

# Deep Neural Networks Analysis of Borescope Images

Markus Svensén, David S Hardwick and Honor E G Powrie

*GE Aviation Digital, Eastleigh, Hampshire, United Kingdom*

*{Markus.Svensen, David.Hardwick, Honor.Powrie}@ge.com*

## ABSTRACT

This paper presents results on applying deep neural networks to analysis of images from borescope inspections of large turbofan engines, carried out in the field. Such inspections are done as a part of routine monitoring and maintenance as well as an initial, investigative response to alerts from automatic monitoring systems, pilots or engineers. Across GE's commercial engines fleet, a substantial number of images have been gathered this way. The deep learning techniques that have come out of computer vision and machine learning research in the last decade offer new possibilities for analyzing and mining such data collections. This paper presents initial results on separating borescope images from images created with regular digital cameras, as well as classifying images containing various engine parts, with average accuracy of 95% and 77%, respectively, on unseen validation data.

## 1. INTRODUCTION

Borescope inspections are vital in the monitoring and maintenance of jet engines and other industrial assets. They allow visual inspection of parts inside the engine, without disassembly of the engine, by inserting a fixed or flexible probe, at the end of which is a camera and a light source. They are very similar to endoscopes, that are used e.g. for medical purposes, such as investigations of the human gastro-intestine tract. Figure 1 shows an example of a commercial borescope system. In addition to allowing an immediate view inside the engine undergoing inspection, most borescopes will allow the operator to 'take a picture' saving the current borescope view as a digital image. This is useful to document the current condition of the engine as well as any problems found and may aid in planning any remedial actions required.

GE Aviation is one of the world's leading manufacturers of jet engines and there are currently tens of thousands of GE Aviation engines in operation around the world. GE Aviation is also involved in a lot of the monitoring, maintenance and repair of these engines. Borescope inspections are undertaken by GE Aviation field service engineers and other staff, and

over the years, GE Aviation has accumulated a considerable amount of borescope images (BIs). These image collections contain information that could potentially be used to better understand how engines are affected by day-to-day operations and the normal deterioration processes that set in over time. Thus, GE Aviation is working on developing methods for extracting useful information from collections of BIs.

In the last decade, important developments at the intersection of machine learning and computer vision has led to dramatic improvements across a range of learning tasks. Based on established machine learning models known as neural networks, the new generation of deep convolutional neural networks has delivered ground-breaking improvements on visual processing tasks. Computers are now able to successfully distinguish between objects from over thousand categories in just a fraction of a second. These developments have been driven by the availability of low-cost, high-performing graphic processing units, mass-produced for home entertainment devices such as games consoles, as well as dedicated software libraries for building and experimenting with deep neural networks that can exploit such hardware.



Figure 1 An example of a borescope showing the main unit with display and controls along with the flexible probe that is inserted into the engine.

Markus Svensén et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

This paper presents the first results on applying deep neural networks in the analysis of BIs collected in the field. Two initial tasks are being considered:

1. Separation of BIs and digital photographs. This is being driven by the fact that GE Aviation's current collections of engine images are not fully labelled and there is a considerable number of images for which we do not know whether they originate from a borescope inspection or from another source.
2. Recognition of key parts in the engine hot section. While part recognition is desirable for all engine sections, the hot section is a natural starting point, since it is being inspected more frequently than any other engine sections and thus constitutes the largest amount of training data. The reason for this is that the hot section is subject to the highest temperatures and pressures in the engine, and many of its parts have a limited lifespan.

These tasks represent the initial building blocks for an image-based condition monitoring system. A system that ultimately, given images from an engine, would be capable of producing statistics that summarise aspects of the health of that engine in a meaningful way, e.g. the location and degree of wear of various engine parts.

The next section provides a review of deep convolutional neural networks. Section 3 describes the data we used in our experiments, experiments that are described in detail in the following section, before conclusion and discussion.

## 2. DEEP CONVOLUTIONAL NEURAL NETWORKS

Neural networks have been subject to research bridging biology, neuroscience, computer science and statistics over the last 60 years. Their popularity has varied over the years, but the last decade has seen a surge of interest following significant advances with architectures known as *deep neural networks*. This has been brought about, in part, by advances in hardware and software, as well as the use of crowdsourcing for producing *large, labelled* data sets.

