Deep Neural Networks for Visual Bridge Inspections and Defect Visualisation in Civil Engineering

Julia Bush[1]*, Tadeo Corradi[2], Jelena Ninić[1], Georgia Thermou[1], John Bennetts[3]
[1]University of Nottingham, UK; [2]Mind Foundry, UK; [3]WSP, UK
*Julia.Bush@nottingham.ac.uk (Lead author email address)

Abstract. Ageing infrastructure is a global concern, and current structural health monitoring practices are coming under review. With a view to streamline the visual bridge inspection process, we assess the classification performance of two Deep Neural Networks, VGG16 and MobileNet, on a challenging dataset of over 70,000 unprocessed bridge inspection images of three defect categories: corrosion, crack, and spalling. Grad-CAM "heatmap" visualisations on VGG16 predictions provide a coarse localisation of the defect region and some insight into the functioning of the network. Similar performance is attained on MobileNet, for applications where speed or computational cost is a consideration. We conclude that with further optimisation this approach could have an application in automated defect tagging.

1. Introduction

Civil engineering infrastructure asset owners such as Highways England and Network Rail in the UK require asset condition information for several purposes: planning maintenance interventions, assessments of load capacity, exploring trends, leaving audit trails and measuring contracted services (Bennetts et al., 2018). Current practice in bridge inspection produces data with significant uncertainty, and the metrics used in defect description are not optimal for life-cycle analysis of deterioration and cost.

The primary source of bridge condition data are visual bridge inspections (Bennetts et al., 2016). Since these are numerous, costly, and may require disruption to the transport network, it is imperative that the data collected be of high quality and suitable for analysis to obtain the information required. As the value of data is increasingly recognised, data collection and recording processes are coming under review to enable meaningful condition information to be derived and represented, and to then be adequately exchanged between all parties involved.

For the purposes of this paper, only visible defects will be considered, mainly: cracks, corrosion and spalling. The current practice for monitoring defects which have no visible signs (such as chloride migration, carbonation, alkali-silica reaction) is to carry out appropriate intrusive testing. This is planned and managed separately from visual inspections and is beyond the scope of this paper.

2. Background: Computer Vision and Deep Neural Networks for Bridge Inspections

Koch et al. (2015) reviewed Computer Vision based defect detection and condition assessment of concrete and asphalt infrastructure. It was concluded that at the time it was not possible to detect, measure, assess and document defects to provide an integrated and comprehensive approach for inspections. More recently, Azimi et al. (2020) have reviewed deep learning approaches in structural health monitoring more generally. Among the challenges identified in the literature to date, the following two emerge as the most pertinent:

- the lack of standardisation in identifying relevant defect parameters to comprehensively represent defect information, and

- the absence of publicly available large datasets to leverage supervised learning methods for the robust detection and classification of several infrastructure defect types.

This paper is intended to respond to both above issues, with a long-term view towards an automated end-to-end digital bridge inspection process, and eventual digital twinning of infrastructure assets.

Liang (2019) provide a successful precedent for use of VGG16 (Simonyan and Zisserman, 2015) initialised on ImageNet (Russakovsky et al., 2015) for bridge damage classification. Class Activation Mapping (Selvaraju, 2017) has been applied to VGG16 initialised on ImageNet by Perez et al. (2019) to classify and locate building defects. In this paper, we adopt a similar approach to treat images of bridge defects.

3. Methodology

3.1 Image Data

A sample of over 200,000 images of bridge defects was obtained from Highways England for the work presented in this paper. In contrast to many publications to date, the number of images stated here refers to distinct photographs of bridge defects taken on site, which have not been cut up to generate multiple images from a single photograph. Neither have they been cropped to place the object of interest (the defect region in our case) in a prominent position within the image, which would require manual processing of a similar level of labour intensity as bounding box annotations.

The scenes have complex backgrounds and both object position and scale vary (see Figure 2 in Section 5). This, along with other inconsistencies (in lighting and weather conditions, camera, angle, resolution, shadows, background and foreground noise, surface markings, weather-induced surface wetness, irrelevant surface alterations such as small holes or stains) makes this dataset an important step towards developing a benchmark dataset for Computer Vision methods applied to bridge defects.

For any neural network architecture to be usable in real on-site conditions, it must be robust against the noise and variations (as described above) in the images it receives for making predictions. For those who seek to add value to the Civil Engineering industry, therefore, it is imperative to seek methods which move away from the clean laboratory image data and towards accommodating the real complex noisy image data encountered by bridge inspectors on site.

