UG RESEARCH WORK

TITLE: Deep learning-based damage detection and its significance in structural health monitoring

By

HIMANSHU SINGH

(Institute Roll No. : 180104044)

DEPARTMENT OF CIVIL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, GUWAHATI

Guwahati – 781039.

TABLE OF CONTENTS

1. Abstract

Whole world has been growing rapidly with new high-rise buildings, bridges, dams, tunnels and other urban infrastructure coming up demanding for more advanced and efficient Structure Health Monitoring Systems. The application of health monitoring in various structures is highly beneficial as it provides enhanced public safety, improved life span of structures, and early detection of risks thus helping in improving the overall performance of the structures. The process of manually inspecting structural conditions and damage however is time-consuming, error-prone, and can be dangerous for the human inspectors. Presence of subsurface defects in the structure may lead to severe accidents as the operational stress tends to concentrate around these defects. Ultrasonic waves can propagate over relatively long distances and are able to interact sensitively with and uniquely with different types of defects. The captured back-propagated waves can give the precise location of the defects and necessary steps can be taken based on the severity of the defect. This research work is aimed to develop a novel method to detect surface and subsurface damage by using computer vision and deep learning models trained on phased-array ultrasonic images. Structure Health Monitoring using computer vision and deep learning techniques have gained significant attention in recent decades. Deep learning can contribute to the traditional discipline with much better performance than existing methods. The major limitation in this approach is that limited data is available for training the models but with high efficient numerical simulations and data augmentation, more number of damaged scenario data than real, can be simulated which can boost the accuracy of these models.

Keywords: Structural health monitoring, Phased-array ultrasonic data, Computer vision, Deep learning, Data augmentation.

2. Theory

The subsurface flaw/crack detection system proposed in our research work consist of two parts:

1. Phased-array ultrasonic data:

   Phased array ultrasonic systems utilise multi-element probes, which are individually excited under computer control. By exciting each element in a controlled manner, a focused beam of ultrasound can be generated. Software enables the beam to be steered. Two and three dimensional views can be generated showing the sizes and locations of any flaws detected.[1]

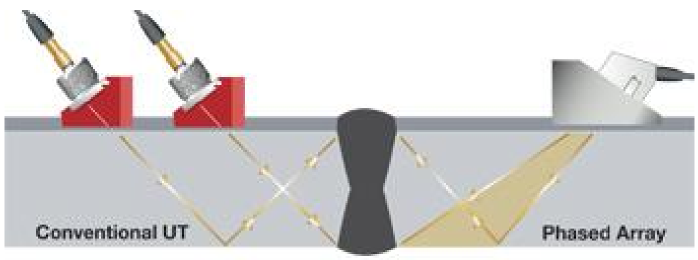

Figure 1 [2]

Figure 2

2. Convolutional neural network for predicting if the structure is flawed/cracked or not:

   A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign learnable weights and biases to various aspects/objects in the image and be able to differentiate one from the other[3]

   Optimizer: Optimizers are algorithms or methods used to change the attributes of the neural network such as weights and learning rate in order to reduce the **losses**. There are several optimizes(adaptive) available though for our research work, we have considered three widely used optimizers namely:

- Adam[4]
- SGD[5]
- RMSprop[6]

3. Methodology

3.1. Data

The data used in our research work was taken from [7] where phased-array ultrasonic was used in the detection of flaws in a butt-weld in an austenitic 316L stainless steel pipe. Three thermal fatigue cracks with depths 1.6, 4.0 and 8.6 mm were implemented in the inner diameter of the pipe near the weld root and scanned with ultrasonic equipment as briefly described in the theory.

3.2. Data Augmentation

Data augmentation was implemented to both training and validation dataset using keras data generator. The augmentation included rotation, horizontal shift, horizontal flip, zooming and brightness variations. The original dataset contained 20000 samples in total for training and validation. The data generator didn't add any new image to the dataset instead, it provided slightly altered images in each epoch thus enhancing the robustness of the model and reducing overfitting. The model performances with and without data augmentation was compared and is shown in the further section.

