Journal of Airline Operations and Aviation Management
The Netherlands Press

Article

DETECTION OF STRUCTURAL CRACKS OF AN AIRCRAFT USING DEEP NEURAL NETWORKS

NETKACHEV Aleksandr G.

Phd in robots, mechatronics and robototechnics system, Department of Automation, Peter the Great St.Petersburg Polytechnic University (SPbPU),
Saint-Petersburg, Russian Federation.

Orchid Id: https://orcid.org/0000-0003-2710-8743

E-mail: netkachev.alexamdrg@yandex.ru

Abstract.

Thanks to developments in machine learning (ML), particularly deep learning, which now has the greatest performance among algorithms, narrow artificial intelligence, often known as "weak AI," has progressed during the past few years. further machine learning for deep learning to provide potent model parameters that can forecast the occurrence of certain events in the future, massive volumes of data, also referred to as "big data" and must be rapidly collected. Such a big damage event dataset is not accessible in many other fields, such as visual inspection of aeroplanes, and this makes it difficult to train deep learning algorithms to work effectively. Good at spotting physical damage to aircraft structures. In order to reach this human-level intelligence in aircraft damage inspection, it is possible to include inductive bias into deep learning. This paper provides an illustration of how to incorporate expertise in aircraft engineering into the creation of deep learning algorithms. The effectiveness of our method, which builds a deep convolutional neural network that categorises crack lengths based on break propagation curves acquired from fatigue testing, was shown on aerospace grade aluminium samples.

Keywords: Machine Learning, Artificial Intelligence, fatigue, Prediction, Accuracy, crack length.

Journal of Airline Operations and Aviation Management Volume 1 Issue 1
Received Date: 13 May 2022
Accepted Date: 12 June 2022
Published Date: 25 July 2022

1. Introduction

Airlines have implemented routine inspections and part replacement procedures, which have helped to partially tackle the issue of ageing aircraft. This approach avoids having to purchase a new plane, which is the best but most expensive choice, but it merely delays the issue. The parts replacement system will probably become overburdened as the size of aged fleets increases due to the high volume of replacement components needed. Airlines must have a monitoring system in place to alert maintenance staff when a part needs to be changed because of wear and tear rather than service life in order to save time and money [1].

The ropes assist transmit the skin load to the interior structure and often handle a greater aircraft load. Almost typically, the ropes are fastened to the aircraft's structure, sides, or skin. The pins used to affix the fasteners to the framework gradually develop fatigue and stress fractures. Due to their tiny size (about 1.25 mm or 0.05 in.), proximity to and location below the fastener's top, and frequent existence under the surface coating, these fissures are likely to go undiscovered. Additionally eroded and needing inspection is the contact area between the pile and the skin[2].

These circumstances create intriguing technological issues. First, it is challenging to find any possible fractures that may be present close to the fastener since they are small, stretch in all directions, and are frequently buried. Second, it might be challenging to detect latent interlayer corrosion in laminated aluminium structures. In such constructions, the necessary corrosion detection threshold typically equals 10% material loss. The system must be able to identify these utterly distinct sorts of problems with only a single probe and test equipment in order to be genuinely practical.

Every machine learning model needs some sort of architectural design, as well as perhaps some initial presumptions regarding the data to be studied. Every foundational idea and conviction that data production involves inductive bias is generally true. Machine learning models' capacity to generalise to new data is significantly impacted by inductive biases. Our model could go in the direction of global optimality if there is a significant inductive bias. On the other hand, weak inductive bias is heavily impacted by random changes in the starting state and might lead to the model finding only local optima[3].

