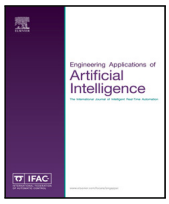
Engineering Applications of Artificial Intelligence 156 (2025) 111097

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Research paper

A novel triplet loss architecture with visual explanation for detecting the unwanted rotation of bolts in safety-critical environments

Tom Bolton [a,*], Julian Bass [a], Tarek Gaber [a], Taha Mansouri [a], Peter Adam [b], Hossein Ghavimi [b]

[a] University of Salford, Department of Science, Engineering, and the Environment, The Crescent, Salford, M5 4WT, United Kingdom
[b] ABL Group Limited, 1 Albyn Place, Aberdeen, AB10 1BR, United Kingdom

ARTICLE INFO

Keywords:
Deep learning
Triplet loss
Convolutional neural networks
Explainability
Continuous maintenance
Bolts and bolting

ABSTRACT

In the commonly used method of bolting to secure parts of equipment and structure, the bolts must be tightened to an adequate preload force. Failure to do so could affect the integrity of the structure, as well as the efficient running of the site and, crucially, employees' safety. In this project, we consider the use of artificial intelligence (AI) techniques to analyse maintenance videos and identify the unwanted loosening of bolts over time in order that they might be used as additional tools in a continuous maintenance plan.

We found that accuracy levels of up to 97% could be achieved in identifying bolt rotation with our proposed machine learning-based triplet loss architecture. The use of gradient-weighted class activation mapping (Grad-CAM) visualisations to identify areas of the image where change had occurred enabled us to test how robust our model was to noise in the data. This explanation may assist users in safety-critical environments guiding them to the problem, and helping mitigate the black-box nature of machine learning algorithms.

Whilst the accuracy of the models varies depending on the rotational angle of the bolt, we clearly demonstrate that triplet loss is a good basis for performing change detection in industrial settings. Furthermore, Grad-CAM has shown to be a useful technique to help a user understand the decisions made by the network and allow them to see where unwanted rotation has occurred.

1. Introduction

In the commonly used method of bolting to secure parts of equipment and structure, the bolts must be tightened to an adequate preload force sufficient for a rated level of mechanical security. A mechanical torque wrench (Wang et al., 2013) is one tool that can be used for this task. The correct preload force is crucial: the Piper Alpha gas platform explosion in 1988, which took 167 lives, was found to be the fault of an incorrectly installed blind flange, secured with bolts (Drysdale and Sylvester-Evans, 1998).

Driven largely by machine learning, artificial intelligence (AI) is increasingly used to accompany existing predictive maintenance techniques such as visual inspection and signal processing analysis (Li et al., 2024). Convolutional neural networks such as Faster R-CNN (Ren et al., 2017) and You Only Look Once (YOLO) (Redmon et al., 2016) have made great advances in static object detection within images and video. For analysis of maintenance issues such as corrosion (Yu et al., 2021; Bastian et al., 2019; Nash et al., 2020) and cracking (Liu and Yeoh, 2021; Gopalakrishnan et al., 2017; Ji and Xiaodong, 2021), these technologies are proving their worth. However, for a condition such as bolt rotation, we need to move towards a method of artificial intelligence analysis that considers the apparatus or component as it changes over time as a snapshot will tell us comparatively little.

Machine learning algorithms undergo training in order to learn a loss function and adjust internal weights such that decisions are made by the model with good accuracy in a particular problem space. This can present an issue in that the weights themselves are not human interpretable and, therefore, the decisions made by the model are unaccountable (Mansouri and Vadera, 2022; Retzlaff et al., 2024). In a safety critical environment, it is crucial that these decisions are reliable and accurate; techniques such as gradient-weighted class activation mapping (Grad-CAM) can help by offering visualisations denoting critical regions in the input images (Selvaraju et al., 2017).