To the best of the authors' knowledge, this is the first time a dataset of this size and complexity has been examined. Inevitably, even an optimally designed methodology will require such a volume of data which is sufficient to overcome the noise. Given the complexity of the features which are sought to be learned, we expect dataset sizes to grow beyond what can be reasonably hand-crafted, even for the simplest case of image-level labels only. To pave the way for handling such datasets, the approach presented in this paper is focused on removing as much human input from data pre-processing as possible.

3.2 Data Set

The dataset consists of 200,852 photographs, tagged with one of a total of 161 possible defect types. Direct classification on the 161 labels is both undesirable and unlikely to succeed, as the classes are heavily imbalanced and, in many cases, represent overlapping concepts. Therefore

we decided to create supergroups comprising several classes and selected three of them as a first attempt at an already challenging classification problem (Table 1).

Table 1: 3-class supergroup dataset to train VGG16 and MobileNet classifiers. No data augmentation.

Defect class	Number of images	Data volume, GB
Corrosion	23,474	4.1
Crack	26,775	5.2
Spalling	19,837	3.3
Total	*70 086*	*12.6*

The chosen supergroups represent defect types that ultimately are of highest interest in industry: corrosion, crack and spalling. For the remaining classes (excluding corrosion, crack and spalling), Figure 1 gives an indication of the numbers of images per class, for those classes which contain 1,000 or more images.

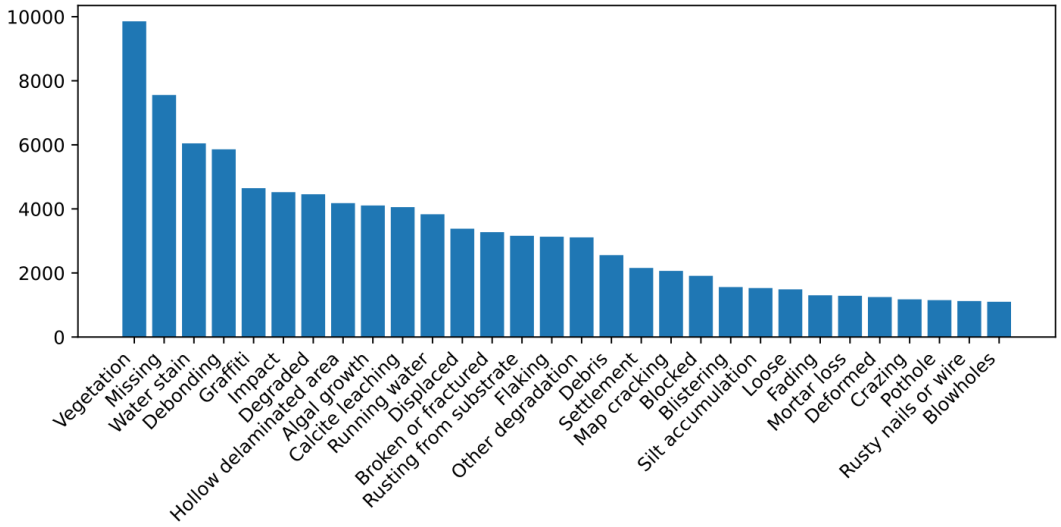


Figure 1: Number of images of other defect types

3.3 Neural Network Architecture

The VGG16 (Simonyan and Zisserman, 2015) was used following the example of previous applications of this architecture to building and bridge defects. In the spirit of searching for the simplest solution which produces predictions of sufficient complexity and accuracy, we also used MobileNet (Howard et al., 2017). The complexity and performance indicators of VGG16 and MobileNet are compared in Table 2, where the top-1 and top-5 accuracy refer to the model's performance on the benchmark ImageNet (Russakovsky, 2015) validation dataset (not on the dataset presented in this paper). Depth refers to the topological depth of the network, and includes activation layers, batch normalisation layers etc.

Table 2: Comparison of complexity and performance of VGG16 and MobileNet.

Architecture	Size, MB	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
VGG16	528	0.713	0.901	138,357,544	23
MobileNet	16	0.704	0.895	4,253,864	88

3.4 Localisation

Selvaraju et al. (2017) observe that convolutional layers naturally retain spatial information which is lost in fully-connected layers, so the last convolutional layers are expected to have the best compromise between high-level semantics and detailed spatial information. Their approach, Gradient-weighted Class Activation Mapping (Grad-CAM), uses the gradients of any target concept (say "corrosion" in a bridge defect classifier) flowing into the final convolutional layer to produce a coarse localisation map highlighting the important regions in the image for predicting the concept.