### 2.1. Neural Networks

Neural networks (Bishop 2006) are general, parameterized non-linear regression models. They owe their name to similarities between their underlying computational model and processes assumed to occur in networks of connected neurons in the human brain. A neural network consists of a set of computational units ('neurons') organized in layers, with weighted connections between units in different layers (but typically no connections between units within a layer). Each unit forms the weighted sum of the outputs from the previous layer; the result is either output directly or transformed using some kind of non-linear transfer function, such as the tanh-function. Units in the first layer act as inputs to the network

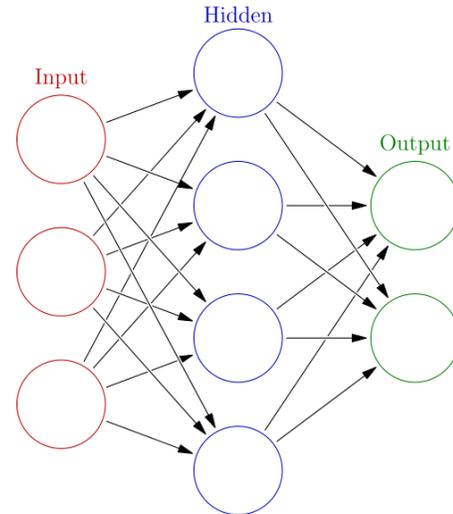


Figure 2 An illustration of a simple neural network with an input, an output and, in between, a hidden layer. The arrows between the layers represent the weighted connections between the units, with the arrows indicating the direction of the forward propagation.

and simply replicate their input on their output. A simple neural network is illustrated in Figure 2. With an input value fed to the inputs of the network, computation can proceed one layer at the time, eventually producing values on the outputs of the units in the output layer; this procedure is referred to as forward propagation.

The weights of the connections between layers form the adjustable parameters of a neural network. Given a data set of input value and matching, desired output values, the weights can be adapted by forward propagating input values through the network and then compare the output that the network produces with the corresponding desired output, using some suitable error measure (e.g. sum-of-squares error in case of least squares regression). This error can then be propagated back through the network, from outputs to inputs to produce a gradient with respect to each weight in the network, a procedure known as back-propagation. These gradients can be used to adjust the weights in order to reduce the total error over the data set. This adaption of the weight parameters is sometimes referred to as learning or training, highlighting analogies with learning in the human brain, but can also be understood as adjusting the weights to maximize the likelihood of the data used for training under the model.

### 2.2. Convolutional Neural Networks

It has been proven that the neural networks discussed in the previous section are universal function approximators (Bishop 2006), i.e. they can *in principle* learn any mapping between inputs and outputs, given a sufficient number of hidden units and training data. However, relying on this theoretical property will rarely work in practice for hard problems