With the wider aim of developing a system that might be able to detect bolt rotation in non-aligned images, and explain its decisions such that a user can see the regions of the image in which the change occurred, we ask the following research questions (RQs):

* Corresponding author.
E-mail address: t.j.e.bolton@edu.salford.ac.uk (T. Bolton).

https://doi.org/10.1016/j.engappai.2025.111097
Received 21 November 2024; Received in revised form 31 March 2025; Accepted 5 May 2025
Available online 3 June 2025

- RQ1: using a dataset depicting bolts in various degrees of rotation to train machine learning models, what level of accuracy is achievable in determining change in the rotational angle of bolts?
- RQ2: is it possible to use a visualisation technique to explain the decisions the trained network is making?
- RQ3: is any achievable visualisation sufficient to enable a user to see where change in the image has occurred?

Current studies on the use of machine learning to detect bolt rotation use a variety of different techniques that range in complexity. Some are based around signal analysis (Wang and Song, 2020; Yuan et al., 2022; An and Sohn, 2012; Zhang et al., 2019b) which is not always transferable to a new bolted connection. Techniques that use visible threads to determine bolt looseness either have a fairly wide margin of minimum 4 mm visible thread (Gong et al., 2022; Ramana et al., 2019; Cha et al., 2016; Yuan et al., 2021; Zhang et al., 2020) or reduce that margin through the use of specialist 3D cameras (Pan et al., 2023; Pan and Yang, 2023).

A system that needs only input from a 2D camera would require no 3D imaging or electronic sensors. It should be transferable to new work sites without having to replace bolts or manually mark existing bolts (Zhao et al., 2019; Yang et al., 2022). It should have accuracy comparable with the state of the art in other methods that use only 2D image data (Luo et al., 2024, 2023; Huynh et al., 2019).

To detect change in a bolt's position over time we look to comparative methods. Metric loss architectures, in which two or more parallel convolutional neural networks are used as backbones to extract features that are then used to learn a shared loss function, have been used to good effect in satellite imagery comparisons (Chen et al., 2022; Zhang et al., 2022), face detection (Schroff et al., 2015a), and video-based object tracking (He et al., 2018).

Bolted connections can be made with a variety of fasteners. Whilst a bolt generally comprises a long, thin cylindrical shaft that is threaded, the head of the bolt may take many forms such as hex ("Allen") key, slotted or crosshead ("Phillips") screw, and others besides (Mushtaq et al., 2023). For simplification in these experiments, we consider bolts that have a hexagonal head designed to be turned using a spanner or socket.

For detecting bolt rotation, we need to capture a time series of images depicting the connection that we wish to monitor. This data might come from a body camera worn by a maintenance engineer performing regular inspections. Use of unmanned aerial vehicles (drones) and robots to capture image data for use in industrial maintenance is increasingly commonplace (Hu and Assaad, 2023). However, capturing the data in this way means that it could present several challenges when using machine learning for analysis:

- *Temporal image alignment:* it cannot be guaranteed that each of the images in a time series captured by the camera will show the bolted connection from the same angle or height resulting in unpredictable geometric noise (Bianchi et al., 2023). Likewise, we cannot rely on a known distance between the camera and the subject in any given image.
- *Gaussian noise:* a time series will inevitably feature images and video that is captured at different times of day and in different lighting conditions, causing variance in shadow, reflections, and contrast which may affect the accuracy of the models used (Rodríguez-Rodríguez et al., 2024).
- *Spurious noise:* it is possible that the surrounds of the bolted connection might change over time. Equipment left nearby during repair work, for example, might be captured in one of the time series of images.

To address these challenges, we propose a metric loss architecture based on the triplet loss function (Schroff et al., 2015a). This method negates the need for precise alignment of images captured over time, and can be trained to be invariant to both geometric and radiometric noise. In addition, triplet loss can be trained with data having relatively weak annotations meaning that bounding box or pixel-level annotations are not necessary, saving a great deal of time and human effort.