As will be seen in Section 5, this coarse localisation map can provide clues as to the functioning of the trained neural network, allowing us to peek into the model which is traditionally considered "black box". Furthermore, Selvaraju et al. (2017) provide successful examples of Grad-CAM being used as seed for weakly supervised segmentation, an approach which the authors intend to apply to bridge defect images in later work.

4. Implementation

Implementation in Python 3.7 using Keras high-level neural network library, which is in turn built on TensorFlow 2.3.0 machine learning library, using a CUDA[1] 10.1 backend and CUDNN[2] 7. During network training, the dataset was randomly split into 80% training and 20% validation subsets.

4.1 VGG16

The VGG16 was trained using the standard approach of first training the classifier head only, and consequently unfreezing all layers (initialised with ImageNet weights). The classifier head consisted of four layers, namely, flatten, dense, dropout, dense, comprising 3,232,161 trainable parameters.

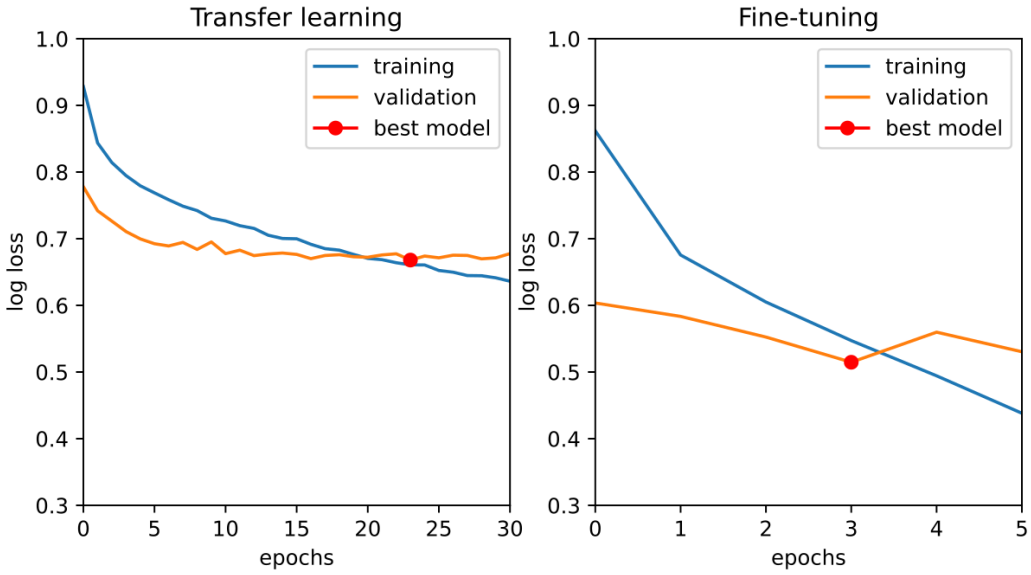
Firstly, we used the full dataset of 200,852 images belonging to 161 classes as per the original defect type image labels. As expected, this yielded low accuracy (Table 3). Secondly, the dominant classes were grouped into a three-class (corrosion, crack, spalling) dataset, transfer learned for 5 epochs, and fine-tuned for 10 epochs (Figure 2). The latter achieves a considerable validation accuracy of 0.81. Section 5 provides a discussion of possible sources of errors.

Table 3: VGG16 training and accuracy.

Training mode	Dataset	Epochs	Validation accuracy
Transfer learning	Raw, 161 classes	20	0.23
Fine tuning	Raw, 161 classes	10	0.31
Transfer learning	Selective, 3 classes	5	0.71
Fine tuning	Selective, 3 classes	10	0.81

[1] https://developer.nvidia.com/cuda-toolkit

[2] https://docs.nvidia.com/deeplearning/cudnn/archives/cudnn_765/cudnn-release-notes/index.html



(a) (b)

Figure 2: VGG16 learning curves for the 3-class dataset (train loss in blue and validation loss in orange). (a) transfer learning for 5 epochs; (b) fine-tuning for 10 epochs

In Table 4 True Positives (along the diagonal in blue) indicate the numbers of correct predictions for each of the three classes, corrosion, crack, and spalling. False Positives (upper right in orange) tell us, for example, that 391 images whose true classification is "crack" were predicted to be "corrosion". An example of False Negatives (lower left in pink): 226 whose true classification is "corrosion" and whose predicted classification was "crack".