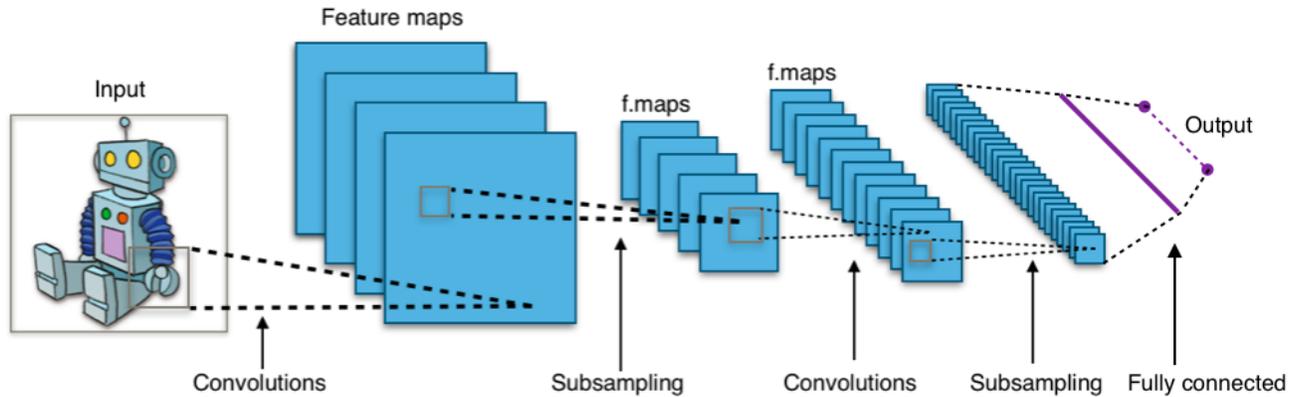


Figure 3 An illustration of a convolutional neural network; see text in Section 2.2 for details.

such as object recognition. It has long been understood that tackling such problems is more likely to succeed if the network structure can exploit structure present in the data. The prime example of this is image data. The pixels in an image get their meaning not just from their own RGB/grayscale values, but also from the values of neighbouring pixels. Similarly, local patches of pixels often form patterns that carry meaning in the context of patterns in neighbouring local patches. It is this property of image data that is being exploited in convolutional neural networks (CNNs) (LeCun, Boser, et al. 1989) (LeCun, Bottou, et al. 1998).

Figure 3 shows an illustration of a simple convolutional neural network. The units in each hidden layer are organised into feature maps, typically with a common shape and size, matching the shape of the preceding layer. Units in convolutional feature maps take their inputs from a patch of units in the preceding layer and all the units in one feature map uses the same weights for the connections to their respective patches. As in conventional neural networks, each unit forms the weighted sum of its inputs. In effect, the feature map at this stage equals the convolution of the input ‘image’ and the discrete spatial filter defined by the connection weights. The weighted sums are typically then transformed using some kind of non-linearity; deep CNNs often use the computationally attractive rectified linear units (ReLU) (Krizhevsky, Sutskever and Hinton 2012). Normally, a layer of convolutional features maps is followed by a corresponding subsampling layer, in which ‘patches are mapped to pixels’ by means of some sort of aggregation, such as averaging or computing the mode or the maximum over the patch. Convolutional feature maps act as detectors of local patterns, the output of which is then smoothed by the following subsampling. CNNs can contain multiple such paired convolution-subsampling layers. The last hidden layer, before the output layer, is typically fully connected to the preceding layer and the following output layer, with no internal, spatial structure (as the hidden layer in Figure 2).

### 2.3. Deep Convolutional Neural Networks

While CNNs have featured in the machine learning literature for almost 30 years, it is only in the last decade that these models have found more widespread use. Several developments, that have to some extent happened in parallel, have contributed to this. In the early–mid 2000s, neural network research focussed on successfully training networks with multiple hidden layers, known as deep neural networks, started to yield results (Hinton, Osindero and Teh 2006) (Bengio, et al. 2007). Although these results did not in themselves have a direct impact on how CNNs are trained, they stimulated research into training deep CNNs (Krizhevsky, Sutskever and Hinton 2012), i.e. CNNs with more layers than had been used previously. The creation of large labelled image databases, such as the ImageNet database (Deng, et al. 2009), which contains millions of labelled images, has meant that there are now sufficient data to train also very large models, with millions of weights. At the same time, hardware capable of delivering the necessary computational power to train large neural networks on large data sets has become affordable. This was initially driven by demand for home games consoles and gaming PCs, but now there is a considerable market for hardware dedicated to (e.g.) train and use large neural network models. Another, more recent development is the availability of free, high-quality, cross-platform software libraries<sup>1</sup> that significantly eases the building, training and evaluation of deep CNNs. Finally, the idea of *transfer learning* (Yosinski, et al. 2014), means that learning tasks where only more moderate amounts of labelled data are readily available can still be attempted using deep CNNs. The idea is to reuse models that have been trained on ‘general’ tasks, such as general object recognition, and re-train these models by adapting only a subset of the network weights, typically those connecting to the final, fully connected layer. The part of the network that is left unchanged in effect act as an

<sup>1</sup> See e.g. [www.tensorflow.org](http://www.tensorflow.org), <http://deeplearning.net/software/theano/>, <http://caffe.berkeleyvision.org/>.

advanced feature extractor, yielding a representation of the image data that is suitable for the learning task at hand.