To accompany traditional maintenance techniques, we propose a metric loss architecture to aid in the detection of unwanted bolt loosening. We offer a visual explanation of our architectures using Grad-CAM. In summary, our contributions are as follows:

- Through experimentation with selected triplet loss networks having two different feature extraction backbones, we make a scientific comparison of the accuracy achievable in detecting bolt rotation and find that levels of more than 90% are achievable without specialist camera or sensor equipment, or reliance on a particular bolt type or marking
- We further employ a feature extraction backbone fine-tuned specifically on bolts to further compare accuracy levels
- We present a visual explanation of the model's feature extraction backbones, enabling us to understand the regions of the image that led to the model's decision — something that is essential in a safety-critical environment

To the best of our knowledge, this is the first triplet loss network to both detect the rotation of bolts, and offer a visual explanation of its decisions. Our findings show that high levels of accuracy (greater than 90%) are achievable with our proposed model when detecting bolt rotation of ten degrees or more. We find that fine-tuning can increase accuracy levels at greater degrees of bolt rotation with an associated tradeoff in accuracy levels with bolt rotation angles of ten degrees or less.

Using Grad-CAM functionality to visualise the model offers a human-interpretable output enabling us to see what is driving the networks' decisions and enables the user to see exactly whereabouts in the image change has occurred. Whilst the interpretation of this output is qualitative, it can be seen from the results that the bolts that have moved are clearly and unequivocally highlighted.

With good accuracy and a usable explanation, the proposed model has potential applications beyond bolt rotation. Problem areas such as cracking and corrosion that have previously employed AI as a snapshot analysis are domains in which the detection of slowly-worsening degradation could be very useful.

The rest of this paper is organised as follows: Section 2 reviews existing research related to this article; Section 3 discusses the methodology used in the experiments; Section 4 notes the results of the experiments; Section 5 is a general discussion of these results; Section 6 concludes the paper and makes recommendations for future work. A CReDIT author statement, acknowledgements, and a glossary of abbreviations can be found at the end of the document.

2. Related work

Neural networks, as a branch of artificial intelligence, can trace their roots back to the 1940s, and – in particular – a paper by Walter Pitts and Warren McCulloch in which two neural mechanisms were described that '…exhibit recognition of forms' (Pitts and McCulloch, 1947); the authors set out to use a simulation of the human nervous system to solve learning problems mathematically.

The current popularity of deep learning used in our experiments can be ascribed to several factors (Wu et al., 2019):

- The availability of large-scale annotated datasets such as ImageNet (Deng et al., 2009)
- The availability of performant, affordable parallel computing in the form of graphical processing units (GPUs)
- Advances in network design and training strategies

In the years since LeCun et al. proposed LeNet, one of the first modern convolutional neural networks (Lecun et al., 1998), research into vision-based feature extraction has been prolific. Newer object detection algorithms such as ResNet (He et al., 2016) and MobileNet and its derivatives (Sinha and El-Sharkawy, 2019) perform more accurately on fewer data.

The training data required for models to learn niche problems such as bolt rotation is scarce. To help make the most use of few data, transfer learning can be employed ahead of the training process. Transfer learning is a technique by which learning in one domain is improved by transferring information from another domain Weiss et al. (2016). The 14 million-sample ImageNet is one of several publicly available datasets that can be used as a basis for transfer learning, containing hundreds of thousands of real-life images for each node of a set hierarchy of classifications (Deng et al., 2009).

2.1. Bolts and bolt rotation

Research into detecting bolt loosening varies in the overarching methodology employed — this can be loosely categorised into several distinct groups.

Studies have used sensor-based methods in which signals are applied to a bolted connection and read using transducers or recording equipment, the resulting signals often analysed using a form of machine learning (Wang and Song, 2020; Yuan et al., 2022; An and Sohn, 2012; Zhang et al., 2019b). These studies vary in the complexity of apparatus required, from "*...an IIG measurement hardware system developed by integrating an arbitrary waveform generator, a digitizer, two high-speed multiplexers and a self-sensing circuit*" (An and Sohn, 2012), to a simple audio recorder (a smartphone) and a hammer (Yuan et al., 2022; Zhang et al., 2019b).

