classification performance and processing time. Both the SpeedyNet and two reference models were implemented using the deep learning toolbox in Matlab 2020b (*Mathworks, 2020*). To accelerate the computations when faced datasets and multiple runs, this study utilised an advanced computing machine from NVIDIA called DGX Station A100 for both training and inference (Nguyen et al., 2022).

Table 3 presents the typical performance scores of the three methods when working with the four datasets. As illustrated, SpeedyNet consistently outperformed the reference methods across all cases of datasets. Across the three performance indices, SpeedyNet achieved consistently high values in the range of 0.97–0.99, whereas those scores of GoogleNet and Xception fluctuated substantially in the ranges of 0.77–0.97 and 0.75–0.98, respectively.

Reviewing Table 3 more closely, one can clearly see both reference methods struggled when faced the Comb data with all performance indices scored less than 80%. Under the impact of the compounded noise, Xception performed no better than GoogleNet, if not to say that it was worse at Recall which was the reason for its slightly lower F1-score. Furthermore, the computation time indicates that GoogleNet was more than twice as fast as Xception while keeping similar classification performance scores with all other datasets. These results once again proved the effectiveness of GoogleNet as a relatively small CNN model for image-based crack detection (Nguyen et al., 2022).

On the other hand, SpeedyNet performed well in all cases, including when working with the Comb data. Additionally, the computing time of the method has been dramatically lowered when compared to reference models, taking only one sixth and one thirteenth of the computing time of GoogleNet and Xception, respectively. This outstanding performance of SpeedyNet can be attributed to not only the unique individual functionality of ST as well as LDA but also their novel partnership with DNN to provide a significant boost in performance for SpeedyNet. The impact of these two key components were further investigated in section 3.3.

Table 3 Performance Assessment of Methods

Method	Image Data	Recall	Precision	F1-score	Time (Seconds)
GoogleNet	Nrm	0.97	0.97	0.97	778.96
	SP	0.92	0.94	0.93	795.75
	MB	0.91	0.93	0.92	765.19
	Comb	0.78	0.77	0.77	809.75
Xception	Nrm	0.97	0.98	0.98	1687.51
	SP	0.91	0.94	0.93	1725.00
	MB	0.92	0.94	0.93	1677.22
	Comb	0.75	0.77	0.76	1686.45
SpeedyNet	Nrm	0.98	0.98	0.98	128.63
	SP	0.99	0.99	0.99	126.56
	MB	0.99	0.98	0.99	128.03
	Comb	0.98	0.97	0.98	129.23

3.2 Comparing Computational Efficiency

This assessment established a criterion to determine the most efficient method for classifying images under adverse conditions, by taking both performance score and computing cost into account. An efficiency index (EI) was therefore formulated as follows:

$$EI = \frac{Pm}{CT} \times 100 \tag{11}$$

Where Pm shows the average performance of each method based on the F1-score and CT stands for the computational time mentioned in Table 3. To provide a comprehensive overview of each technique, the performance and computing time presented in this index are the averages across normal and noisy situations. A higher EI value indicates improved performance with less computing time. As shown Figure 5, SpeedyNet was an outstanding winner on efficiency of image classification, achieving an EI of approximately 0.8, whereas Xception was the worst option in this regard, with an EI of less than 0.1. Comparatively speaking, SpeedyNet increased classification efficiency sevenfold when compared to GoogleNet, and fourteen times when compared to Xception.