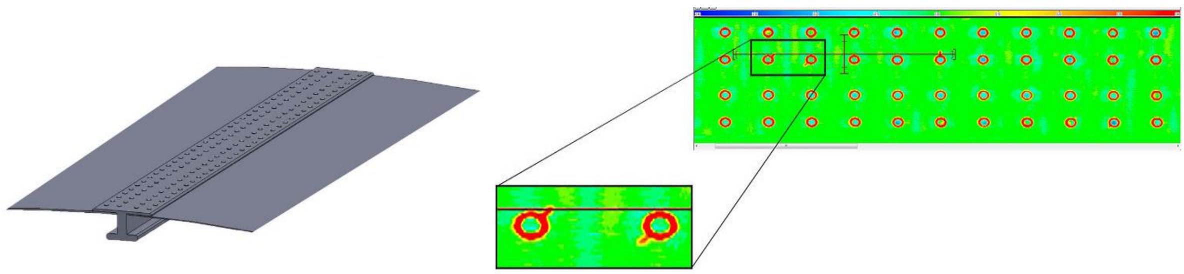

Figure 1: longeron and cracks on longeron[4]

2. METHODOLOGY:

A multilayer perceptron is regularised via CNN. Every neuron in one layer is linked to every neuron in the next layer, which is what is meant by a multilayer perceptron in most cases. As a result of their "complete connectedness," these networks are susceptible to data overfitting. Typically, training factors (such weight loss) are penalised, or connectivity is decreased to control or avoid overfitting (dropping connections, dropping out, etc.) [5]. CNNs have a different method to regularisation: they use smaller, simpler samples that are embossed in their filter and take use of the hierarchical model of the data and an increasingly complicated collection of patterns. surname. CNNs are therefore quite low on the connectivity and complexity scale [6].

Biological processes provide as inspiration for constitutional networks. It has a network of connections between neurons that resembles the way the visual cortex of animals is set up. Only a small portion of the visual field, known as the receptive field, is used by individual cortical neurons to react to inputs [7]. Different neurons' receptive areas partially overlap one another to fill the whole visual field [8].

Comparatively speaking to other image classification techniques, CNNs employ a little amount of pre-processing. In contrast to traditional techniques, which construct these filters by hand, machine learning enables the network to learn how to optimise filters (or cores) [9,10]. This feature extraction's independence from prior information and human interaction is a significant benefit.

In civil infrastructure including buildings, walkways, and concrete surfaces, CNN has been used to identify cracks[11,12]. These results are encouraging and a first step in improving picture identification for CNN crack detection, but more generalisation of domain knowledge matching is required to provide logical explanations for the deep learning outcomes that can be explained[13].

With regard to this, Deep CNN has developed and tested a fracture length categorization system based on the results of fatigue testing's crack propagation curves. The network is therefore based on the physics of crack propagation behaviour, in contrast to picture segmentation and thresholding[14]. The classification results of CNN architecture crack images with domain knowledge are compared with the classification results of images of the same networks are not included in the domain, but pre-clustered by unsupervised machine learning, in order to assess the performance of domain knowledge inclusion.

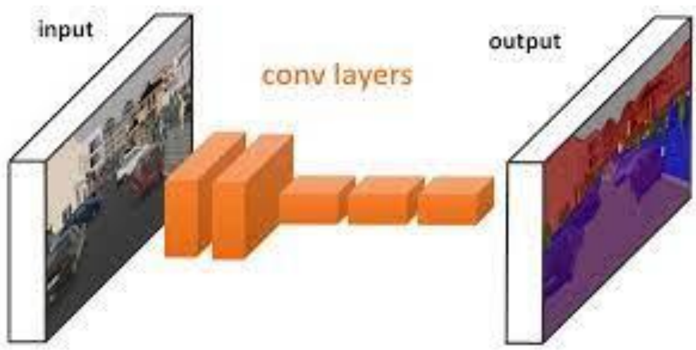

Figure 2: Convolutional neural networks[15]

3. Dataset preparation and Pre-processing:

The dataset of aircraft crack photos is taken, and each image is divided into three classes: big, medium, and tiny. The last class is None. Images of aeroplanes with no cracks are in the None class. Based on the crack lengths found during the fatigue test, these cracks are categorised. Each class contains photos with crack lengths and the results of each fatigue test. There are 11803 total images in the data set. The crack lengths were divided into four groups by testing those photos. Ninety-two percent (92%) of the 11803 RGB photos were used for training, sixteen percent (9%) for validation, and six percent (652) for testing the trained model (6 percent). The RGB pictures were initially pre-processed by turning them into grayscale images (i.e., pixel-averaged value) in order to decrease the computational load before feeding the CNN with training data.

