Segmenting Delaminations in Carbon Fiber Reinforced Polymer Composite CT using Convolutional Neural Networks

Daniel Sammons[1,2,a)], William P. Winfree[1], Eric Burke[1] and Shuiwang Ji[2]

[1]*NASA Langley Research Center, Hampton, VA, 23681*
[2]*Old Dominion University, Norfolk, VA 23529*

[a)]Corresponding author: daniel.m.sammons@nasa.gov

Abstract. Nondestructive evaluation (NDE) utilizes a variety of techniques to inspect various materials for defects without causing changes to the material. X-ray computed tomography (CT) produces large volumes of three dimensional image data. Using the task of identifying delaminations in carbon fiber reinforced polymer (CFRP) composite CT, this work shows that it is possible to automate the analysis of these large volumes of CT data using a machine learning model known as a convolutional neural network (CNN). Further, tests on simulated data sets show that with a robust set of experimental data, it may be possible to go beyond just identification and instead accurately characterize the size and shape of the delaminations with CNNs.

INTRODUCTION

Nondestructive evaluation (NDE) utilizes a variety of techniques to inspect various materials for defects without causing additional damage to the material. X-ray computed tomography (CT) produces large volumes of three dimensional image data. Often, an expert must analyze these large, volumetric datasets. Such manual analysis is a very labor-intensive task that would greatly benefit from automation.

Convolutional Neural Networks (CNNs) are a class of highly non-linear machine learning models that recently became popular after achieving state-of-the-art results for image recognition [1]. CNNs have also been successful at a number of other tasks including object detection and image segmentation in medical imagery [2, 3].

This work investigates the use of CNNs to automate the analysis of CT data. In particular, it focuses on using a CNN to segment delaminations from carbon fiber reinforced polymer (CFRP) composite CT data.

RELATED WORK

Most previous work analyzing NDE data utilizes traditional image processing techniques. For example, [4] experimented with applying several different image processing techniques, such as adaptive thresholding, to CFRP composite. Applying neural networks to NDE is not unprecedented, however, as [5] used a neural network to classify welding defects after segmenting the images using a background removal technique.

Some recent work has attempted to apply a novel statistical anomaly detection technique known as "cross-hatch regression" to CFRP composite [6]. This work was very successful at identifying delaminations in CFRP composite structures. The challenge for this approach will be adapting to different types of structures.

From a computer vision and machine learning standpoint, there has been a large amount of work analyzing similar types of imagery. In particular, [3] and [2] have been successful at analyzing medical imagery with CNNs. More closely related to NDE, [7] used CNNs to classify steel defects.

TABLE 1. Example architecture for patch size of 201 × 201 pixels. Lines between layers represent ReLUs.

Layer	Type	Output	Kernel Size
0	input	1 map of 201 × 201	
1	convolutional	48 maps of 198 × 198	4 × 4
2	max pooling	48 maps of 99 × 99	2 × 2
3	convolutional	48 maps of 96 × 96	4 × 4
4	max pooling	48 maps of 48 × 48	2 × 2
5	convolutional	48 maps of 46 × 46	3 × 3
6	max pooling	48 maps of 23 × 23	2 × 2
7	convolutional	48 maps of 20 × 20	4 × 4
8	max pooling	48 maps of 10 × 10	2 × 2
9	convolutional	48 maps of 8 × 8	3 × 3
10	max pooling	48 maps of 4 × 4	2 × 2
11	convolutional	48 maps of 2 × 2	3 × 3
12	max pooling	48 maps of 1 × 1	2 × 2
13	fully connected	200 maps of 1	1 × 1
14	fully connected	2 maps of 1	1 × 1

BACKGROUND

As previously mentioned, CNNs are a class of highly non-linear machine learning models. Figure 1 shows an example of a CNN processing CFRP composite.

CNNs are typically made up of several alternating layers that perform learned transformations to the input and eventually map the input down to a one-dimensional vector that is classified with a traditional neural network. There are many different configurations of the various types of layers for CNNs. For this work, however, only convolutional layers and max pooling layers were utilized. Convolutional layers operate by convolving a learned kernel of weights with the output from the previous layer. Max pooling layers operate by taking the maximum value of a small region and assigning that value to the output for that region. More details about convolutional layers, max pooling layers, and CNNs in general can be found in [8].

