A comparison of deep learning techniques for corrosion detection*

Tom Bolton[1 [0000−0003−1599−8400]], Julian Bass[1 [0000−0002−0570−7086]], and Tarek Gaber[1 [0000−0003−4065−4191]]

School of Science, Engineering & Environment, University of Salford, Salford M5 4WT, UK

Abstract. Corrosion - degradation in metal structures - is problematic, expensive to rectify, and can be unpredictable in the rate at which it spreads. Traditional preventative maintenance techniques are complemented by human visual inspection, in turn complemented by artificial intelligence vision techniques. The primary objective of this paper was to determine the most accurate deep learning model for use in corrosion detection; to achieve this, we devised an experimental comparison that tested five machine learning algorithms for the detection of corrosion from image data. The deep learning that forms the basis of algorithms used to solve object recognition problems traditionally requires large amounts of training data. As this data requires manual labelling by a person who is expert in the domain of corrosion, it is difficult and expensive to obtain; time and expense that increase considerably as more sophisticated pixel-level annotation is applied. We discovered that high levels of accuracy (98%) can be achieved using deep learning to detect corrosion using samples annotated with simple, image-level labels. We achieved this headline accuracy through the application of transfer learning using models that had been trained on the ImageNet dataset. With many deep learning algorithms to choose from, we systematically determined the most accurate model to use as a basis for further experimentation.

Keywords: Deep learning · Transfer learning · Corrosion.

1 Introduction

In most metals, corrosion is inevitable and manifests as a degradation of a metal's properties due to interaction with its environment [3]. Degradation of metal components caused by corrosion can have untoward and severe consequences, including plant shutdowns, loss of liquid assets due to leakage, reductions in efficiency, and contamination.

In industrial settings, predictive maintenance techniques used to combat the effects of corrosion and other forms of degradation, ranging from visual inspection to advanced signal processing analysis, have been proven more accurate than time-based preventative methods in such safety critical situations as nuclear power plants [9]. However, visual inspections are not failsafe and human

* Supported by the University of Salford and Add Energy.

error is an ongoing problem in maintenance [2]. Signal analysis, whilst more accurate, can not help if a deformed component at fault is non-electrical, and comprises, for example, only a metal plate and several bolts; no real time measurement data is available from such simple components.

This research focuses on detection of corrosion using artificial intelligence (AI)-based machine learning to analyse video footage of components in the field, offering several advantages. Using AI systems could improve accuracy, whilst working consistently and without any reduction in performance due to fatigue, when compared with a visual inspection carried out by a person. A study carried out for the United States' National Nuclear Security Administration, in which the author examined the reliability of visual inspections of precision manufactured parts used in nuclear weapons, found that defective parts were correctly rejected in 85% of cases [17]. This figure, not much higher than the industry average of 80%, was contrasted by the 35% rejection rate of acceptable items by the qualified inspectors; as inspectors became fatigued, confidence fell and the rejection rate increased. It is hoped that the use of novel machine learning techniques could analyse video footage and detect component degradation with an accuracy similar to or even greater than the 93% - 97% values achieved in studies reviewed by Gangsar et al [16] which used signal detection to diagnose worn induction motor components.

As machine learning research intensifies, deep learning algorithms have become more complex; recent projects such as Mask R-CNN [10] and Google's DeepLab [5] have shown great promise. However, training data - images from which the deep models 'learn' - is expensive to obtain and annotate, and often in short supply [18]. Data that has been annotated to a pixel level, the semantic segmentation required by Mask R-CNN and DeepLab, is vastly more time consuming to prepare than a dataset containing samples annotated at an image level.

In order to work towards an AI-driven corrosion detection system, this paper aims to conduct a study that compares five deep convolutional neural networks to determine which is most accurate according to set evaluation metrics. We use a dataset with image-level annotations in order to determine the accuracy levels possible without recourse to semantic segmentation; we also study the effectiveness of transfer learning - the use of algorithms pretrained on non-corrosion datasets - on the accuracy of the models. We ask the following research questions (RQ):

RQ1 - given a set amount of training data annotated with image-level labels, which deep learning algorithm offers the greatest accuracy when used to detect corrosion?