Table 4: VGG16 confusion matrix.

True class:	***corrosion***	***crack***	***spalling***
Prediction:			
corrosion	4199	391	648
crack	226	4387	787
spalling	269	577	2532

Accuracy (the total number of correct predictions divided by the total number of predictions made) alone can be an overly optimistic indicator of network performance. Table 5 provides a summary of more robust machine learning classification metrics. It is desirable to attain high precision, while low recall is acceptable, in applications where it is not important to identify all positive instances, but it is important that when an instance is identified as positive, this is with high certainty. High recall, on the other hand, corresponds to capturing the maximum number of true positives, and false positives are well tolerated (low precision). Ideally we would like both precision and recall to be high, and the F1 score combines both into a single metric. Weighted average can be very different from macro average if the network is simply guessing by predicting the majority class(es). In our case, all values are similar to the accuracy score, confirming that this is a valid indicator of performance. "Support" is simply the number of images of a given class which were used for validation.

Table 5: VGG16 classification metrics.

	Precision	Recall	F1 score	Support
corrosion	0.80	0.89	0.85	4,694
crack	0.81	0.82	0.82	5,355
spalling	0.75	0.64	0.69	3,967
accuracy			0.79	14,016
macro average	0.79	0.78	0.78	14,016
weighted average	0.79	0.79	0.79	14,016

4.2 MobileNet

MobileNet was designed as an attempt to reduce the intensive computational burden of earlier deep network architectures. It comprises a large number of narrow layers, and can be tuned to achieve a compromise between predictive performance and speed. Its name stems from its intended use on mobile devices, on which it is often important to create a fast prediction without heavy power consumption.

Where VGG16 provides an indication of the ultimate potential of a state-of-the-art neural network for the purpose of bridge defect classification, MobileNet gives a realistic prospect of what could be achievable in an eventual deployed application on a portable mobile device. For the purpose of transfer learning, we remove the top fully-connected layer and replace it with a simple network initialised with random weights (average pooling followed by four dense layers, comprising 164,611 trainable parameters).

Figure 3 shows the train and validation loss at each epoch. Tables 6 and 7 show the performance statistics after 15 epochs (5 for transfer learning). The performance in almost every metric is below that of VGG. However this is attained using considerably less computing power.

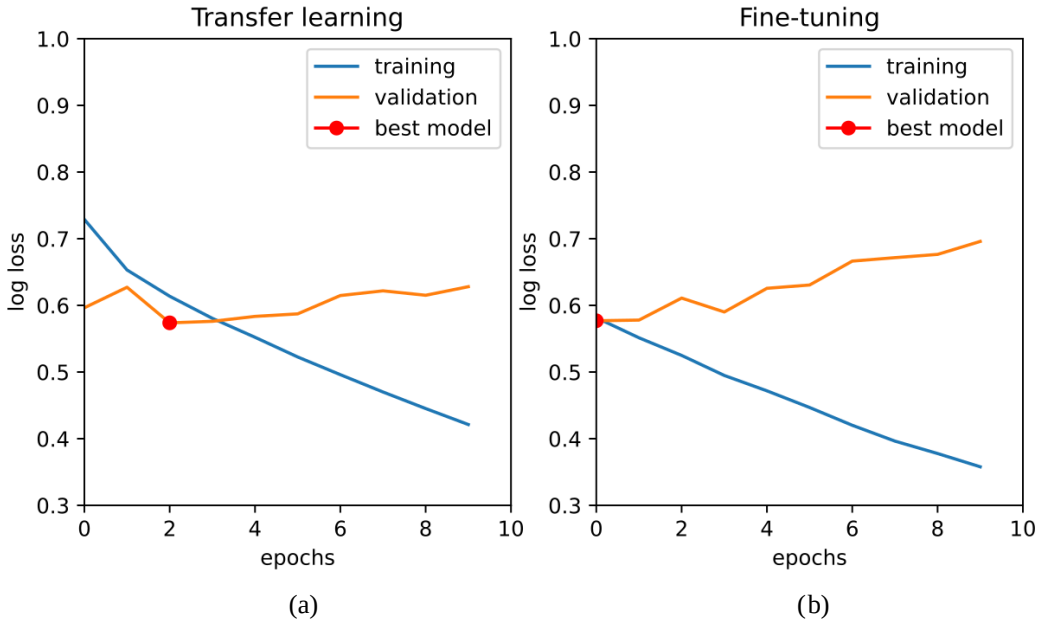


Figure 3: MobileNet learning curves for the 3-class dataset, during transfer learning (a) and during fine-tuning (b).