Of the studies that employ sophisticated electronic equipment, Wang et al. demonstrate very high accuracy of over 98%; additionally, the authors' setup was able to detect a loose bolt in a multi-bolt join (Wang and Song, 2020). The two studies making use of percussion and recording are of interest not only because the apparatus required is minimal (a recorder and a hammer) but because this methodology is transferable to different bolted connections (Yuan et al., 2022; Zhang et al., 2019b). Negating the need for a permanent installation of transducers reduces cost and increases flexibility in a continuous maintenance plan. Moreover, a very attractive property of these studies is that they are able to detect different preload forces meaning that bolt loosening could potentially be detected early on.

The next methodology considered in the literature is that which uses a vision object detection model to consider different classes of bolt, namely 'tight' and 'loose'; these studies are predicated on the bolt having differing lengths of visible threads (Gong et al., 2022; Ramana et al., 2019; Cha et al., 2016; Yuan et al., 2021; Zhang et al., 2020). The approaches taken in these studies differ, but one consistent factor remains — the minimum visible thread considered is 4 mm which means the bolt must be fully loose before any change is detected. Raman et al. report very high accuracy (over 90%) in most of their experimental scenarios with the exception of those using the most acute vertical camera angle; obfuscation in the region of interest is an issue for any vision approach to the problem (Ramana et al., 2019).

Research into the use of light detection and ranging (LiDAR) technology is demonstrating that the minimum 4 mm visible thread can be reduced to under a millimetre (Pan and Yang, 2023; Pan et al., 2023). The use of 3D points clouds increases accuracy but requires specialist cameras; ordinary footage captured by existing 2D cameras during a maintenance inspection would not be suitable for this approach.

The use of a known pattern on the head of a bolt is featured in research that first employs a vision model to calculate a bounding box around the head of the bolt and, separately, the pattern; corners of these boxes are used to calculate the relative rotational angle of the bolt head (Zhao et al., 2019). This method was tested with bolt rotation angles as small as 10 degrees with an average error of 4.47%; it does, however, require the use of a particular type of bolt which cannot be guaranteed on an existing work site. A similar study using manually-applied markings to bolt-nut joins uses a YOLO model to analyse images and demonstrates detection of rotation down to two degrees (Yang et al., 2022). Whilst markings would have to be manually applied, this is a less invasive and time-consuming task than replacing every bolt.

Finally, research into the use of a combination of naive techniques – Canny edge detection and Hough line transforms – coupled with machine learning has demonstrated detection of rotational movement in bolts down to ten degrees (Luo et al., 2024; Huynh et al., 2019). Canny edge detection can be sensitive to image lighting and contrast, prompting Luo et al. to propose a grey gradient method to improve Canny edge detection (Luo et al., 2023).

Few of these studies make mention of detecting change over time — many are dedicated to detecting the angle of a bolt as a snapshot. Where the concern lies not in tracking the angle but in determining that a bolt has moved, we look towards the use of change detection machine learning architectures.

2.2. Change detection - Metric loss architectures

The problem of object tracking in a sequence of video frames was studied by He et al. who devised a novel architecture SA-Siam (He et al., 2018). The network uses, as input, a pair of patches cropped from the target frame and the current frame. The network SA-Siam comprises an appearance branch, a clone of another work (SiamFC) which is, itself, a metric loss architecture for object tracking. SA-Siam's semantic branch is a convolutional neural network trained as an object detection model. The dataset used for training is large: the ILSVRC-2015 dataset comprises around 1.3 million frames with 2 million tracked objects. The authors claim results that outperform all other real-time trackers (at the time of writing).