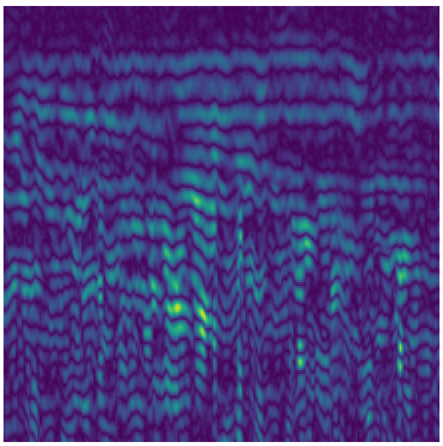

Figure 3: Original Image, from the dataset

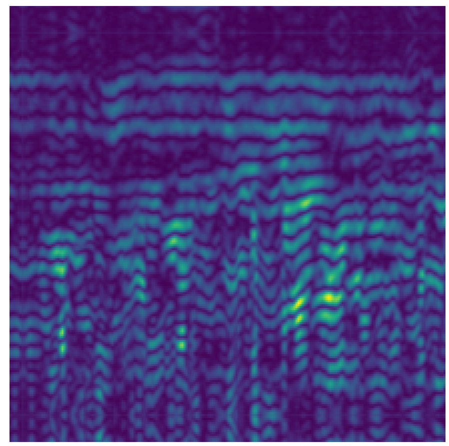

Figure 4: Augmented data, generated using keras

3.3. Model architecture:

ResNet-50 architecture was used for training the dataset, due to its ability to train very deep neural networks without encountering the vanishing gradient problem. The model was slightly modified at the end to cater the needs of our experiment. The fully-connected output layer in the original ResNet-50 architecture was replaced by two fully-connected layers preceded by global average pooling and dropout layers as represented in Fig.[6]

ResNet-50 is one of the members of the ResNet family which has emerged recently showing state-of-the-art performance. It utilized the concept of skip connections or shortcuts to jump over some further layers within a residual block. Several residual blocks were stacked up to build really deep neural networks whose details are mentioned in the paper [8].

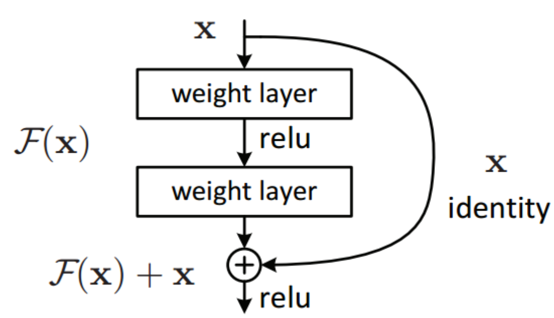

Figure 5: Residual learning: a building block

Dropout [9] layer was used to prevent the model from overfitting. Dropout works by randomly setting the outgoing edges of hidden units (neurons that make up hidden layers) to 0 at each update of the training phase.