RQ2 - does the application of transfer learning improve the accuracy levels achievable when compared with starting from random weights?

The rest of this paper is organised as follows: Section 2 examines existing research in the field; Section 3 outlines the background of the techniques and methods used in the paper; Section 4 contains the methodology and approaches

used in the experiments; Section 5 shows the results of the experiments, and discussion of the outcomes.

2 Related Work

Bastian et al. proposed a convolutional neural network of their own design that could be used to identify corrosion in water, oil, and gas pipelines [4]. To work towards this goal, the authors assembled a labelled dataset containing a large number of images of pipelines that exhibited varying levels of corrosion. The authors first gathered images by extracting frames from videos of corroded pipework - 60,000 images were collected. To increase this number, the existing images were flipped and rotated to create 'new' images. The resultant dataset contained 1,420,400 samples. The authors' use of a custom deep neural network architecture was designed to address problems with conventional computer vision methods; specifically, that corrosion does not have a definite shape, colour, or pattern. The custom model achieved 98.8% classification accuracy.

A study by Yu et al. used a combination of deep learning algorithms to develop AMCD: "an accurate deep learning-based metallic corrosion detector for MAV-based real-time visual inspection." [22] Using images taken from a micro aerial vehicle (MAV), the authors combined You Only Look Once (YOLO) object detection with a custom convolutional neural network classifier to detect the presence of corrosion. By constructing a novel deep learning algorithm with an acceptable frame rate for use on an MAV, the authors proposed the first use of seperable convolution layers to reduce model parameter numbers whilst retaining accuracy. Using five separate evaluation metrics, the authors' network AMCD achieved an accuracy of 84.96%: greater than other networks. This headline accuracy was achieved using far fewer data than the Bastian study; AMCD used 5625 images containing 27039 annotations; these images were, however, annotated with bounding boxes which decreases the number of samples required to achieve high accuracy, but introduces time and cost into the annotation process.

A number of studies take into consideration the impact that transfer learning has on the authors' experiments [11] [14] [15] [21]. Gopalakrishnan et al. produced a study in which the authors successfully used a convolutional neural network that had been pre-trained on the ImageNet database to detect distress such as cracking in pavements [7].

3 Preliminaries

In order to understand the research that has been performed to date in the domain of machine learning-based corrosion detection, it is necessary to understand the background of the techniques used.

3.1 Deep Learning

The neural networks used in deep learning can trace their roots back to the 1940's and a paper by Pitts et al. in which two neural mechanisms were described that '. . . exhibit recognition of forms' [19]; the authors set out to use a simulation of the human nervous system to solve learning problems in a mathematical fashion. Computational applications of the neural networks that would eventually become what is now recognised as deep learning enjoyed a resurgence in the 1980s and

1990s when Rumelhart et al proposed the back-propagation algorithm [6] which allowed networks to repeatedly adjust their weights to reduce loss.

Deep learning as used in computer vision applications such as object detection, image retrieval, and semantic segmentation, has been studied extensively in recent years due in part to the availability of affordable, high-performance graphical processing units [8]. Deep convolutional neural networks, used to solve the annual ImageNet Large-Scale Visual Recognition Challenge, have evolved from Krizhevsky et al.'s winning algorithm AlexNet in 2012 [1] into complex, multi-layered networks that are capable of performing object recognition to a very high level of accuracy.

Deep learning, in its use of multi-layer neural networks, attempts to learn high-level abstractions and has shown to be highly performant. It is, however, reliant on large quantities of labelled data - i.e., that which has been manually annotated by a person, or 'oracle'. This labelled data is not only expensive to collect, it can be hard to obtain at all in any great quantity in a novel environment [18].

3.2 Transfer Learning

Transfer learning is a technique by which learning in one domain is improved by transferring information from another domain. The process has been likened to examples in the real world, comparing two people who would like to learn to play the piano; one person has no experience of playing music whilst the other has gained musical knowledge through playing the guitar. The person who plays the guitar will be able to learn the piano more efficiently by transferring the musical knowledge they have learned on another instrument [20].