A change detection model using a U-Net backbone, developed to address traditional remote sensing algorithms' poor performance on complex change tasks, was proposed by Chen et al. (2022). Siamese_AUNet uses a feature attention module for spatial and channel attention on the deep feature layer, with atrous spatial pyramid pooling, enabling the capture of contextual information at multiple scales. The authors demonstrated F1 scores of between 86% and 93%, all of which were higher than the four other networks with which the authors' architecture was compared. These results are, however, predicated on access to readily-prepared datasets having accurate segmentation masks.

FaceNet is a metric loss architecture using three parallel convolutional feature extraction backbones and a novel triplet loss function (Schroff et al., 2015b). FaceNet uses the feature extraction backbones with a classifier layer that reduces the feature maps to an embedding in Euclidean space. The loss function is learned with triplets of training data, each consisting of an anchor, a similar positive, and a dissimilar negative. The loss function is learned such that the distance between the anchor-positive embeddings is less than that of the anchor-negative embeddings. It is an elegant solution that does away with the bottleneck layer used in previous metric loss architecture designs that relied on the contrastive loss function.

2.3. Explainability

In the paper outlining Meta AI's Segment Anything Model (SAM), the authors consider the fairness of their algorithm's detections (Kirillov et al., 2023). A section of the article explores how SAM fairs in a responsible AI test. The data used in training is analysed according to geographical and income-related distributions. The fairness of the model is also investigated with respect to "*segmenting people across perceived gender presentation, age group, and skin tone*". The results of this latter are enlightening — detection and segmentation underrepresentations are reported for females, older and younger people, and

darker skin tone. The authors "*...acknowledge biases may arise when SAM is used as a component in larger systems*". (Kirillov et al., 2023) SAM does remain, however, without explanation. The considerations of its bias came from observing its decisions — detection and segmentation. The weighting and inner workings of the model that led it to these decisions are not observable.

There are various methods for implementing explainability in a model — each method has its advantages and shortcomings. Retzlaff et al. authored a paper in which they examine various explainable AI (xAI) techniques and present a decision tree for data scientists and developers of machine learning models to guide them towards a suitable choice (Retzlaff et al., 2024). This is a thorough study, one of the main tenets of which is whether a particular xAI technique is post- or ante-hoc. This is an important distinction: an ante-hoc model is intrinsically explainable via interpretability methods built in to the architecture or during training. A post-hoc model, conversely, can be explained after the event by examining the model when a decision has been made.

Mansouri et al. make a similar distinction in a study that proposes a novel explainable layer that the authors call Gumbel-Sigmoid eXplanator (GSX) (Mansouri and Vadera, 2022). GSX is an ante-hoc explanation method that ascribes importance to features as the model is trained. GSX hinges around an instance-wise feature selection layer that is differentiable and trainable. This layer outputs the selected features for each instance along with an explanation of which were important in the decision. Regularisation limits the number of features that are considered. This has great potential for use in vision problems where the selected features – areas of an image – could present a useful interpretation of the model's workings.

The use of post-hoc xAI for change detection has been explored by Zhang et al. (2022). The authors firstly establish a baseline for change detection using bi-temporal image pairs or triplets, and an associated loss function — contrastive loss for pairs, and triplet loss for groups of three. In doing so, the authors move away from other forms of change detection that use classifier-based methods and towards metric learning. By calculating losses for changed and unchanged regions, we see how the authors have devised an architecture that not only produces similarity scores, but outputs region maps delineating where change has occurred.

Class Activation Mapping (CAM) describes a procedure for generating activation maps using global average pooling, thereby offering a visual indication of the discriminative regions used by a convolutional neural network (CNN) to identify that class (Zhou et al., 2016). Gradient-weighted Class Activation Mapping (Grad-CAM) improves upon the CAM work by using the gradients of a target concept (a class) flowing into the final convolutional layer of a CNN to produce a coarse localisation map (Selvaraju et al., 2017). Grad-CAM, unlike its predecessor, can be used with a wide variety of CNN model families.

Further work explores the use of Grad CAM in a study that offers visual explanations of an embedding network (Chen et al., 2020). The authors used only data with segmentation and bounding box annotations. Livieris et al. likewise explore the use of Grad CAM in a contrastive loss network using data annotated at only image level (Livieris et al., 2023).