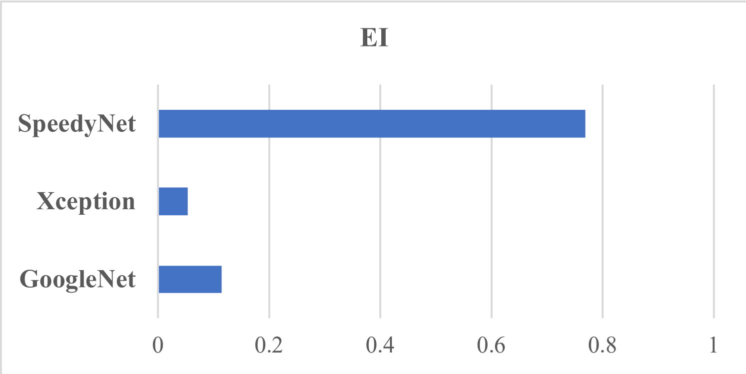


Figure 5. Comparing Methods Using EI

3.3 Impact of Model Components

This part analysed the effect of each component individually. To do so, the use of ST and LDA in combination as in SpeedyNet was substituted twice, one with LDA and another with ST, and the classification accuracy was evaluated using all datasets and averaged. As illustrated in Figure 6 (left), ST had the lowest calculation cost at 119 seconds. Compared to this, using LDA exclusively resulted in approximate 14% increase in computing time. Apparently, SpeedyNet has benefited predominantly from the speed of ST as reflected by the fact that the computing times of these two cases were nearly the same.

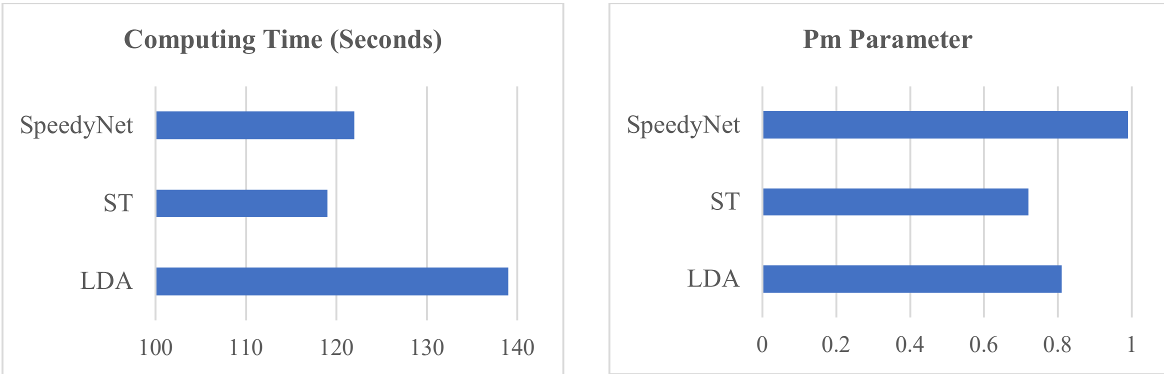


Figure 6. Impact of SpeedyNet Components in Computing Time (Left) and Classification Performance (Right)

Concerning the classification performance, Figure 6 (right) illustrates the Pm parameter for these three cases. Clearly, neither LDA nor ST had an exclusive impact on the performance of SpeedyNet. However, integrating LDA into ST increased efficiency from 72% in ST to around 98% with SpeedyNet. The examination of SpeedyNet's core parts has demonstrated the positive impact of feature transformation and reduction on classification performance and computational costs.

3.4 Robustness of SpeedyNet

To evaluate SpeedyNet's performance and accuracy when the network parameters and input image are uncertain, this part addressed three most common concerns of deep learning users i.e., the batch size of the deep learning model, the image size, and number of training images.

3.4.1 Impact of Network Parameters

To begin with, SpeedyNet was evaluated for efficacy by adjusting the batch size of the deep network and running it repeatedly. As a result, we investigated two batch sizes of 16 and 32, each with ten runs for each image dataset. Figure 7 illustrates the results for a variety of batch sizes and four datasets. In each subplot, the different colours represent those ten validation curves, with the ideal repeatability occurring when each validation curve closely coincides with their counterparts. As demonstrated, SpeedyNet achieved the desired accuracy in most cases across the two batch sizes proving the excellent stability of the method. Nonetheless, the smaller batch size (16) showed a better result with the MB data.