### 3. IMAGE DATA

GE Aviation has large collections of images from its fleets of different engines. These collections mostly contain BIs, but there is a substantial proportion of digital photographs, most of which have been collected when engines have come in for visits in workshops. Images are generally stored as they were collected, without any additional processing. As is the case with collections of general images, there is substantial variability across these engine images, e.g. in terms of lighting, viewpoint and engine condition.

For the experiments reported below, we used images collected from the GENx fleet. Images were rotated, cropped, resized and transformed as necessary to obtain a data set of 23186 320×240 JPG images. However, during subsequent data exploration work, it was established that among these images were a significant number of duplicates and there were also images that had no relationship to jet engines, so that the actual number of distinct, relevant images ended up being closer to 18800.

While the images have associated annotations in various forms (meta-data, image filenames, etc.), there is no established system by which images of a particular kind or showing a particular engine part can be readily identified. It would thus be useful to develop analytic modules for automatically and consistently labelling images, e.g. according to the type of image or the engine part shown in the image. These would become initial links in a chain of increasingly sophisticated analytic modules forming an image-based condition monitoring system.

### 4. EXPERIMENTS

In this section, we consider these two applications of deep CNNs to engine image data, for the purpose automatic image labelling.

#### 4.1. Borescope Images vs Digital Photographs

The first task we try to address is the separation of BIs and digital photographs, collected by handheld digital cameras, including tablets and mobile phones. While human observers can relatively easily learn to distinguish these two kinds of images with a high level of accuracy, it is difficult to develop a general, algorithmic solution to this problem. This is one of the hallmarks of a problem where supervised machine learning techniques can provide a solution, provided a sufficient labelled data is available.

While images may not have dedicated labels immediately available, many images carry ‘markers’ that reliably identify them as either BIs or digital photographs. The most prominent example is found with BIs collected using GE produced borescopes. Many of these BIs feature the GE monogram in



Figure 4 Two example BIs showing the GE monogram in the bottom-left (left image, also showing a fuel nozzle) and top-left (right image, also showing two high-pressure turbine blades) corners of the images.

the top or bottom left hand corner of the image, as shown in the example images in Figure 4. The placement of the monogram as well as its spatial pattern and colours are all highly consistent across images where it appears, making it relatively straightforward to apply a simple correlation test with a corresponding template to determine whether the monogram is present in either one of the two possible locations. If it is, the image can be labelled as a BI. Another feature that was found useful for labelling images was the original resolution of the image (before it was cropped and resized as described in Section 3); images that have a resolution of more than 2000 pixels either horizontally or vertically will be assumed to originate from a digital camera. Finally, some of the raw source images come with Exif meta-data (Wikipedia contributors n.d.) that explicitly named the camera make and model; where present, these invariably identified known digital cameras or borescopes.

Using these ‘markers’ to label images, we ended up with 12165 BIs, 3016 digital photographs and 3624 images that did not meet the criteria for labelling and hence remained unlabelled. The labelled data was divided into training and validation sets as detailed in the bottom row of Table 1. This division was done such that the images from any one engine all ended up in either training or validation data. Finally, the images were cropped to a size of 290 × 180, excluding the regions potentially containing the GE monogram. The resulting data were used to build and evaluate a CNN for the task of separating BI and digital photographs. This CNN was relatively small and shallow by today’s standards. It used a sequence of two identical blocks, each containing a 3 × 3 × 32 convolutional layer with ReLU activation, followed by a 2 × 2 max-pool layer; these were followed by a similar block, but where the convolutional layer had shape of 3 × 3 × 4; finally, there was a flat, fully connected layer with 32 ReLU activated units feeding the (single) logistic output unit. The size of the final convolutional layer as well as the number of units in the fully connected layer was determined by experimentation, while monitoring the performance on validation data. We also experimented with using different degrees of regularisation and drop-out (Krizhevsky, Sutskever and Hinton 2012); for results reported below, we only used drop-out at a rate of 0.5 for the final, fully con-

nected layer. We used Keras (Chollet and others 2015), running on top of Tensorflow (Abadi and others 2015), to build, train and evaluate the model.