Therefore, while we focus primarily on VGG, we consider that smaller architectures such as MobileNet have high potential, particularly for problems in which latency or power consumption are limiting factors.

Table 6: MobileNet confusion matrix.

True class:	***corrosion***	***crack***	***spalling***
Prediction:			
corrosion	3718	315	411
crack	446	4406	1215
spalling	511	446	2313

Table 7: MobileNet classification metrics.

	Precision	**Recall**	**F1 score**	**Support**
corrosion	0.84	0.80	0.82	4,675
crack	0.73	0.83	0.83	5,315
spalling	0.68	0.59	0.63	3,939
accuracy			0.75	13,929
macro average	0.75	0.74	0.74	13,929
weighted average	0.75	0.75	0.75	13,929

5. Results

While classification accuracy and other metrics stated in Section 4 give some positive indication of the neural network performance, a more informative discussion of results lies in close inspection of classification predictions and their associated Grad-CAM visualisations. All examples given here have been drawn from the validation subset of the VGG16 transfer learned for 5 epochs and fine-tuned for 10 epochs on the 3-class dataset.

Unlike semantic segmentation, which requires a class label for every pixel in every image for training, classification requires only one label for the entire image. By extracting features common to images belonging to the same class, the trained network can not only make class predictions for a given image, but also give some indication of which pixels are more or less pertinent to that prediction. In Figure 4(a) the image is correctly classified as belonging to the "corrosion" class, and the main corroded region is correctly located. This remains true for scenes with complex backgrounds, such as Figure 4(b), where the network largely ignores the irrelevant buildings, trees, fences etc.

Many images in the dataset contain signs of multiple defects, presenting a challenge for prediction accuracy assessment. Grad-CAM visualisations in Figure 5 illustrate that while multiple defect features may be correctly identified, the image has a single "correct" class against which to score the prediction.

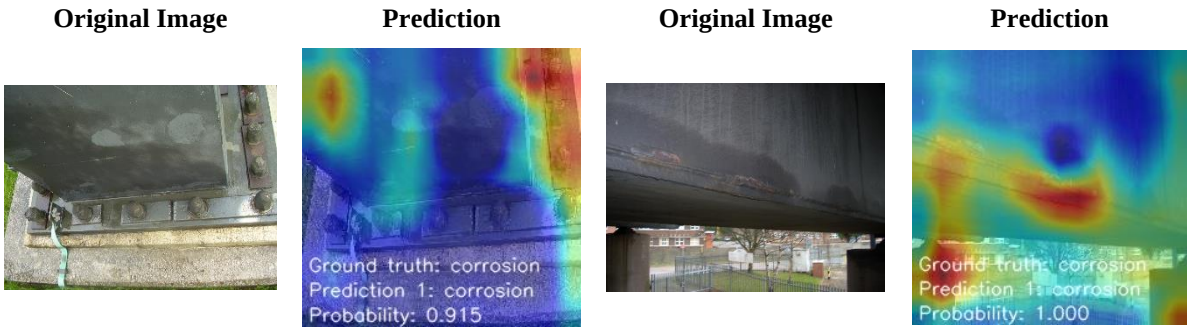


Figure 4: VGG16 trained on corrosion, crack, and spalling classes. Grad-CAM visualisations reveal those regions of the image which have been the most pertinent for the classification process.