APPROACH

The following describes specific adaptations that were made to train a CNN to segment delaminations in CFRP composite CT.

Patchwise Training

Typically, CNNs are used to make a prediction on an entire image. However, CNNs can be adapted to perform segmentation by classifying small patches sampled from the image where the class of the patch corresponds to the class of the center pixel in the patch. For example, in Fig. 1, the blown up region represents a patch whose class would be delamination since the center pixel in the patch lies within the delamination. By classifying each patch, the trained CNN is able to make a prediction for each pixel, effectively segmenting the image.

The network was trained using the trivial approach of treating each patch as an image. When applying the trained network for classification of a whole image, however, there is significant overlap between neighboring patches. So, instead of processing each patch individually, the learned kernels are convolved with the entire image instead of with each individual patch. This saves significant computation time compared to classifying each patch individually. More details on this approach, including how to deal with max-pooling layers, are given in [9].

Because of similarity of CT CFRP composite data to neuronal data, the architectures used for this work took significant inspiration from [3]. Thus, initially, a patch of 65 × 65 pixels was used. After some testing, however, it was found that a patch of size 65 × 65 pixels did not provide enough context to perform accurate segmentation since the delaminations often are larger than 65 × 65 pixels. In order to accommodate this, the patch size was increased to 201 × 201 pixels. Table 1 shows an example architecture using patches of 201 × 201 pixels.

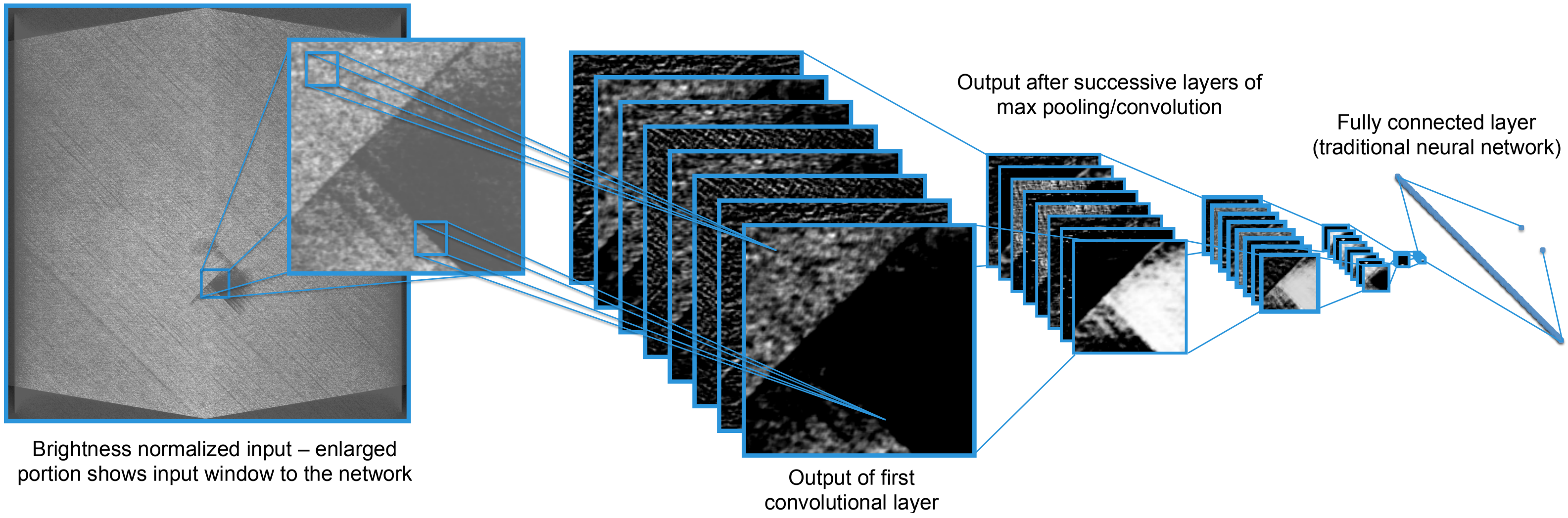
Brightness normalized input – enlarged portion shows input window to the network
Output of first convolutional layer
Output after successive layers of max pooling/convolution
Fully connected layer (traditional neural network)

FIGURE 1. An example CNN processing a patch of CFRP composite. The images in each layer display the features derived from the patch at that layer.