Figure 1: large,medium, and small cracks

3.1. Data Pre-processing:

The RGB photos are turned into grayscale images in the data pre-processing stage so that the pixel values may be reduced to 0 and 1. Typically, pixels in RGB photographs fall between 0 and 255. When the input values fall between the range of 0 and 1, neural networks operate well. Thus, a grayscale image is created by converting an RGB image to pixels. Additionally, scaling work is done at the pre-processing stage. The basic goal of scaling the photos is to uniformly scale all of them in order to reduce network size mistake.

3.2. Building a CNN Model:

The input shape must be established before convolutional neural networks can be constructed. The network's input shape is the size of the scaled picture. The network is initially given a convolutional layer with a filter size of 32 and a window size of 3x3, and ReLU is utilised as the activation function. The network is then given a pooling layer with a window size of 2x2. The pooling layer shrinks the pixel size to its smallest possible size. Following that, a block of layers of the same size is added, followed by the addition of two layers with the same characteristics but a size of 64. Using the model, the final pixel values are flattened, and a dense layer with 16 neurons is then added to the network. ReLU performs the activation function in the dense layer. Due to overfitting concerns, a hidden layer of the same size is added once again, and then after verifying that there is no dropout function with neurons that are 50% active, a dropout layer is added. The last output layer has four neurons, each of which represents one of the four classes—Large, Medium, Small, and None. The SoftMax function's activation is present in the final output layer.

```
model = Sequential()

model.add(Conv2D(32, (3, 3), input_shape=input_shape,activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(32, (3, 3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(64, (3, 3)),activation='relu')
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(64, (3, 3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(16))
model.add(Activation('relu'))
model.add(Dense(16))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(4))
model.add(Activation('softmax'))
```

The model summary includes the parameters for each layer, which affects how big the image is. The output class, which consists of four classes, is then included in the outcome.

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_8 (Conv2D)           (None, 98, 98, 32)        896

 max_pooling2d_7 (MaxPooling  (None, 49, 49, 32)       0
 2D)

 conv2d_9 (Conv2D)           (None, 47, 47, 32)        9248

 max_pooling2d_8 (MaxPooling  (None, 23, 23, 32)       0
 2D)

 conv2d_10 (Conv2D)          (None, 21, 21, 64)        18496

 max_pooling2d_9 (MaxPooling  (None, 10, 10, 64)       0
 2D)

 conv2d_11 (Conv2D)          (None, 8, 8, 64)          36928

 max_pooling2d_10 (MaxPoolin  (None, 4, 4, 64)         0
 g2D)

 flatten_1 (Flatten)         (None, 1024)              0

 dense_5 (Dense)             (None, 16)                16400

 activation_10 (Activation)  (None, 16)                0

 dense_6 (Dense)             (None, 16)                272

 activation_11 (Activation)  (None, 16)                0

 dropout_4 (Dropout)         (None, 16)                0

 dense_7 (Dense)             (None, 4)                 68

 activation_12 (Activation)  (None, 4)                 0

=================================================================
Total params: 82,308
Trainable params: 82,308
Non-trainable params: 0
_________________________________________________________________
```

Figure 4: Model summary

3.3. Training a model:

The loss function, optimization function, and metrics to assess the model's performance must be calculated before a neural network model may be trained. The categorical cross entropy is the loss function in this case. A loss function used in multi-class classification challenges is the cross-entropy classifier. In certain situations, the sample can only fit into one of several categories, and the model must select that category. It is intended to quantitatively measure the variation between two probability distributions. This model uses the optimization function adam with a learning rate of 0.001. Adam is an alternative stochastic gradient descent optimization technique for deep learning model training. Adam creates an optimization technique that can handle sparse gradients in noise issues by combining the best elements of the AdaGrad and RMSProp algorithms. Accuracy is calculated as metrics.

```
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

3.4. Saving the trained model:

A ".h5" file contains the trained model. The data for final weights and bias are contained in that file. This prevents the need for repeated training, can be simply applied to new models, and can be used to deliver work on the front end using flask or other web frameworks.