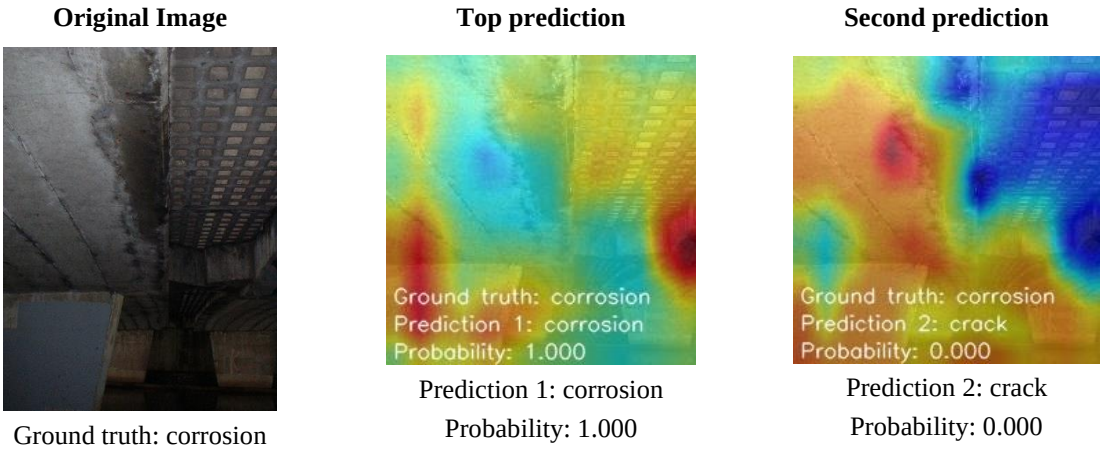


Figure 5: VGG16 trained on corrosion, crack, and spalling classes. Signs of multiple defects on the same image are correctly located.

We gain further insight into the inner workings of the network by observing the examples given in Figure 6. The top row contains examples of correctly predicted image classes, however the heatmaps clearly show that the classifier relied on component features (namely the geometry of the bolt and the steel connection) rather than the defect features (such as the colour and texture typical of corrosion) to make its prediction. This can easily happen where there is positive correlation between a component type and a defect type (for example, if the dataset contains many images of corroded bolts, the network will tend to classify any image containing any bolt as "corrosion" without any signs of corrosion itself). This type of error can be overcome by balancing the dataset (for example, by including images of non-corroded bolts).

Another likely source of errors is poor correspondence between the image scene and the ground truth label. Taking the examples along the bottom row of Figure 5, we see that the network is correctly identifying the crack and spalling features and hence predicting "crack" and "spalling". However this prediction will be scored as erroneous during validation since the ground truth labels are "corrosion" in both cases. This situation may arise when the inspector is not able to gain better access to the defect and has to take the photograph from an unsuitable position, or when the ground truth classification is given according to the underlying causes rather than the visual cues (as per the bottom right example in Figure 6). Moreover, the ground truth classification may sometimes be simply incorrect, for example, due to human error.