As a baseline for this transfer of knowledge, there are several large, publicly available databases of annotated, categorised images with which researchers in computer vision can train models. The 14 million-sample ImageNet is one such, containing hundreds of thousands of real-life images for each node of a set hierarchy of classifications [12]. The data is free for researchers to use in non-commercial settings.

4 Methodology

The methodology used during experimentation to determine the most accurate model from the five tested is outlined here. The models compared were AlexNet, VGG-16, ResNet-50, Bastian et al.'s Custom Net [4], and ZFNet. In order to train and compare the models, the dataset must first be prepared.

4.1 Dataset Preparation

The dataset used in these experiments consists of 39,600 images of varying sizes, labelled at an image level with four classifications of corrosion severity: none, low, medium, high. The images are varied in size, with both horizontal and vertical dimensions ranging from 200 to 1,000 pixels; the images were obtained by the researchers from video that was filmed in a variety of lighting conditions. The data has been subjected to augmentation: a combination of cropping, skewing, and rotation has been applied to make better use of a limited set of original samples.

As the images in the dataset had been subject to augmentation, random selection was used to organise the samples into training, validation, and test sets; the randomisation reduced the likelihood of similar, consecutive images appearing in a given set of samples, thus increasing variance in the dataset as a whole. Each of the four classes was divided as follows: 8,000 for training, 1,000 for validation, and 900 for testing. To ensure consistency across the experiments with different algorithms, the same randomly distributed data was used each time.

4.2 Evaluation Metrics

To assess the effectiveness of each experiment, accuracy will be measured with the following evaluation metrics. The three metrics use a combination of true positives (TP), false positives (FP), and false negatives (FN) to calculate precision, recall, and F1 score.

Precision indicates the ratio of correctly predicted positives against the total number of positives for a given label.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Recall is the ratio of correctly predicted positives against all positives for a given label.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

F1 Score is given as the harmonic mean of precision and recall for a given label.

$$F1\ Score = 2.\frac{Precision.Recall}{Precision + Recall} \tag{3}$$

4.3 Training the Models

Each of the models was implemented in code, using Python 3.9, TensorFlow 2.6, and Keras 2.6. The computer used for training was equipped with an 11th generation Intel Core i7 CPU, 32GB of RAM, and an RTX3080 GPU with 16GB of VRAM.

5 Results and Discussion

In this section we train five convolutional neural networks with the dataset outlined in Section 4.1 and compare the results. Firstly, all five models were trained starting from randomly initialised weights; all weights in each model were trained. Secondly, two of the models (ResNet and VGG-16) were trained using weights learned from the ImageNet dataset; the transfer learning from the pretraining was retained by freezing the convolutional layers of the models' feature extractors such that only the fully-connected classifier layers were trained.

Each round of training was carried out over 20 epochs; for each model, six different learning rates were tested: 0.1, 0.01, 0.001, 0.0001, 0.00001, and 0.000001. An Adam optimser was used with beta_1 (first moment exponential decay rate) and beta_2 (second moment exponential decay rate) set to their default values

of 0.9 and 0.999 respectively. The batch size was set to 32 and each epoch run for 1,000 steps to account for all 32,000 samples in the training dataset. The training time per epoch varied depending on the model, between two and four minutes.

5.1 Results

These experiments combined yielded 840 saved models. Each of these was evaluated using the test data from the dataset - 3,600 images, 900 from each class. Table 1 shows the best results - the highest precision, recall, and F1 scores - for each model after 20 epochs. Not all models achieved their highest scores at the same learning rate; as such, the learning rate at which the score was achieved is also shown in Table 1.

Table 1. Comparison of all networks tested

	Precision	**Recall**	**F1 Score**	**Learning Rate**
ResNet 50 (ImageNet)	0.980	0.980	0.980	0.0001
VGG-16 (ImageNet)	0.963	0.963	0.963	0.0001
ZFNet (random weights)	0.959	0.959	0.959	0.0001
VGG-16 (random weights)	0.949	0.949	0.949	0.00001
ResNet 50 (random weights)	0.937	0.937	0.937	0.0001
Bastian (random weights)	0.930	0.930	0.930	0.0001
AlexNet (random weights)	0.923	0.923	0.923	0.00001

As a further experiment, the most accurate model (ResNet 50, ImageNet pretrained) was trained again twice; this further training used a learning rate of 0.0001, a batch size of 32, and was carried out over 20 epochs. For each training, the beta_1 (first moment exponential decay rate) and beta_2 (second moment exponential decay rate) hyperparameters of the Adam optimiser were reduced. The results of the training were observed and can be seen in Table 2.