Figure 5 shows receiver operating characteristics (ROC) curves for this model evaluated on training and validation data; corresponding confusion matrices are shown in Table 1. ROC curves and confusion matrices are established ways of presenting and assessing the performance of classifiers (Fawcett 2006). An *ROC curve* shows the trade-off between the fraction of correctly classified positive data (true positive rate or TPR) and the corresponding fraction of incorrectly classified negative data (false positive rate or FPR). It is obtained by varying the threshold used to separate positive and negative data, based on the probability of a datum being positive output by the classifier. The ideal classifier would have a TPR of 1.0 for an FPR of 0.0, with the corresponding ROC curve following the left and top axis of the ROC plot. A classifier that randomly picks its output according to the fraction of positive and negative data (known as the base-rate or no-information-rate) would have an expected ROC curve (in the case of infinite data) in the shape of a 45° line from the lower-left to the top-right corner. An ROC curve can be summarised in a single scalar by computing the area under the curve (AUC). The ideal classifier would have a AUC of 1.0 whereas a base-rate classifier would have an expected AUC of 0.5. A *confusion matrix* provides an alternative, tabular summary of the performance of a classifier. Rows and columns in the table correspond to predicted and true classes, respectively. The entry in the  $m^{\text{th}}$  row and  $n^{\text{th}}$  column indicate the number of data of class  $n$  that were classified as class  $m$ . The ideal classifier would thus only have non-zero entries along the main diagonal of the table. Dividing a diagonal element by the sum of the elements in the corresponding column yields a quantity known as *recall* for the corresponding class, which equals the TPR. Dividing the same element by the sum of the elements in the corresponding row yields what is known as the *precision* for the class. Dividing the sum of all the diagonal elements by the sum of all elements in the table gives the average accuracy.

From Table 1, we see that the average accuracy on the validation data is >95%, but predictions are more accurate on BIs than on digital photographs. We also used the trained CNN to generate predicted labels for the unlabelled images; while the lack of actual labels meant that no quantitative evaluation could be done, from visual inspection of some of the images and their predicted labels, it seemed the error rate would be somewhat higher. This is not unexpected, since the unlabelled images deviated from the labelled images in certain ways (e.g. the imaging device used), so the CNN has to extrapolate from the training data to a certain degree to generate the predicted labels. This does suggest that manually labelling of at least some of the unlabelled data could improve the generalisation capabilities of the model.

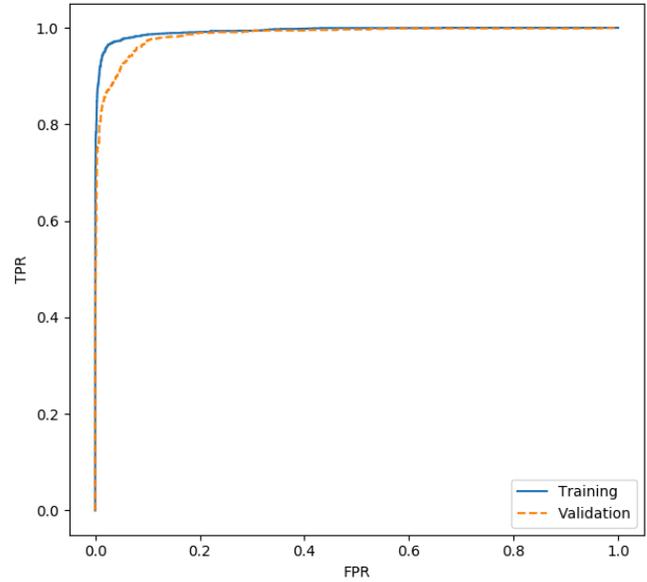


Figure 5 ROC curves for the CNN separating BIs and digital photographs. The curve corresponding to the training data (solid) has an AUC of 0.994 while the curve corresponding to the validation data (dashed) has an AUC of 0.985.