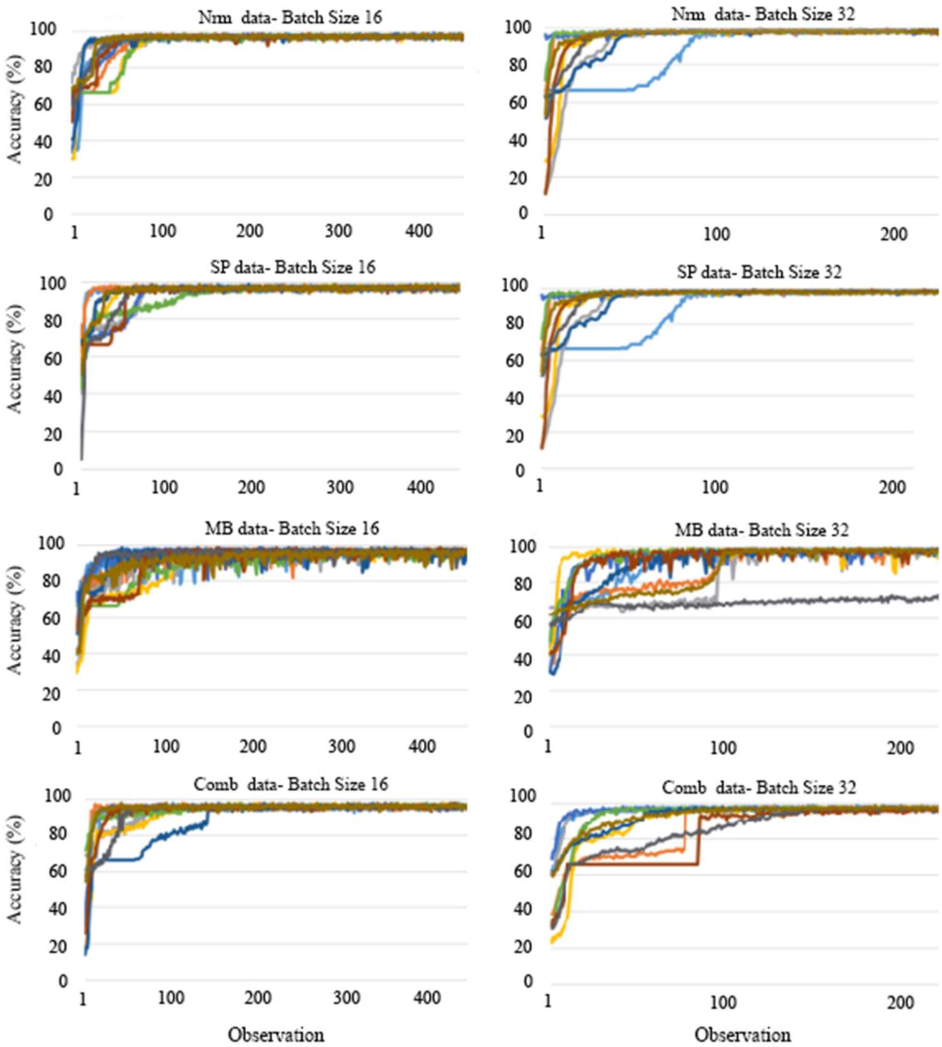
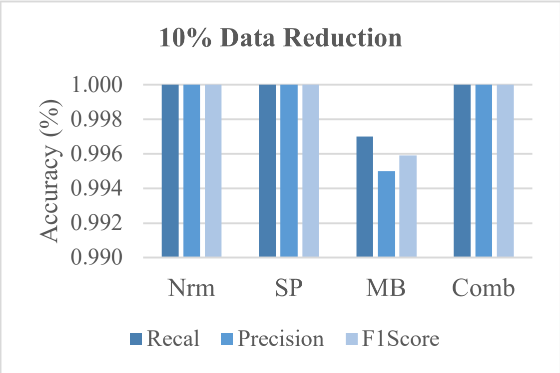


Figure 7. Stability of SpeedyNet versus Batch Sizes

3.4.2 Impact of Number of Training Images

This part investigated the impact of input data using different image sizes. To accomplish this, the number of images was reduced by 10% in one case and 20% in another and these reductions were applied across 4 datasets (Nrm, SP, MB and Comb). As illustrated in Figure 8, SpeedyNet performed extremely well in most cases. The most noticeable changes were with the MB data, but this reduction in performance was less than 1%.