2.4. Research gap

From the literature it can be seen that there are gaps in current research in the problem domain of bolted connections.

The first is the application of a machine learning-based approach that considers the problem over time; several studies show us how we might make a snapshot appraisal of the position of a bolt, either rotational or from the perspective of visible threads (Zhang et al., 2019a; Huynh et al., 2020; Wang et al., 2018). There are none that appear to consider how the rotational angle might change over time, with later images analysed and compared with earlier images. It is change over time that is absolutely necessary for us to be able to identify a bolt that is loosening.

In addition, the use of explainable AI has yet to be considered in this context. Outputs from current metric loss networks are showing a good level of accuracy in satellite imagery and other domains (He et al., 2018; Chen et al., 2022; Schroff et al., 2015b), but the decisions that the models are making have not been taken into account. In a safety-critical environment, it is essential that the algorithms in use are able to demonstrate their accuracy.

3. Methodology

We constructed a triplet loss-based machine learning architecture to compare and analyse temporal triplets of bolt images to detect differences in bolt rotation angle. Using a novel training dataset to establish a baseline accuracy, the architecture can show a visual explanation of activations that contribute to its decision.

The use of a metric loss architecture is proposed to counter the problem of deviation in the camera angle. It cannot be guaranteed that the bolted connection will be captured from the same angle every time, requiring a model that is invariant to this geometric noise. However, we accept that a model, no matter how accurate, cannot predict deviations in a bolt angle that are not obvious to the human eye.

To establish any correlation in accuracy and deviation in bolt rotation angle, we devised a series of experiments using a known novel dataset.

3.1. Dataset

For the purposes of these experiments, we required images showing a bolted apparatus with the bolts rotating incrementally and by known amounts. As there was no such data publicly available, the authors compiled a dataset containing 1112 images in laboratory conditions. The images depict a purpose-built apparatus having five bolts, three of which were rotated by various degrees simulating change. The degree of bolt rotation, camera angles, and focal length were carefully measured and recorded and used as annotations for each sample.

The dataset was constructed with the aim of establishing a baseline from which further work could be carried out. The samples introduce a variety of geometric noise in known quantities to simulate the various angles and focal lengths from which real-world images may be obtained.

During the research for this study, we were able to use a quadruped robot – Boston Dynamics' "Spot" – to obtain images of the test apparatus that was used in the image dataset. Spot was programmed to simulate a walkaround as might take place in a maintenance inspection, during which time the robot paused to capture images of the bolted apparatus using an ordinary webcam with a vertical resolution of 1080p. Using the Python library OpenCV to extract discrete frames from the captured video reveals some of the issues that must be considered when using footage captured in an industrial setting.

Fig. 1 clearly shows the bolted apparatus; the image is of little use as it is significantly distorted with motion blur.

Fig. 2 shows that clear images can be captured from the video with no sharpening or other post-processing; this image was present in the video footage only a few frames after that in Fig. 1. Geometric variation is evident in the camera angle — the apparatus is being viewed from above.

Fig. 3 shows that geometric deviation – height and horizontal angle – as well as increased focal length can be expected in the images that are captured. This image, like that in Fig. 2, is clearly focussed. To establish a baseline, our dataset considers this geometric noise in its different classes.

Training a triplet loss architecture requires data arranged in temporal triplets (Schroff et al., 2015a). These consist of a ground truth image depicting the object of interest in a known state (the 'anchor'), an image