4. RESULTS AND DISCUSSIONS:

Images used for training and testing are sent into the convolutional neural network. The loss in the output is back-propagated through the optimization function in the network, which comprises the layers described. This will continue for the 300 epochs that were predetermined before the output layer, which comprises 4 neurons and shows the classes of "Large, Medium, Small, No Crack," contains neurons. The class that the network predicted is represented by the maximum of the four neurons, and the accuracy and loss of the training and testing sets of data are shown with the use of the matplotlib package. The accuracy of the exam was 88.76 percent, while the accuracy of the training was 86.2 percent. As a result, the accuracy of difficulties with picture categorization, which is greater than 80%, is significantly better. As a result, the model does well with this categorization issue.

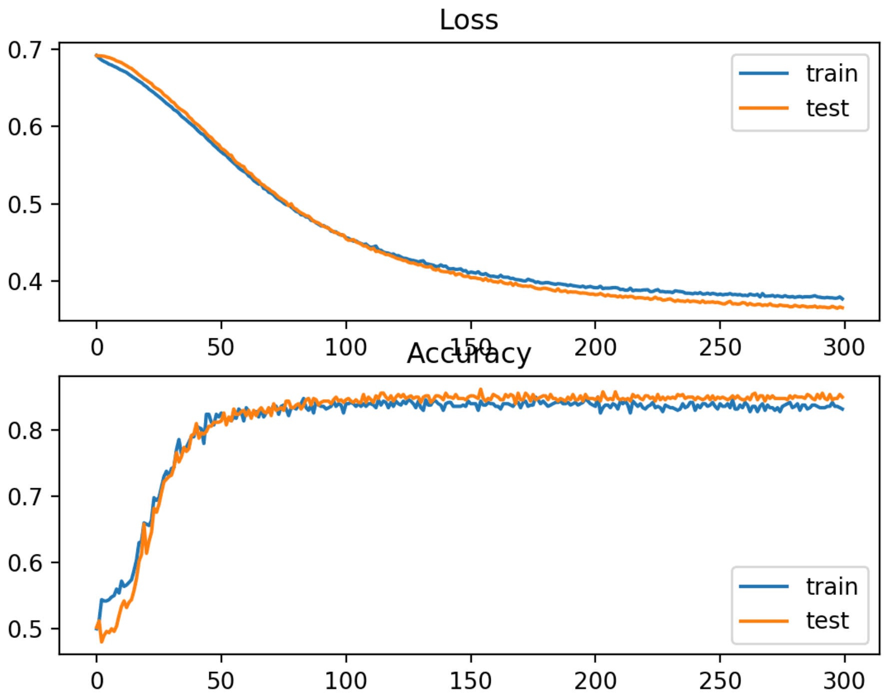

Figure 5: loss and accuracy plots of test and train data

The confusion matrix for the model performance evaluation test is displayed. The item's actual worth is depicted by the vertical line. The class to which the sample belongs in your data collection is referred to as the true value. You have four courses, of course. The horizontal refers to the classes that the model predicted for those samples. Recall, f1-score, and accuracy may be calculated from this confusion matrix in order to assess the model performance.

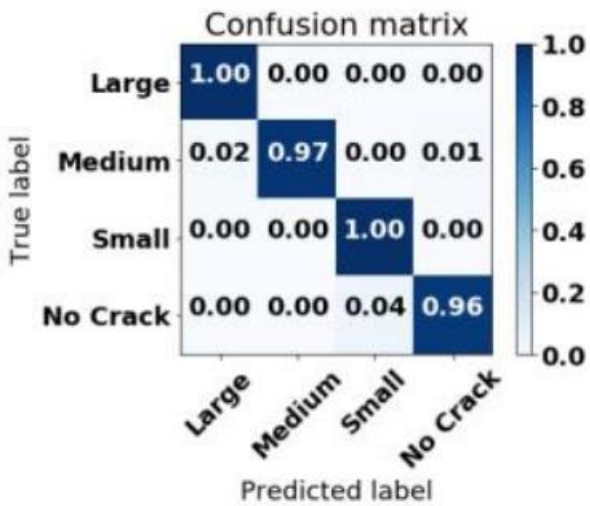

Figure 6: Confusion matrix

The picture is transmitted to the stored model, which contains the final weights and bias, to test the model. The expected values are displayed in figure 6 after passing a test image. These photographs have a 99.8% accuracy rate for predicting major cracks, a 100% accuracy rate for predicting no crack images, and a lower accuracy rate for other images because of poor image clarity. Therefore, this model may be used to locate cracks in aircrafts, categorise large and tiny fractures, and offer engineers information they can use to designate crack-free areas for repairs. rather than using the eyes, which take a lot of time.