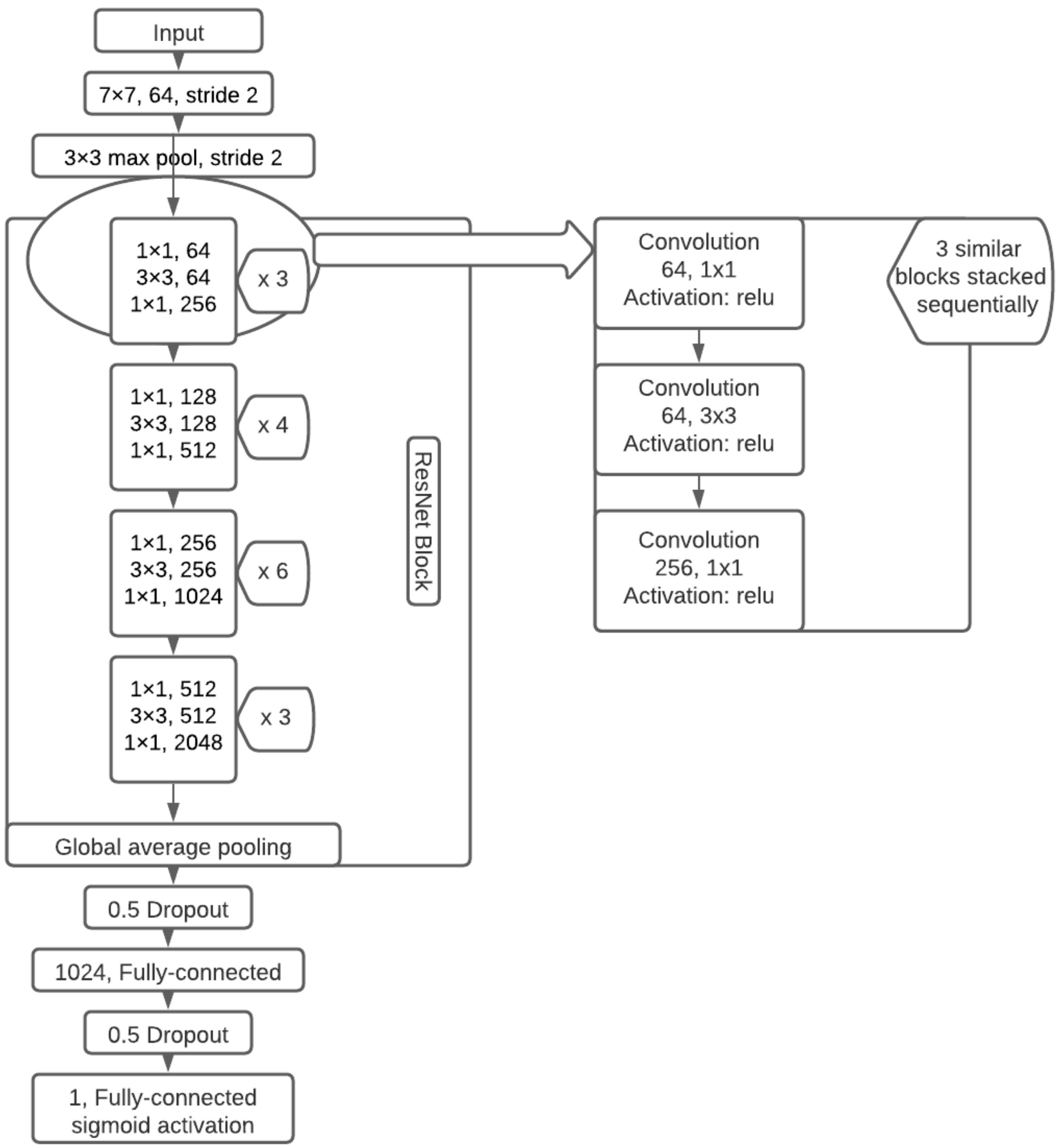

Figure 6: The model architecture

3.4. Training the model

The computation was implemented with the Keras library [10] using the TensorFlow as back-end [11].

The data was read in the saved mini-batches, converted to 32 bit floating point numbers and normalized by subtracting the mean and dividing by standard deviation. A small value of 0.00001 was added to avoid division by zero [7]. The network was trained for 60 epochs with a batch size of 32. The model was trained with binary cross entropy as the loss function with three different optimizers namely, Adam, SGD and RMSprop, set with learning rates {0.001,0.0001}. The performances of all the combinations were compared and is shown in further sections.

4. Observations

- Model performance using augmented dataset for different optimizers and learning rate

I. Learning rate = 0.001

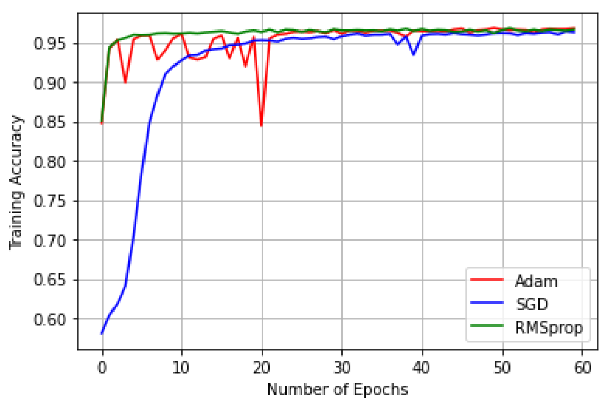

Figure 7: Training accuracy during training for 60 epoch, using different optimisers