TABLE 2. Precision and recall of found regions at various intensities.

Intensity	1	2	3	4	5	6	7	Total
Precision	74.82%	80.58%	72.01%	71.78%	73.15%	73.57%	73.58%	74.12%
Recall	22.80%	71.91%	81.98%	82.80%	81.60%	80.30%	78.41%	71.64%

Intensity Normalization

CT data can have significant variation in the range of the raw data both between data sets and between slices in a data set. Such variation is best described as differences in the "intensity" of the images and can cause significant trouble for CNNs. As such, a method known as "histogram matching" was used to map all of the data into a common range.

Histogram matching typically works by first calculating the histogram and cumulative distribution function (CDF) for an image. Then, linear interpolation is used to create a mapping from each point in the CDF to the corresponding point in the original image. This mapping can then be used to map the CDF of another image to a histogram similar to that of the original image. For this work, instead of using the CDF of a particular image as the original or "base" image, the CDF of a Gaussian distribution was used. Thus, all of the data were mapped to a Gaussian distribution on a slice-by-slice basis.

Simulating Data

Since manually segmenting CT images is a labor-intensive process and is cost prohibitive to acquire, there was not a large set of labeled data to use for training. In order to accommodate for this lack of training data, delamination-like shapes were added to undamaged CFRP composite backgrounds, simulating images that are typical of damaged CFRP composite. This approach allowed for a large and robust training set to be produced quite rapidly. Further, simulated data allowed for more comprehensive testing as a set containing different variations of delaminations could be produced. For example, by changing a single parameter in the simulation script, delaminations of several intensity differences were produced. Examples of simulated data at seven intensity difference can be seen in Fig. 2.

Fine-tuning

Although the simulated data was able to capture a large amount of the variation in the data for training, it still was not fully representative of the variation that exists in experimental data. Thus, to accommodate for this when classifying experimental data, a small set of labeled experimental data was used to "fine-tune" a network that had previously been trained on simulated data. Fine-tuning, a common technique used to train CNNs, involves taking the weights of a network trained for a different task or on a different data set and using them as the initial values of the weights when training with a new data set. So, in this case, the weights that were obtained from training the network on simulated data were used as the initial values of the weights of a network that was trained on experimental data. The idea behind fine-tuning is that, particularly in the lower levels, the weights that are learned should exhibit some sort of common pattern that is independent of the task. In other words, they should already be close to their final values. As a result, the second batch of training (i.e. fine-tuning) will require significantly less data and time to adapt the weights to useful values.

RESULTS

The network was first trained on a subset of simulated data and tested using the remaining simulated data. Afterwards, the network was fine-tuned using a small set of labeled experimental data and then used to classify other examples of experimental data. The following describes performance of the network on both simulated and experimental data in more detail.

Testing on Simulated Data

Figure 2 shows the results from classifying two slices of simulated data with the delamination at several different intensities. Table 2 describes the pixel-wise precision/recall achieved by the network at each intensity.