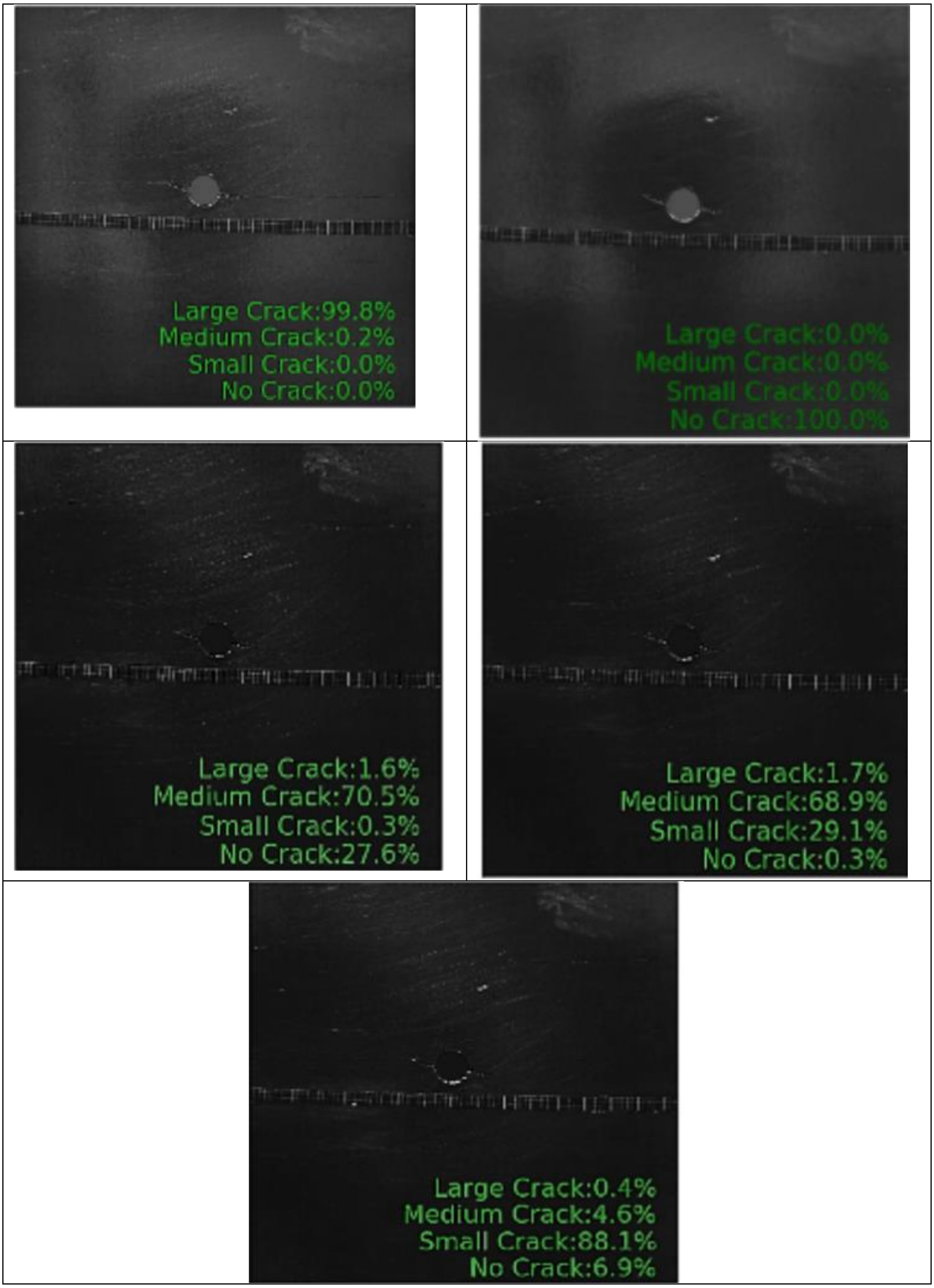

Figure 2: test images and results

5. Conclusion:

This research demonstrates the use of domain knowledge in automated visual inspection for creating a deep learning forecast that can be justified. The clustering method was able to divide the 11803 photos into 4 distinct groups, but not in accordance with the 4 different crack lengths, but rather in accordance with variations in light intensity that occurred while the photographs were being recorded. The network that was trained using variations in light intensity achieved an accuracy of 88.76%. On the other hand, the biased CNN, which is taught using various fracture lengths, obtained accuracy of 87.2 percent. These findings support our earlier hypothesis that AI-assisted NDT or structural health monitoring (SHM) should incorporate human input in addition to pure unsupervised learning. However, as human judgement may sometimes be flawed, we think that rather than relying only on either computers or humans, future NDT and SHM should be built on reciprocal human-computer interaction.

References

[1] Rice M, Li L, Ying G, Wan M, Lim ET, Feng G, Ng J, Jin Li MT, Bab VS. Automating the Visual Inspection of Aircraft. Singapore Aerospace Technology and Engineering Conference, Singapore, 2018.

[2] Albawi S, Mohammed TA, Al Zawi S. Understanding of a Convolutional Neural Network. International Conference on Engineering and Technology (ICET), Antalya, 2017.

[3] Sawaf F, Groves RM. Phase Discontinuity Predictions Using Machine-Learning Trained Kernel. Applied Optics, 2014. Vol. 53: pp 5439 – 5447.

[4] M. Sirois and A. Raude, "Using Eddy Current Array in Detecting Cracks and Corrosion in Aircraft Longerons | 2016-07-01 | Quality Magazine," Using Eddy Current Array in Detecting Cracks and Corrosion in Aircraft Longerons | 2016-07-01 | Quality Magazine, Jul. 01, 2016.