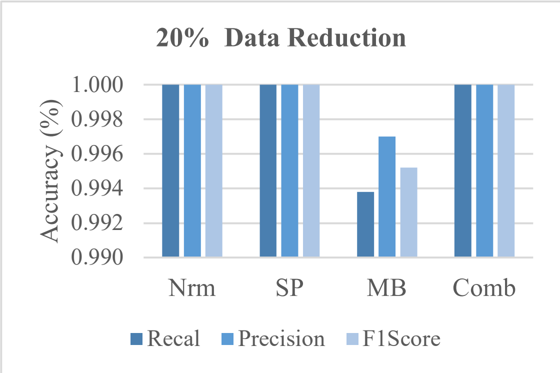
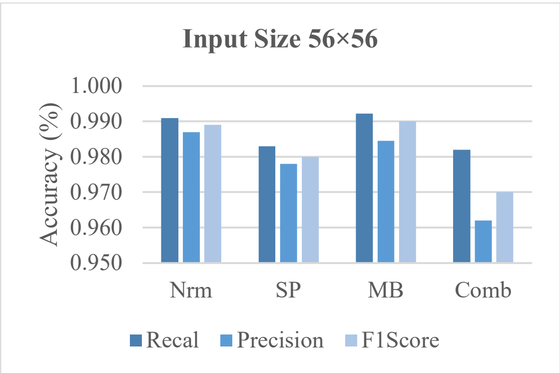


Figure 8. Impact of Training Data Reduction on SpeedyNet: 10% (Left) and 20% (Right)

3.4.3 Impact of Image Size

In this part, the effect of input image size on the performance of SpeedyNet was investigated. With some similarity to the previous section, two types of new image sets were defined by reducing the input image dimension from 256×256 pixels to 56×56 pixels and 48×48 pixels. Then the performances of SpeedyNet were re-evaluated. The results for the two new image datasets are displayed in Figure 9. As shown, the reduction in performance was less than 5% in all cases, including with the Comb data which was once again proven the most difficult dataset.



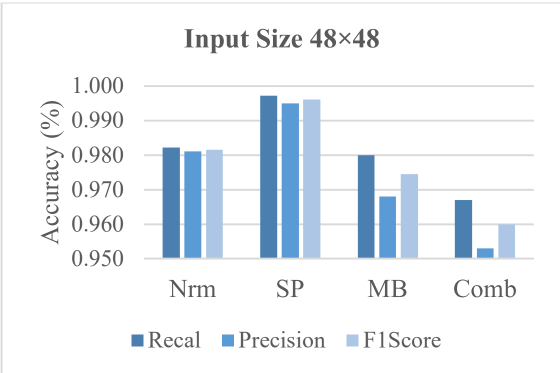


Figure 9. Impact of Input Image Size on SpeedyNet: 56×56 (Left) and 48×48 (Right)

4 Conclusion

The primary goal of this study was to develop a robust and computationally efficient crack detection approach for classifying images of concrete cracks taken in various adverse conditions. To simulate these conditions, three types of image filters were used, i.e., salt and pepper (SP), motion blur (MB), and a combination of these two (Comb) along a concrete crack dataset with minute cracks to create a total of four challenging datasets. Taking on these challenges, we successfully developed a novel deep learning-based image classification method and named it SpeedyNet to reflect the most desirable feature of the algorithm, that is, its speediness.

SpeedyNet encompasses three computing phases, the Stockwell image transformation, the LDA feature conditioner, and a simple 12-layer DNN for classification. To evaluate the effectiveness of SpeedyNet, two of the most common CNN models, Xception and GoogleNet, were employed as reference models working on the same four datasets. The fourth dataset (Comb) had become the biggest challenge for the two CNN models with the result showing approximate 20% drop across all performance indices of both models. On the other hand, SpeedyNet championed in all four cases, including when working with the Comb data, showing its consistent top-performance, scoring between 97% to 99%, while taking only one sixth of computational time of the runner-up, that is, GoogleNet. A combined index called EI showed that SpeedyNet had remarkably increased the classification efficiency sevenfold and when compared to GoogleNet, and fourteen times when compared to Xception.

To characterize the algorithm, the impact of each triple component of SpeedyNet was applied separately to the image data. Results demonstrated the following points:

- Using ST alone may marginally reduce computational time, but this comes at the significant expense of accuracy (over a quarter).
- Using LDA alone is not recommended as this causes a 15% jump in the computational time compared the use of SpeedyNet.
- When compared to the single use of ST or LDA, their combination in SpeedyNet improved accuracy by 27% and 18%, respectively. This novel arrangement together with the use of simple but effective DNN are among the most significant contributions of this research.

Eventually, the proposed method was investigated further against the three common uncertainties in the deep learning network and image dataset. Accordingly, the batch size, the number of input images and the image size were varied during the evaluation. It was concluded that:

- The developed approach was robust to the change across two most common batch sizes (16 and 32) with similar performance scores achieved for most datasets. However, the smaller batch size (16) was recommended due to its higher robustness against the MB data.
- SpeedyNet demonstrated exceptional stability when working with varying numbers of input images and image dimensions, with performance reductions of less than 5% in most circumstances.