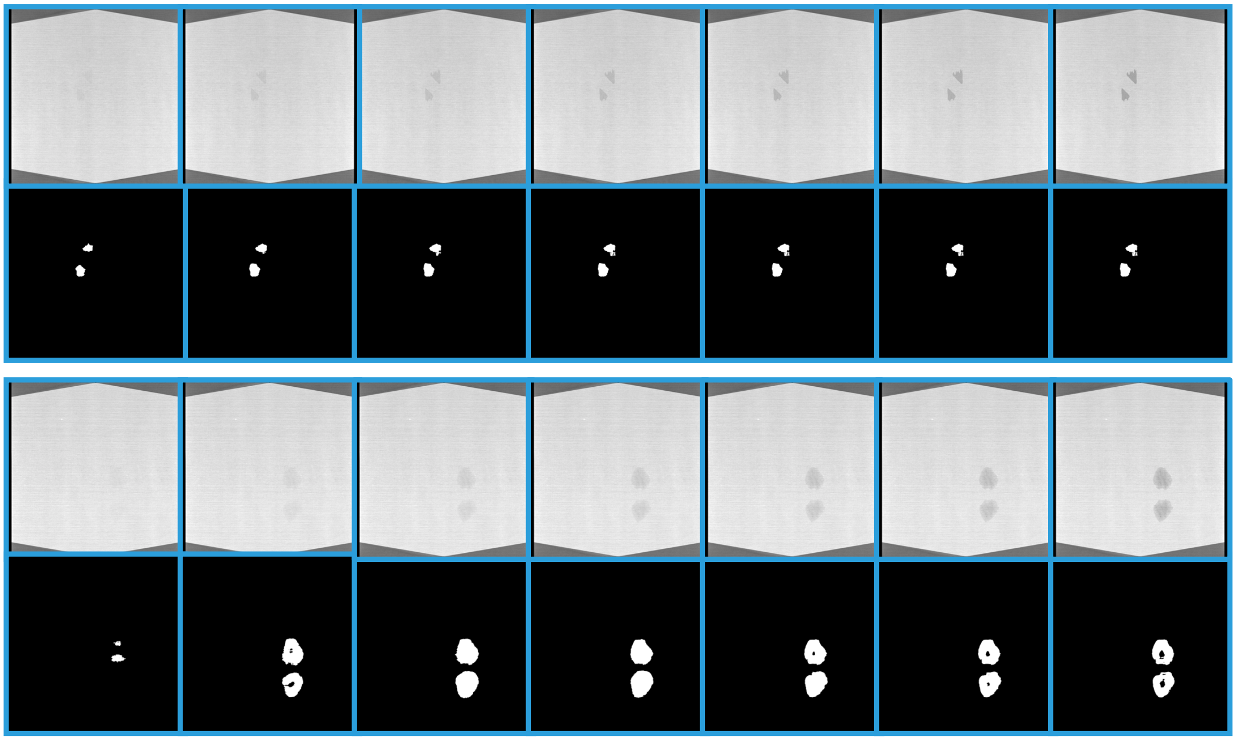

FIGURE 2. Two slices with delamination at varying intensities and the results from classifying those slices.

TABLE 3. Number of ROIs found, missed, and number of false positive ROIs. An ROI is considered found if at least one pixel located in the ROI is returned.

	Count	Percent
Located ROIs	733	89.17%
Missed ROIs	89	10.83%
False Positive ROIs	127	13.19%

The CNN was able to locate the simulated delaminations for all intensity differences though it consistently struggled to find the entire delamination at the smallest intensity difference. Interestingly, the network also struggled to find the entire delaminated region for larger delaminations as well (Fig. 3). The inability to appropriately size large delaminations results from the region the CNN is using to classify a particular pixel falling completely within the delaminated region which does not have the variation at the edges of the region that is typical for smaller delaminations. This indicates that more context may be needed to accurately segment larger regions. Comprehensive results detailing the number of regions of interest (ROI) the network found/missed and the percent of false positives from the entire simulated data set are shown in Table 3. Generally, the network was able to locate the regions with a relatively low false positive rate (≈13%). When the network did miss an ROI completely it was typically either a small delamination or a delamination of very low intensity.

Testing on Experimental Data

As previously mentioned, in order to deploy the model on experimental data, the model that was initially trained with simulated data and then fine-tuned using a small set of labeled real data. Such an approach works very well for NDE where it is often difficult to get an extensive set of labeled experimental data but it is possible to simulate data. Some results from classifying a different, unlabeled subset of real data are shown in Fig. 4. Generally, the model is able to locate delaminations in images where damage exists with relatively few false positives.