Table 2. Comparison of ResNet 50 training using different beta_1 and beta_2 settings

	Precision	**Recall**	**F1 Score**	**Learning Rate**
beta_1=0.9, beta_2=0.999	0.980	0.980	0.980	0.0001
beta_1=0.6, beta_2=0.6	0.974	0.973	0.973	0.0001
beta_1=0.3, beta_2=0.3	0.964	0.963	0.963	0.0001

5.2 Discussion

The complexity of the models used in this study varies widely from AlexNet's eight layers to ResNet with 51 layers; the computational load when training each model changes accordingly. However, our study showed that good results were achievable using these models trained over only 20 epochs, with a consumer-

grade laptop able to complete one epoch in a few minutes for even the most complex model.

From the results it can be seen that very high levels of accuracy can be achieved using deep learning to detect corrosion; a headline F1 score of 0.98 (98%) achieved using ResNet 50 is greater than the highest accuracy level achieved in studies reviewed by Gangsar et al [16] which used signal detection to diagnose worn induction motor components.

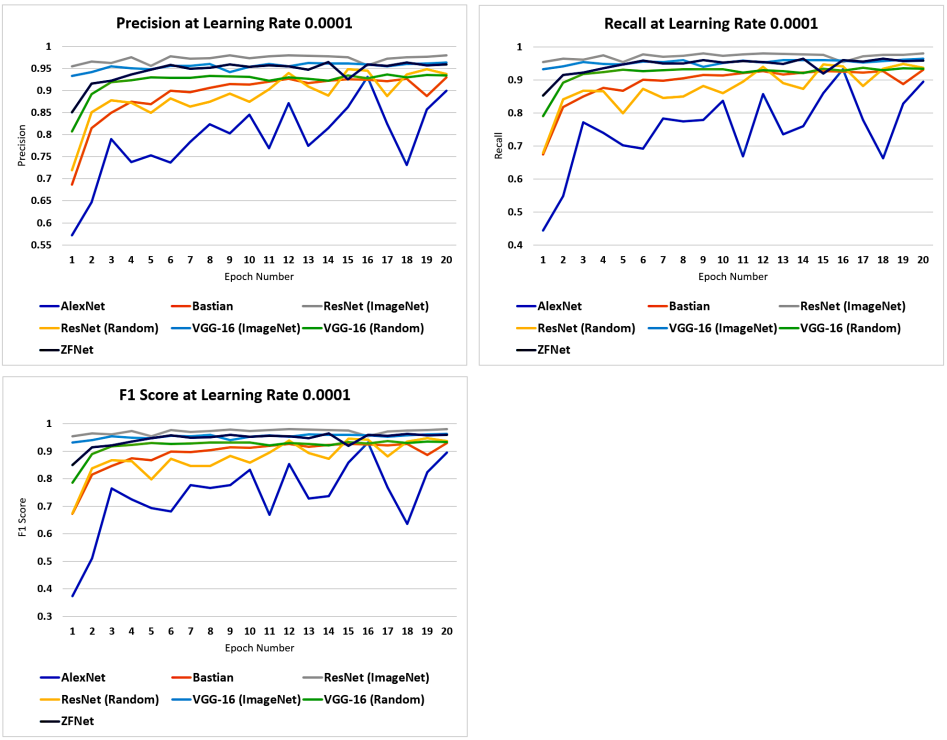

Fig. 1. Precision, recall, and F1 score at learning rate 0.0001 for all models

Careful selection of hyperparameters during training is essential to achieving accuracy levels as high as possible; from the graphs of precision, recall, and F1 score for each epoch depicted in Figure 1, it can be seen that a learning rate of 0.0001 suited ResNet and enabled that model to achieve a very high rate of accuracy. It can also be observed that AlexNet, in particular, suffered from oscillation at that learning rate, causing fluctuations around the local minima whilst attempting to reduce loss. Had the training stopped at 18 epochs instead of 20, AlexNet would have achieved an F1 score of only 0.64 - in fact, training with a learning rate smaller by a factor of 10, AlexNet achieved its best F1 score of 0.92 after the full 20 epochs.