[5] Cha YJ, Choi W, Büyüköztürk O. Deep Learning Based Crack Damage Detection Using Convolutional Neural Networks. Computer Aided Civil and Infrastructure Engineering, 2017. Vol. 32: pp 361 – 378.

[6] Gopalakrishnan K, Khaitan SK, Choudhary A, Agrawal A. Deep Convolutional Neural Networks with Transfer Learning for Computer Vision-Based Data-Driven Pavement Distress Detection. Construction and Building Materials, 2017. Vol. 157: pp 322 – 330.

[7] Da Silva WRL, De Lucena DS. Concrete Cracks Detection Based on Deep Learning Image Classification. 18th International Conference on Experimental Mechanics (ICEM18), Brussels, 2018.

[8] Li S, Zhao X. Convolutional Neural Networks-Based Crack Detection for Real Concrete Surface. SPIE Smart Structures and Materials + Nondestructive Evaluation and Health Monitoring, Denver, 2018.

[9] Yokoyama S, Matsumoto S. Development of an Automatic Detector of Cracks in Concrete Using Machine Learning. Procedia Engineering, 2017. Vol. 171: pp 1250 – 1255.

[10] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-Based Learning Applied to Document Recognition. Proc. of IEEE, 1998. Vol. 86: pp 2278 – 2324.

[11] Mitchell TM. Machine Learning. McGraw Hill, 1997.

[12] Rosasco L, De Vito E, Caponnetto A, Piana M, Verri A. Are Loss Functions All The Same? Journal of Neural Computation, 2003. Vol. 16: pp 1063 – 1076.

[13] Baxter J. A Model of Inductive Bias Learning. Journal of Artificial Intelligence Research, 2000. Vol. 12: pp 149 – 198.

[14] Standard Test Method for Measurement of Fatigue Crack Growth Rates: ASTM E647 – 15E1. American Society for Testing and Materials.

[15] L. P. Zuckerman, "Efficient method for running Fully Convolutional Networks (FCNs) | by Liad Pollak Zuckerman | Towards Data Science," Medium, Sep. 13, 2019.