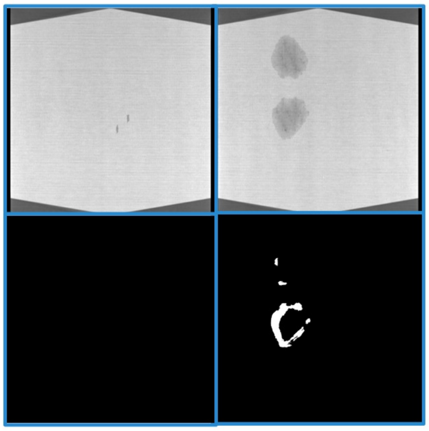

FIGURE 3. Even when using larger patches, the network struggled to find smaller and larger delaminations

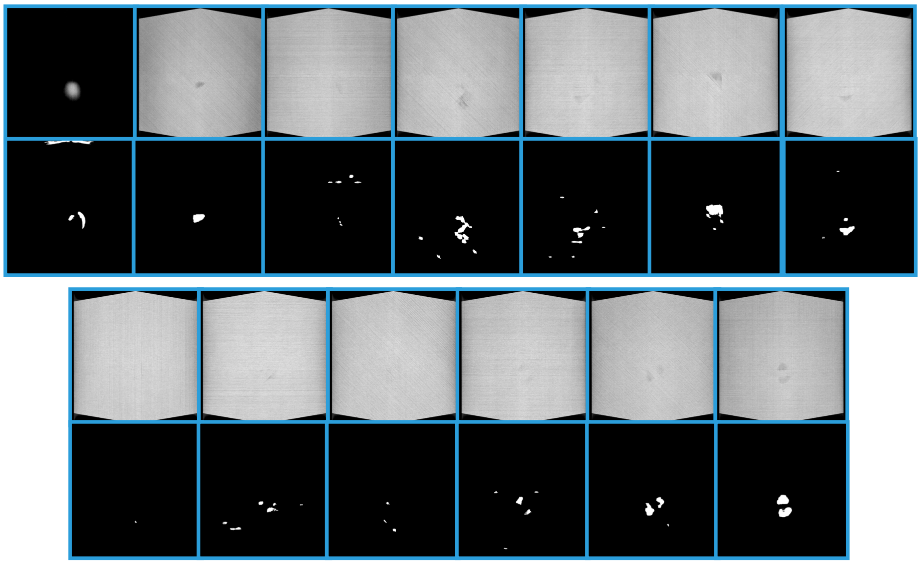

FIGURE 4. Real data that was classified after fine-tuning with a small set of labeled real data. The first image is empty while the rest of the images contain some amount of damage.

CONCLUSIONS

This work shows that a CNN can be used to automatically identify delaminations in CFRP composite CT. Moreover, success of sizing delaminations in simulated data suggests that with more experimental training data, it should be possible to use CNNs to approximate the size of delaminations in CFRP composite CT. As was noted in the results, larger delaminations caused problems when the entire patch the CNN used for prediction was mostly inside the delaminated area. Because of computational limitations, the naive solution of further increasing the window size is infeasible. Instead, other techniques that increase the context that the model is using to make predictions should be used. Multi-scale architectures for CNNs [10, 11] seem to fit this requirement and will be explored in future work.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems* (2012), pp. 1097–1105.

[2] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013* (Springer, 2013), pp. 411–418.

[3] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, “Deep neural networks segment neuronal membranes in electron microscopy images,” in *Advances in neural information processing systems* (2012), pp. 2843–2851.

[4] A. K. Jain and M. Dubuisson, Pattern Recognition **25**, 257–270 (1992).

[5] G. Wang and T. W. Liao, NDT & E International **35**, 519–528 (2002).

[6] C. D. Lockard, “Anomaly detection in radiographic images of composite materials via crosshatch regression,” Ph.D. thesis, MILLS COLLEGE 2015.

[7] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, and G. Fricout, “Steel defect classification with max-pooling convolutional neural networks,” in *Neural Networks (IJCNN), The 2012 International Joint Conference on* (IEEE, 2012), pp. 1–6.

[8] Y. Bengio, I. J. Goodfellow, and A. Courville, “Deep learning,” (2015), book in preparation for MIT Press.

[9] A. Giusti, D. C. Cireşan, J. Masci, L. M. Gambardella, and J. Schmidhuber, “Fast image scanning with deep max-pooling convolutional neural networks,” arXiv preprint arXiv:1302.1700 (2013).

[10] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, IEEE Transactions on Pattern Analysis and Machine Intelligence **35**, 1915–1929 (2013).

[11] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” arXiv preprint arXiv:1411.4038 (2014).