All these results have proven the consistent robustness and versatility of SpeedyNet. Future works may focus on the utilization of similar image transformations such as contourlets and wavelets. Other kernel-based feature conditioning and optimization method applications for tuning the network parameters can be investigated in the future, along with crack segmentation.

5 References

Abbas, M., & Shafiee, M. (2018). Structural health monitoring (SHM) and determination of surface defects in large metallic structures using ultrasonic guided waves. Sensors, 18(11), 3958.

Balas, V. E., Solanki, V. K., & Kumar, R. (2020). An Industrial IoT Approach for Pharmaceutical Industry Growth: Volume 2. Academic Press.

Balakrishnama, S., & Ganapathiraju, A. (1998). Linear discriminant analysis-a brief tutorial. Institute for Signal and information Processing, 18(1998), 1-8.

Bhowmick, S., Nagarajaiah, S., & Veeraraghavan, A. (2020). Vision and Deep Learning-Based Algorithms to Detect and Quantify Cracks on Concrete Surfaces from UAV Videos. Sensors, 20(21), 6299.

Bridge Inspector Killed When Truck Tips on I-84 Ramp in West Hartford. (2015). www.nbcconnecticut.com

Busia, A., Dahl, G. E., Fannjiang, C., Alexander, D. H., Dorfman, E., Poplin, R., McLean, C. Y., Chang, P.-C., & DePristo, M. (2019). A deep learning approach to pattern recognition for short DNA sequences. BioRxiv, 353474.

Chen, J. G., Wadhwa, N., Cha, Y.-J., Durand, F., Freeman, W. T., & Buyukozturk, O. (2014). Structural modal identification through high speed camera video: Motion magnification. In Topics in Modal Analysis I, Volume 7 (pp. 191-197). Springer.

da Costa, G. B. P., Contato, W. A., Nazare, T. S., Neto, J. E., & Ponti, M. (2016). An empirical study on the effects of different types of noise in image classification tasks. arXiv preprint arXiv:1609.02781.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. 2009 IEEE conference on computer vision and pattern recognition.

Ditommaso, R., Ponzo, F., & Auletta, G. (2015). Damage detection on framed structures: modal curvature evaluation using Stockwell Transform under seismic excitation. Earthquake Engineering and Engineering Vibration, 14(2), 265-274.

Dung, C. V. (2019). Autonomous concrete crack detection using deep fully convolutional neural network. Automation in Construction, 99, 52-58.

Fujita, Y., & Hamamoto, Y. (2011). A robust automatic crack detection method from noisy concrete surfaces. Machine Vision and Applications, 22(2), 245-254.

Gharehbaghi, V. R., Kalbkhani, H., Noroozinejad Farsangi, E., Yang, T., Nguyen, A., Mirjalili, S., & Málaga-Chuquitaype, C. (2022). A novel approach for deterioration and damage identification in building structures based on Stockwell-Transform and deep convolutional neural network. Journal of Structural Integrity and Maintenance, 7(2), 136-150.

Gharehbaghi, V. R., Noroozinejad Farsangi, E., Noori, M., Yang, T., Li, S., Nguyen, A., Málaga-Chuquitaype, C., Gardoni, P., & Mirjalili, S. (2021). A Critical Review on Structural Health Monitoring: Definitions, Methods, and Perspectives. Archives of Computational Methods in Engineering, 1-27.

Ghose, S., Singh, N., & Singh, P. (2020). Image denoising using deep learning: convolutional neural network. 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence).

Han, X., Zhong, Y., Cao, L., & Zhang, L. (2017). Pre-trained alexnet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification. Remote Sensing, 9(8), 848.

Jain, A., Tompson, J., LeCun, Y., & Bregler, C. (2014). Modeep: A deep learning framework using motion features for human pose estimation. Asian conference on computer vision.

Jayasundara, N., Thambiratnam, D. P., Chan, T. H. T., & Nguyen, A. (2020). Locating and quantifying damage in deck type arch bridges using frequency response functions and artificial neural networks. International Journal of Structural Stability and Dynamics, 20(10), 2042010.