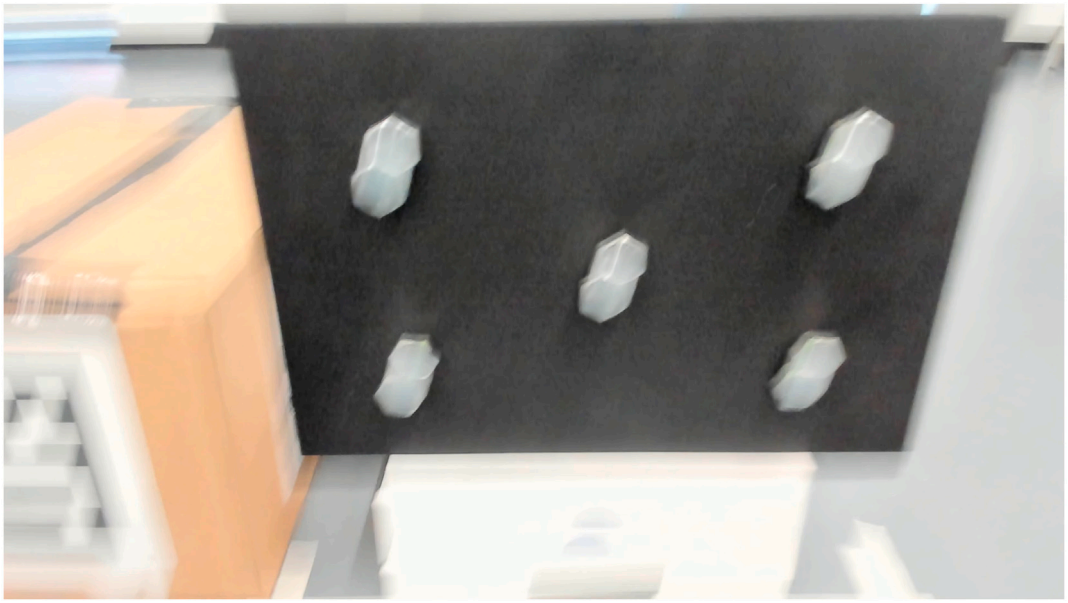


Fig. 1. A blurred image of the bolted apparatus captured by Spot the robot dog.

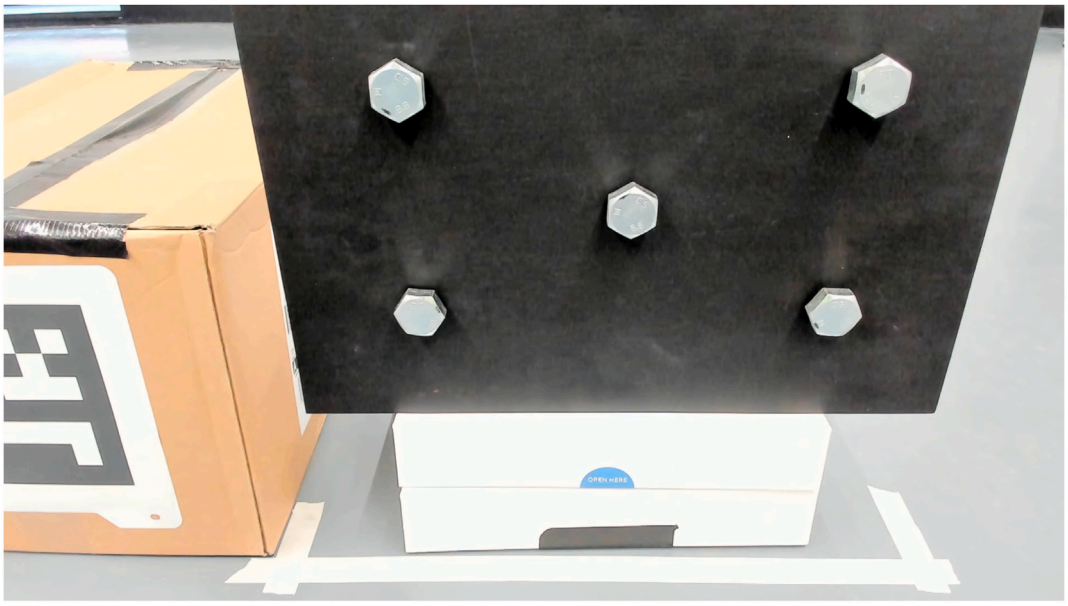


Fig. 2. A clear image of the bolted apparatus captured by Spot the robot dog.

depicting the same unchanged object with some variation in geometry (the 'positive'), and an image depicting the object in a changed state (the 'negative').