Table 1 Confusion matrices for the CNN separating BIs and digital Photographs for training and validation data. The bottom row shows the distribution of BIs and digital Photographs across training and validation data.

	Training		Validation	
	BI	Ph	BI	Ph
BI	8480	105	3445	114
Ph	158	2035	82	762
Total	8638	2140	3527	876

Attempts were also made to retrain the final, fully connected layer of an instance of the VGG16 deep CNN (Simonyan and Zisserman 2014) on this task. However, these attempts failed to produce a model that was significantly better than the no-information-rate (~80.0%, the fraction of borescope images in the data). This was a somewhat surprising result, but only a limited amount of effort was spent experimenting with this model, so it cannot be regarded as conclusive. At the same time, it may well be that the VGG16 model, which was developed to distinguish between 1000 of real world object categories may not provide useful features to separate BIs and digital camera photographs. It turned out to be more useful, however, in the task we tackled next.

## 4.2. Engine Part Recognition

The second task we considered was the detection of engine parts in BIs from the hot sections of the engine (combustor and high-pressure turbine); this section of the engine is exposed to the highest level of stress in terms of temperatures

	Training	Validation
Mixer	353	145
Combustor	937	385
Fuel Nozzle	836	342
HPT Blade	758	310
Other	466	190

Table 2 The distribution of part recognition data across training and validation sets.

and pressures and is therefore subject to borescope inspections more frequently than other sections.

For this task, automatic labelling was not a viable option, due to the limitations of the existing labelling information, and so we resorted to manually labelling 7098 images. However, existing labels, explicit and implicit (e.g. through file names), were used to aid the manual labelling. 2376 images were found irrelevant or too difficult to label with sufficient certainty and were hence excluded. Each image was labelled with one of the following labels:

- Mixer
- Combustor
- Fuel Nozzle (an example is shown in the left image of Figure 4)
- High-Pressure Turbine Blade (HPT Blade; an example is shown in the right image of Figure 4)
- Other

The first four categories correspond to parts in the hot section of the engine whereas the last is a catch-all category for images of other parts of the engine, not necessarily from the hot section. It should be noted that these categories are not completely mutually exclusive. Fuel nozzles feed into the combustor and so BIs of fuel nozzles will inevitably also contain part of the combustor; the labelling policy used was to label an image as Fuel Nozzle whenever one or more fuel nozzles were clearly visible in the image while all other images from the combustor were labelled Combustor. The data was divided into training and validation sets in the same way as described in the previous section, yielding the division shown in Table 2.

As engine part recognition is a specific object recognition task, we took the approach of retraining the final, fully connected layer of an instance of the VGG16 deep CNN (Simonyan and Zisserman 2014). In practice, this meant propagating training and validation images through a truncated version of the VGG16 model and storing the outputs of the last layer before the final, fully connected layer. This stored data was then used to train a conventional neural network, similar to the one illustrated in Figure 2 with 384 hidden units and 5 softmax output units. We used a drop-out rate

		Mixer	Combustor	Fuel Nozzle	HPT Blade	Other	Precision
CNN—Training	Mixer	<b>323</b>	0	1	1	1	99%
	Combustor	2	<b>924</b>	25	17	1	95%
	Fuel Nozzle	13	5	<b>803</b>	0	0	98%
	HPT Blade	11	8	5	<b>714</b>	69	88%
	Other	4	0	2	26	<b>395</b>	93%
	<i>Recall</i>	92%	99%	96%	94%	85%	
GLM—Training	Mixer	<b>200</b>	8	39	47	50	58%
	Combustor	17	<b>749</b>	114	143	64	69%
	Fuel Nozzle	72	85	<b>565</b>	102	125	60%
	HPT Blade	30	76	83	<b>421</b>	95	60%
	Other	34	19	35	45	<b>132</b>	50%
	<i>Recall</i>	57%	80%	68%	56%	28%	
CNN—Validation	Mixer	<b>57</b>	0	2	0	5	89%
	Combustor	11	<b>349</b>	26	34	10	81%
	Fuel Nozzle	42	15	<b>307</b>	5	7	82%
	HPT Blade	21	21	4	<b>238</b>	69	67%
	Other	14	0	3	33	<b>99</b>	66%
	<i>Recall</i>	39%	91%	90%	77%	52%	
GLM—Validation	Mixer	<b>38</b>	8	10	23	21	38%
	Combustor	15	<b>270</b>	89	96	17	55%
	Fuel Nozzle	29	53	<b>185</b>	40	56	51%
	HPT Blade	30	46	45	<b>142</b>	52	45%
	Other	33	8	13	9	<b>44</b>	41%
	<i>Recall</i>	26%	70%	54%	46%	23%	