Jayasundara, N., Thambiratnam, D. P., Chan, T. H. T., & Nguyen, A. (2020). Damage detection and quantification in deck type arch bridges using vibration based methods and artificial neural networks. Engineering Failure Analysis, 109, 104265.

Kalbkhani, H., & Shayesteh, M. G. (2017). Stockwell transform for epileptic seizure detection from EEG signals. Biomedical Signal Processing and Control, 38, 108-118.

Kim, H., Ahn, E., Shin, M., & Sim, S. H. (2019). Crack and noncrack classification from concrete surface images using machine learning. Structural Health Monitoring, 18(3), 725-738.

Lagendijk, R. L., & Biemond, J. (2009). Basic methods for image restoration and identification. In The essential guide to image processing (pp. 323-348). Elsevier.

Lateef, F., & Ruichek, Y. (2019). Survey on semantic segmentation using deep learning techniques. Neurocomputing, 338, 321-348.

Li, B., Wang, K. C., Zhang, A., Yang, E., & Wang, G. (2020). Automatic classification of pavement crack using deep convolutional neural network. International Journal of Pavement Engineering, 21(4), 457-463.

Li, S., & Zhao, X. (2019). Image-based concrete crack detection using convolutional neural network and exhaustive search technique. Advances in Civil Engineering, 2019.

Liu, Y., Hou, M., Li, A., Dong, Y., Xie, L., & Ji, Y. (2020). Automatic Detection of Timber-Cracks in Wooden Architectural Heritage Using YOLOv3 Algorithm. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 43, 1471-1476.

Liu, Y., Wang, L., Gu, K., & Li, M. (2022). Artificial Neural Network (ANN)-Bayesian Probability Framework (BPF) based method of dynamic force reconstruction under multi-source uncertainties. Knowledge-Based Systems, 237, 107796.

Luo, Y., Jiang, X., & Fu, X. (2021). Spatial Frequency Domain Imaging System Calibration, Correction and Application for Pear Surface Damage Detection. Foods, 10(9), 2151.

Maeda-Gutierrez, V., Galvan-Tejada, C. E., Zanella-Calzada, L. A., Celaya-Padilla, J. M., Galván-Tejada, J. I., Gamboa-Rosales, H., Luna-Garcia, H., Magallanes-Quintanar, R., Guerrero Mendez, C. A., & Olvera-Olvera, C. A. (2020). Comparison of convolutional neural network architectures for classification of tomato plant diseases. Applied Sciences, 10(4), 1245.

Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., & Omata, H. (2018). Road damage detection using deep neural networks with images captured through a smartphone. arXiv preprint arXiv:1801.09454.

MATLAB (2020). https://www.mathworks.com/help/matlab/

Mohan, G., & Subashini, M. M. (2019). Medical imaging with intelligent systems: a review. Deep learning and parallel computing environment for bioengineering systems, 53-73.

Mosavi, A., Ardabili, S., & Varkonyi-Koczy, A. R. (2019). List of deep learning models. International Conference on Global Research and Education.

Nguyen, A., Chianese, R. R., & Gharehbaghi, V. R. (2021). 3000 ImageData for crack detection. https://doi.org/10.34740/KAGGLE/DSV/2497827

Nguyen, A., Chianese, R.R., Gharehbaghi, V.R., et al. (2022) Robustness of Deep Transfer Learning-Based Crack Detection Against Uncertainty in Hyperparameter Tuning and Input Data. In Guan, H. Chan, T. & Li, J. (Eds.) Recent Advances in Structural Health Monitoring Research in Australia. Nova Science Publishers, New York. https://doi.org/10.52305/QHVI3457

Olschofsky, K., & Köhl, M. (2020). Rapid field identification of cites timber species by deep learning. Trees, Forests and People, 2, 100016.

Panella, F., Boehm, J., Loo, Y., Kaushik, A., & Gonzalez, D. (2018). Deep learning and image processing for automated crack detection and defect measurement in underground structures. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences-ISPRS Archives.

Rao, P., Pereira, N. A., & Srinivasan, R. (2016). Convolutional neural networks for lung cancer screening in computed tomography (CT) scans. 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I).

Sarmadi, H., & Entezami, A. (2021). Application of supervised learning to validation of damage detection. Archive of Applied Mechanics, 91(1), 393-410.