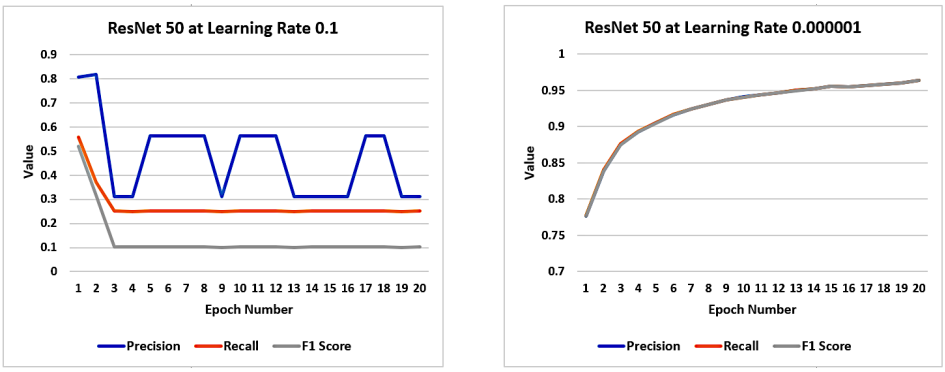

Fig. 2. Precision, recall, and F1 score at learning rate extremes for ResNet 50

To illustrate the effect of learning rate selection on accuracy, results using the extremes of learning rate values in this data can be seen in Figure 2. With the largest learning rate (0.1) the model failed to converge at all; the smallest learning rate of 0.000001 gave much better results and the model converged well. However, the precision, recall, and F1 scores all failed to reach the levels seen with a slightly larger learning rate. Were this model trained for more epochs it is possible that the accuracy may continue to increase.

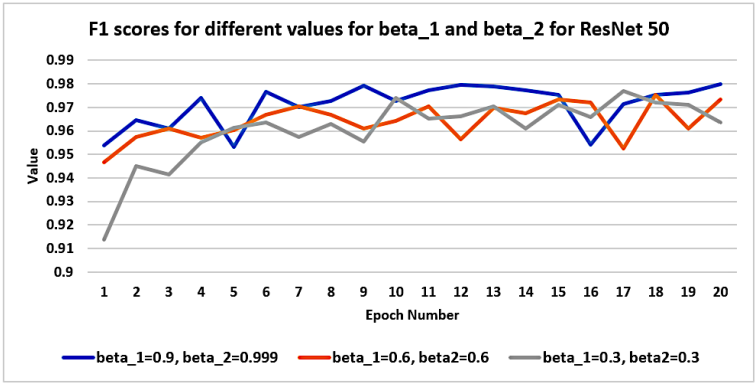

Fig. 3. F1 scores for different beta_1 and beta_2 values for ResNet 50

The authors of the Adam optimiser used in these experiments recommend beta_1 (first moment exponential decay rate) and beta_2 (second moment exponential decay rate) are set to their default values of 0.9 and 0.999 respectively when attempting to solve vision problems [13]. Reducing these values and retraining the most accurate ResNet 50 model caused the precision, recall, and

F1 scores to fall gradually. The results can be seen in Figure 3 and support the authors' recommendation.

5.3 Conclusion

As research into deep learning techniques continues for computer vision-related tasks, the models used become more sophisticated; however, as a side-effect, the level of sophistication in the annotation of the training data required also increases. It is not always possible to obtain data having pixel-level segmentation and, as we have shown, it is possible to achieve very high levels of accuracy using data annotated with simple image-level labels. To answer RQ1, we showed it was possible to train a model - ResNet 50 - to headline precision, recall, and F1 scores of 98%. All of the models in the test performed well, with the lowest accuracy levels achieved by AlexNet still over 90%.

To answer RQ2, we found that the highest accuracy in this study was achieved through the use of transfer learning; by obtaining a model - Resnet-50 or VGG-16 - that has weights pretrained on the ImageNet dataset and freezing the model's convolutional layers, it is possible to train only the classifier with the new corrosion data and achieve higher accuracy than models without the pretraining.