Table 3 Confusion matrices for engine part recognition.

of 0.5 to reduce the risk of overfitting. We also trained a simple generalised linear model (GLM) (Bishop 2006) that used spherical Gaussian basis functions centred on randomly chosen images from the training data. While this is clearly too simple a model for such complex task, it is useful as a straw-man model for gauging the performance of the CNN. For both models, values for the parameters controlling the model complexity were determined experimentally using the performance on the validation data.

Table 3 contains confusion matrices, with corresponding precision and recall figures, for both models on training and validation data. From these, it is clear that the CNN does better than the GLM, which is not surprising given that the GLM is not able to exploit the spatial structure of the image data in the same way as the CNN. Comparing to the no-information-rate of 28% (Combustor), both demonstrate better accuracy, CNN at 77% and GLM at 49%, evaluated on the validation data. For both models, there is a difference in the perfor-

mance on training and validation data, which raises the question whether there is a degree of overfitting to the training data by the models. This possibility is certainly there, and future work will investigate this further using more training data and/or more elaborate validation schemes.

## 5. CONCLUSION AND DISCUSSION

The primary objective of this effort is to enable borescope imaging to become an integral part of the engine's digital thread, through in-situ, automated measurement of key component condition. In this paper we have described the initial steps towards achieving this aim by the application of deep CNNs to the analysis of borescope images. We have shown that it is possible to discriminate between borescope images and 'standard' digital camera photographs, which was necessary to rationalise the historical data set under investigation. Secondly, we have demonstrated the feasibility of engine part identification from individual borescope images, for certain hot section components.

In the short term, there are several directions along which this work will continue. An obvious step is to expand the current effort using historical data from other GE Aviation engine lines, for some of which there are considerably more data available. This will not only enable further verification of our current approach but should also allow recognition to be expanded to additional, and possibly more detailed levels of parts. Whilst the current approach will detect whether an engine part is present somewhere in an image, a significant advancement would be if it could also indicate where in the image the part has been detected. Here there are different possible approaches, ranging from providing a bounding box around the part, to pixel level segmentation. The latter approach has already been successfully applied using fully convolutional neural networks to segment regions of degraded thermal barrier coating (TBC) in images of jet engine turbine blades (Bian, Lim and Zhou 2016). However, they were using images of blades that had been removed from the engine and mounted in a dedicated imaging rig (i.e. not BIs). If similar results could be obtained from BIs collected in the field, it would pave the way for more advanced analysis and use of BI data, allowing it to be integrated in a digital approach to engine monitoring, maintenance and management. Extracting extent and severity of TBC degradation, for example, provides an indicator of component condition that can readily be understood in engineering terms and, if necessary, acted upon. BIs also provide snapshots of engine condition over its lifetime and the information contained could be used, assuming it can be extracted in a suitable form, in conjunction with information from other sources, as validation points for 'digital twin' models of engines (Biba 2017). This would allow

an ongoing 'calibration' of physics models with real data, which we can then adapt and learn from.

## ACKNOWLEDGEMENT

Many people within GE Aviation have supported this work with helpful discussions and suggestions; we are especially grateful to Dr Mani Abedini, Dr Peter Knight and John Simmans. This work was funded as the winner of the GE Aviation 2016 Open Innovation Challenge. The illustrations in Figure 2<sup>2</sup> and Figure 3<sup>3</sup> were obtained from Wikimedia Commons under the Creative Commons Share Alike 3.0/4.0 licenses (see footnotes for details).