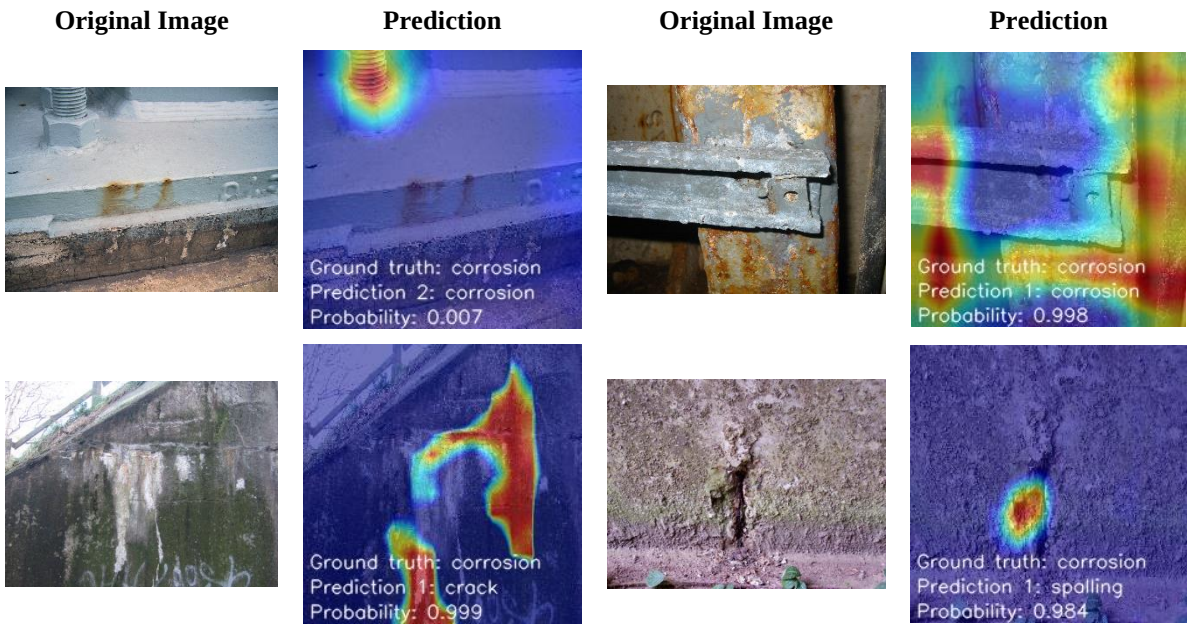


Figure 6: VGG16 trained on corrosion, crack, and spalling classes. Top row: images are correctly classified using incorrect features. Bottom row: defect features are correctly identified, however the predictions are scored as "incorrect" due to poor ground truth labels.

6. Conclusions and Future Work

In this paper we presented an application of deep learning to bridge defect image classification using big data acquired from bridge inspections in the UK over the past 20 years. Established machine learning metrics were used for rigorous performance assessment. The achieved accuracy is significant, however further optimisation of network architecture and training methodology remain possible.

Finally, we provide a reference comparison to a smaller neural network (MobileNet), demonstrating that similar performance it attainable, where speed or computational cost is a consideration.

The following improvements are recommended:

- Where a strong positive correlation between a structural component type and defect type exists, include non-defect (normal) component images in the dataset to prevent classification by component features rather than defect features.
- Create partially annotated datasets to guide the feature-learning process.
- Set aside a test dataset of images which the neural network sees neither during training nor during validation to enable complete network performance assessment.
- Guard against overfitting, for example with regularisation, or dropout.

Another meaningful supergroup could be created of other, smaller, defect classes with strong visual cues (for example, graffiti, vegetation, water-related staining). Since these classes contain relatively few images (around 1,000 per class) compared to the dominant classes of corrosion, crack and spalling, isolating them would create a more balanced dataset.

We conclude that this would be a valid approach in the larger framework of automating selected tasks in the visual bridge inspection process, and could be used as a means of automatic defect

tagging and coarse localisation in a 2D images, which could in turn be extended to a 3D environment.

Acknowledgements

The authors would like to thank Highways England, UK, for the use of bridge defect images and their associated defect types. This research is part of a project funded by the EPSRC, WSP UK and Highways England, and would not have been possible without their support.

References

Azimi, M., Eslamlou, A.D., Pekcan, G. (2020). Data-driven structural health monitoring and damage detection through deep learning: State-of-the-art review. Sensors (Switzerland) 20. doi:10.3390/s20102778.

Bennetts, J., Vardanega, P.J., Taylor, C.A., Denton, S.R. (2016). Bridge data - What do we collect and how do we use it? in: Transforming the Future of Infrastructure through Smarter Information - Proceedings of the International Conference on Smart Infrastructure and Construction, ICSIC2016, ICE Publishing. pp. 531–536. doi:10.1680/tfitsi.61279.531.

Bennetts, J., Webb, G., Denton, S., Vardanega, P.J., Loudon, N. (2018). Quantifying uncertainty in visual inspection data, in: Maintenance, Safety, Risk, Management and Life-Cycle Performance of Bridges - Proceedings of the 9th International Conference on Bridge Maintenance, Safety and Management, IABMAS 2018, CRC Press/Balkema. pp. 2252–2259.doi:10.1201/9781315189390-306.

Howard, Andrew & Zhu, Menglong & Chen, Bo & Kalenichenko, Dmitry & Wang, Weijun & Weyand, Tobias & Andreetto, Marco & Adam, Hartwig. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

Koch, C., Georgieva, K., Kasireddy, V., Akinci, B., Fieguth, P. (2015). A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. Advanced Engineering Informatics 29, 196–210. doi:10.1016/j.aei.2015.01.008.

Liang X. (2019). Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization. Computer-Aided Civil and Infrastructure Engineering 34:415–430. https://doi.org/10.1111/mice.12425.

Perez, H., Tah, J.H., Mosavi, A. (2019). Deep learning for detecting building defects using convolutional neural networks. Sensors (Switzerland) 19.doi:10.3390/s19163556.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang,Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., 2015.ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision 115, 211–252. doi:10.1007/s11263-015-0816-y.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization, in: Proceedings of the IEEE International Conference on Computer Vision, Institute of Electrical and Electronics Engineers Inc..pp. 618–626. doi:10.1109/ICCV.2017.74.

Simonyan, K., Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition, in: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, International Conference on Learning Representations, ICLR.