These results provide a baseline for further experimentation with deep learning for corrosion detection, and experimentation with other forms of component degradation such as cracking. The results also show that by using training data annotated at an image level, instead of more sophisticated pixel-level segmentation, it is still possible to achieve a high level of accuracy.

Acknowledgements We would like to thank Dr. Blossom Bastian [4] for providing the dataset used in these experiments. This research is supported by the University of Salford and Add Energy.

References

[1] Geoffrey E. Hinton Alex Krizhevsky Ilya Sutskever. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25* (2012).

[2] Y. Liu B.S. Dhillon. "Human error in maintenance: a review". In: *Journal of Quality in Maintenance Engineering* (2006).

[3] Robert Kelly Barbara Shaw. "What is Corrosion?" In: *The Electrochemical Society Interface* (2006).

[4] Blossom Bastian et al. "Visual inspection and characterization of external corrosion in pipelines using deep neural network". In: *NDT & E International* 107 (2019).

[5] Liang-Chieh Chen et al. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation". In: *ECCV*. 2018.

[6] Ronald Williams David Rumelhart Geoffrey Hinton. "Learning representations by back-propagating errors". In: *Nature* 323 (1986), pp. 533–536.

[7] Kasthurirangan Gopalakrishnan et al. "Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection". In: *Construction and Building Materials* 157 (2017), pp. 322–330.

[8] Yanming Guo et al. “Deep learning for visual understanding: A review”. In: *Neurocomputing* 187 (2015), pp. 27–48.
[9] H. M. Hashemian. “State-of-the-Art Predictive Maintenance Techniques”. In: *IEEE Transactions on Instrumentation and Measurement* 60.1 (2011).
[10] Kaiming He et al. “Mask R-CNN”. In: *Computer Vision and Pattern Recognition* (2017). URL: https://arxiv.org/abs/1703.06870.
[11] Y. Huang et al. “Cost-Effective Vehicle Type Recognition in Surveillance Images With Deep Active Learning and Web Data”. In: *IEEE Transactions on Intelligent Transportation Systems* 21.1 (2020), pp. 79–86.
[12] ImageNet. *ImageNet*. Web Page. 2021.
[13] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. Conference Paper. 2015. URL: https://arxiv.org/abs/1412.6980.
[14] Yiqing Liu and Justin K. W. Yeoh. “Robust pixel-wise concrete crack segmentation and properties retrieval using image patches”. In: *Automation in Construction* 123 (2021), p. 103535.
[15] Mohammad Sadegh Norouzzadeh et al. “A deep active learning system for species identification and counting in camera trap images”. In: *Methods in Ecology and Evolution* 12.1 (2020), pp. 150–161.
[16] Rajiv Tiwari Purushottam Gangsar. “Signal based condition monitoring techniques for fault detection and diagnosis of induction motors: A state-of-the-art review”. In: *Mechanical Systems and Signal Processing* 144 (2020).
[17] Judi E. See. “Visual inspection reliability for precision manufactured parts”. In: *Human Factors* 57.8 (2015).
[18] Verma Ankush Singla, Elisa Bertino, and Dinesh. “Overcoming the Lack of Labeled Data: Training Intrusion Detection Models Using Transfer Learning”. In: *2019 IEEE International Conference on Smart Computing (SMARTCOMP)* (2019).
[19] Warren McCulloch Walter Pitts. “How we know universals the perception of auditory and visual forms”. In: *The bulletin of mathematical biophysics volume* 9 (1947), pp. 127–147.
[20] Karl Weiss, Taghi Khoshgoftaar, and DingDing Wang. “A survey of transfer learning”. In: *Journal of Big Data* 3 (2016).
[21] Xing Wu et al. “COVID-AL: The diagnosis of COVID-19 with deep active learning”. In: *Medical Image Analysis* 68 (2021), p. 101913.
[22] Leijian Yu et al. “AMCD: an accurate deep learning-based metallic corrosion detector for MAV-based real-time visual inspection”. In: *Journal of Ambient Intelligence and Humanized Computing* (2021).