Sartipi, S., Kalbkhani, H., Ghasemzadeh, P., & Shayesteh, M. G. (2020). Stockwell transform of time-series of fMRI data for diagnoses of attention deficit hyperactive disorder. Applied Soft Computing, 86, 105905.

Saxena, S. (2018). Precision vs Recall. Towards Data Science. Retrieved 21 July 2020 from www.towardsdatascience.com

Sharma, A., Kumar, P., Vinayak, H. K., & Walia, S. K. (2021). Condition Evaluation in Steel Truss Bridge with Fused Hilbert Transform, Spectral Kurtosis, and Bandpass Filter. Structural Durability & Health

Sieberth, T., Wackrow, R., & Chandler, J. (2014). INFLUENCE OF BLUR ON FEATURE MATCHING AND A GEOMETRIC APPROACH FOR PHOTOGRAMMETRIC DEBLURRING. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences.

Singh, M., & Shaik, A. G. (2019). Faulty bearing detection, classification and location in a three-phase induction motor based on Stockwell transform and support vector machine. Measurement, 131, 524-533.

Soleimani, M., Vahidi, A., & Vaseghi, B. (2020). Two-Dimensional Stockwell Transform and Deep Convolutional Neural Network for Multi-Class Diagnosis of Pathological Brain. IEEE Transactions on Neural Systems and Rehabilitation Engineering.

Spencer Jr, B. F., Hoskere, V., & Narazaki, Y. (2019). Advances in computer vision-based civil infrastructure inspection and monitoring. Engineering, 5(2), 199-222.

Stockwell, R. G., Mansinha, L., & Lowe, R. P. (1996). Localization of the complex spectrum: the S transform. IEEE transactions on signal processing, 44(4), 998-1001.

Stockwell, R. G. (2007). A basis for efficient representation of the S-transform. Digital Signal Processing, 17(1), 371-393.

Toğaçar, M., Cömert, Z., & Ergen, B. (2020). Classification of brain MRI using hyper column technique with convolutional neural network and feature selection method. Expert Systems with Applications, 149, 113274.

Verma, R., & Ali, J. (2013). A comparative study of various types of image noise and efficient noise removal techniques. International Journal of advanced research in computer science and software engineering, 3(10).

Vignesh, R., Narenthiran, B., Manivannan, S., Murugan, R. A., & RajKumar, V. (2021). Concrete Bridge Crack Detection Using Convolutional Neural Network. In Materials, Design, and Manufacturing for Sustainable Environment (pp. 797-812). Springer.

Wegmayr, V., Aitharaju, S., & Buhmann, J. (2018). Classification of brain MRI with big data and deep 3D convolutional neural networks. Medical Imaging 2018: Computer-Aided Diagnosis.

Wu, X., Sahoo, D., & Hoi, S. C. (2020). Recent advances in deep learning for object detection. Neurocomputing, 396, 39-64. https://doi.org/10.52305/QHVI3457

Wu, X., & Liu, X. (2021). Building crack identification and total quality management method based on deep learning. Pattern Recognition Letters, 145, 225-231.

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. Insights into imaging, 9(4), 611-629.

Yanez-Borjas, J. J., Camarena-Martinez, D., Valtierra-Rodriguez, M., SaucedoDorantes, J. J., & Amezquita-Sanchez, J. P. (2019). Methodology based on statistical features and linear discriminant analysis for damage detection in a truss-type bridge. 2019 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC).

Ye, X.-W., Dong, C., & Liu, T. (2016). A review of machine vision-based structural health monitoring: methodologies and applications. Journal of Sensors, 2016.

Yu, J., Zheng, X., & Wang, S. (2019). A deep autoencoder feature learning method for process pattern recognition. Journal of Process Control, 79, 1-15.

Zhang, L., Yang, F., Zhang, Y. D., & Zhu, Y. J. (2016). Road crack detection using deep convolutional neural network. 2016 IEEE international conference on image processing (ICIP).

Zhang, W., Jin, L., Song, E., & Xu, X. (2019). Removal of impulse noise in color images based on convolutional neural network. Applied Soft Computing, 82, 105558.

Zhao, Z.-Q., Zheng, P., Xu, S.-t., & Wu, X. (2019). Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11), 3212-3232.