## REFERENCES

- Abadi, M., and others. 2015. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Google LLC, 2015. [www.tensorflow.org](http://www.tensorflow.org).
- Bengio, Y., P. Lamblin, D. Popovici, and H. Larochelle. 2007. "Greedy Layer-Wise Training of Deep Networks." *Advances in Neural Information Processing Systems 19 (NIPS'06)*. MIT Press. 153-160.
- Bian, X., S. N. Lim, and N. Zhou. 2016. "Multiscale fully convolutional network with application to industrial inspection." *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE.
- Biba, E. 2017. "The jet engines with 'digital twins'." *BBC*. February 17. Accessed April 18, 2018. <http://www.bbc.com/autos/story/20170214-how-jet-engines-are-made>.
- Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Springer.
- Chollet, F, and others. 2015. *Keras*. Accessed 2018. <https://keras.io/>.
- Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. "ImageNet: A Large-Scale Hierarchical Image Database." *CVPR09*. <http://www.image-net.org/>.
- Fawcett, T. 2006. "An introduction to ROC analysis." *Pattern Recognition Letters* (Elsevier) 27: 861–874.
- Hinton, G. E., S. Osindero, and Y. Teh. 2006. "A fast learning algorithm for deep belief nets." *Neural Computation* 18: 1527-1554.
- Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2012. "ImageNet Classification with Deep Convolutional Neural Networks." *Advances in Neural Information Processing Systems 25*. MIT Press, Cambridge, MA.
- LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. 1989. "Backpropagation Applied to Handwritten Zip Code Recognition." *Neural Computation* 1 (4): 541-551.

<sup>2</sup> Colored neural network uk.svg, contributed by Glosser.ca, [https://commons.wikimedia.org/wiki/File:Colored\\_neural\\_network\\_uk.svg](https://commons.wikimedia.org/wiki/File:Colored_neural_network_uk.svg).

<sup>3</sup> Typical cnn.png, contributed by Aphex34, [https://commons.wikimedia.org/wiki/File:Typical\\_cnn.png](https://commons.wikimedia.org/wiki/File:Typical_cnn.png).

- LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner. 1998. "Gradient-Based Learning Applied to Document Recognition." *Proceedings of the IEEE* 86 (11): 2278-2324.
- Simonyan, K., and A. Zisserman. 2014. "Very Deep Convolutional Networks for Large-Scale Image Recognition." *CoRR*.
- Wikipedia contributors. n.d. "Exif." *Wikipedia*. Accessed April 18, 2018. <https://en.wikipedia.org/w/index.php?title=Exif&oldid=836272129>.
- Yosinski, J., J. Clune, Y. Bengio, and H. Lipson. 2014. "How transferable are features in deep neural networks?" *Advances in Neural Information Processing Systems 27 (NIPS'14)*. NIPS Foundation, 2014.

## BIOGRAPHIES



**Markus Svensén** works as Senior Staff Data Scientist at GE Aviation Digital. He has a MSc and a PhD in Computer Science. During his career, he has worked and published on a range of, primarily applied, machine learning problems. Before joining GE, he worked at Microsoft Research Cambridge (UK) and at

the Max-Planck-Institute of Cognitive Neuroscience in Leipzig, Germany.



**David Hardwick** has worked in the technical software industry for 35 years. He has 20 years' experience in data acquisition and control systems and now works for GE Aviation on advanced machine learning data analytics. David has degrees in Computer Science (MSc) and Physics (BSc) and is a member of

the IoP and BAA.



**Honor Powrie** has worked in the Aviation industry for more than 25 years, including 20+ years' in Machinery Health Monitoring and Management. Since 2007 Honor has worked for General Electric (GE) Aviation. In her current role as Director of Data and Analytics, Honor manages GE Aviation's

only UK-based Data Science group, which delivers innovative solutions for monitoring and managing GE's extensive and world-renowned commercial engines fleet. Honor is also a Visiting Professor in the Faculty of Engineering and the Environment at the University of Southampton and in 2017, she was awarded a Royal Academy of Engineering Visiting Professorship in Data and Analytics, Asset Condition Monitoring and Management.