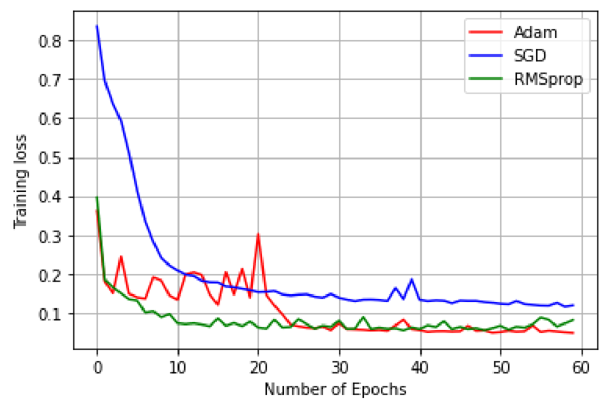

Figure 8: Training loss during training for 60 epoch, using different optimisers

II. Learning rate = 0.0001

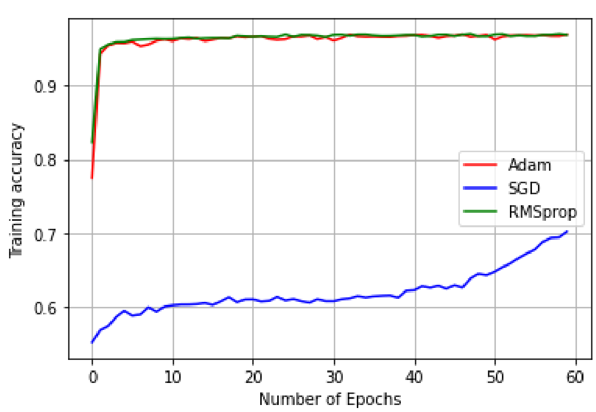

Figure 9: Training accuracy during training for 60 epoch, using different optimisers

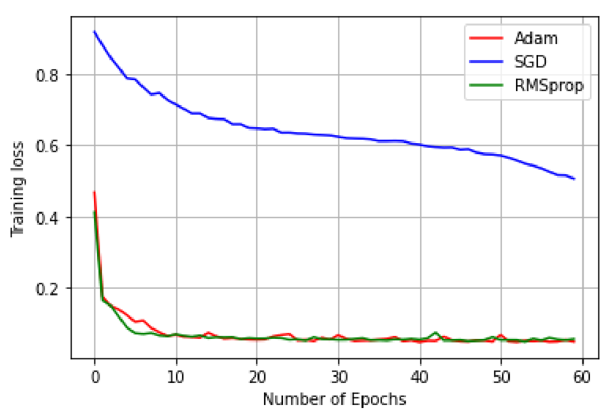

Figure 10: Training loss during training for 60 epoch, using different optimisers

The above results can be summarised in the table shown below:

Model	Learning rate	Optimizer	Training accuracy(%)	Training loss
ResNet-50	0.001	Adam	96.815	0.04812
		SGD	96.324	0.11871
		RMSprop	96.673	0.08116
	0.0001	Adam	96.936	0.04779
		SGD	70.204	0.50610
		RMSprop	96.885	0.05496

Table 1: comparing the performance of ResNet-50 with different optimizers and learning rates

- Model performance comparison with original and augmented data

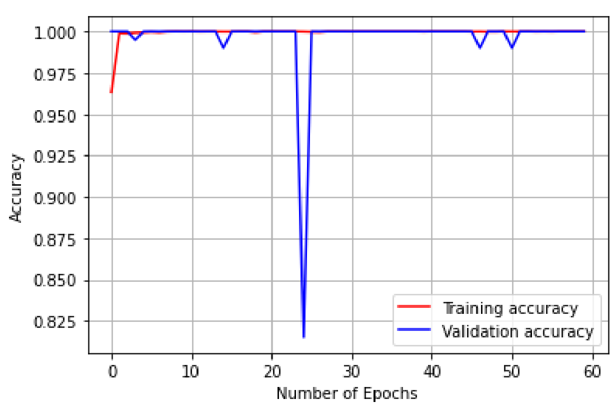

Figure 11: Training accuracy during training for 60 epoch, with original data

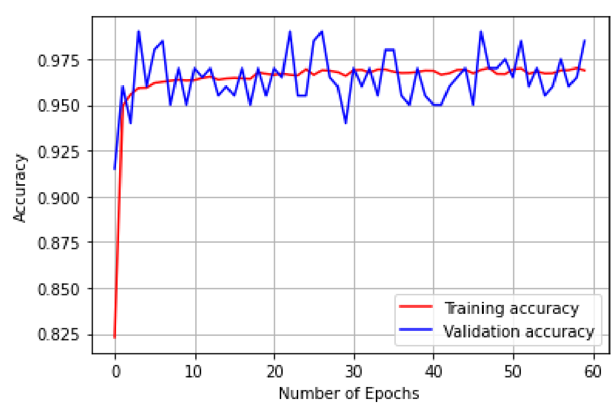

Figure 12: Training loss during training for 60 epoch, with augmented data

	Training accuracy	Validation accuracy
Without data augmentation	100%	100%
With data augmentation	96.936%	97.5%

Table 2: Effect of data augmentation on the performance of neural networks. Without data augmentation, the training and validation accuracy are both constant at 100% which are the signs of spurious prediction.

5. Result and Discussion

- Based on the performance observation of different optimizers, **Adam** was found to be superior over others, at the learning rate of **0.0001**. With these hyperparameters, the model accuracy was found to be 96.936% on training dataset and 97.5% on validation dataset.
- Without data augmentation, training and validation dataset accuracy remained constant at 100% with increasing number of epochs as shown in the Fig[9], which might be due to the insufficient number of features. Data augmentation creates a rich, diverse set of images from a small set of images

6. Conclusions

The following conclusions can be drawn from this study:

- Deep convolutional neural networks are powerful enough to perform brilliantly in detecting cracks from ultrasonic data.
- Data augmentation can help reduce the data overfitting when enough training examples are not available.

7. Future Scope

- Subsurface cracks are particularly dangerous since they can cause surface cracks and breakouts, so it is essential to monitor the structures regularly and accurately. Phased-array ultrasonics have been used for detecting steel cracks for a long time, but the use of deep learning and computer vision for detecting the flaws in the image data are very limited. The use of these techniques not only save a lot of time and resources but are also efficient compared to human inspections[7]
- The phased-array ultrasonics are largely used for detecting cracks in the steel but it can be used for concrete structures and slight modifications in frequency and bandwidth of transducers, as shown in the paper [12]. This would allow us to monitor almost all the civil engineering structures efficiently and frequently.
- Real-time monitoring could be implemented in the future for inspection of mega civil engineering structures. The phased-array ultrasonic system can be installed at specific locations in the structure which are prone to failure for continuous data collection. The data after preprocessing and be fed directly into the neural network to detect the cracks autonomously. This way, early warning signs can be provided to fix the crack and avoid heavy losses, which may have resulted from the sudden collapse of the structure.

References

[1] TWI Ltd. Phased Array Ultrasonic Testing-PAUT-NDT.

[3] S. Saha. 2018. “A comprehensive guide to convolutional neural networks”. Accessed from: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[4] Kingma, D.P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. *CoRR, abs/1412.6980*.

[5] Ruder, S. (2016). An overview of gradient descent optimization algorithms. *ArXiv, abs/1609.04747*.

[6] G., Hinton, N., Srivastva, K., Swersky, Overview of mini-batch gradient descent, Neural Networks for Machine Learning

[7] Virkkunen, I., Koskinen, T., Jessen-Juhler, O., & Rinta-Aho, J. (2019). Augmented Ultrasonic Data for Machine Learning. *ArXiv, abs/1903.11399*.

[8] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[9] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1 (January 2014), 1929–1958.

[10] Chollet, F. (2015). Keras.

[11] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I.J., Harp, A., Irving, G., Isard, M., Jia, Y., Józefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *ArXiv, abs/1603.04467*.

[12] Paris, O., Poidevin, C., Rambach, J., & Nahas, G. (2006). STUDY OF PHASED ARRAY TECHNIQUES FOR CONCRETE INSPECTION. *International Journal of Microstructure and Materials Properties, 1*, 274-281.