Fig. 4 shows a sample from the dataset depicting the bolted apparatus with deviations in both camera and bolt rotation angle. In this image, the horizontal camera angle has moved by 40 degrees from perpendicular, the camera height is 1590 mm from floor level, and the lower three bolts have been rotated counter-clockwise from the ground truth position.

With these images, it was possible to construct many triplets having an anchor and positive image with the same bolt angle, and a negative in which the lower three bolts are at a different angle of rotation.

Fig. 5 shows an example triplet constructed from the bolt rotation dataset showing an anchor, positive, and negative:

- *Anchor* - *Fig. 5(a):* bolts rotated 10 degrees, horizontal camera angle deviation of 25 degrees, focal length of 50 mm
- *Positive* - *Fig. 5(b):* bolts rotated 10 degrees, horizontal camera angle deviation of 5 degrees, focal length of 35 mm
- *Negative* - *Fig. 5(c):* bolts rotated 40 degrees, horizontal camera angle deviation of 35 degrees, focal length of 80 mm

Using triplets compiled in this way, we can vary the rotational angle of the bolts, and introduce other geometric variations such as camera angle and focal length. The goal is that the model will learn that the change in the bolts' rotation is the region of interest and become invariant to other noise.

3.2. Change detection

Metric loss networks, sometimes referred to as 'Siamese', are used in a variety of similarity comparisons such as satellite imagery analysis (Chen et al., 2022) and face detection (Schroff et al., 2015a). Using

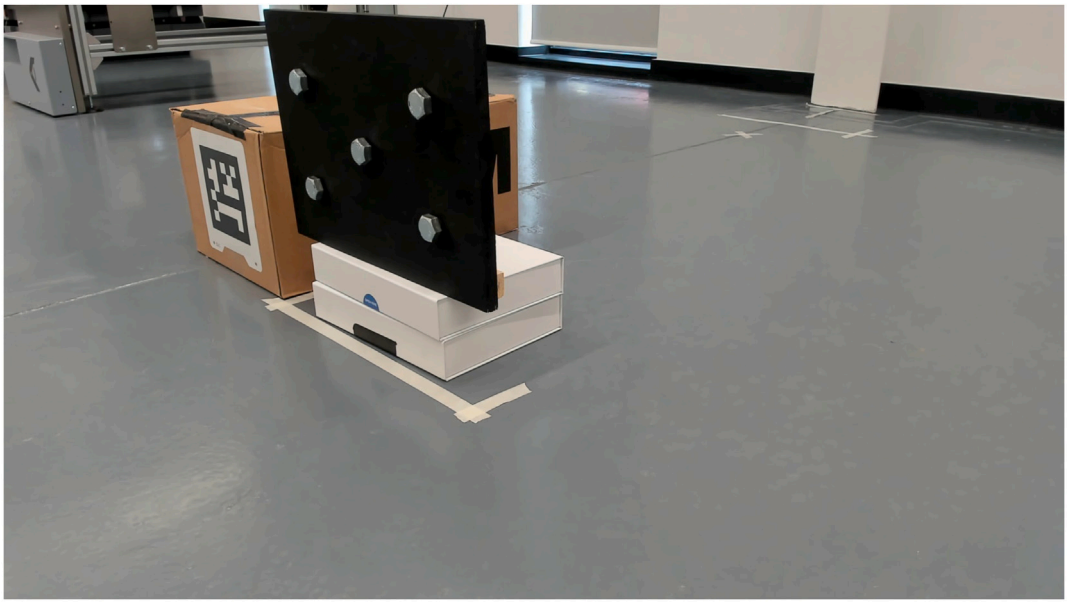


Fig. 3. A different clear image of the bolted apparatus captured by Spot the robot dog.

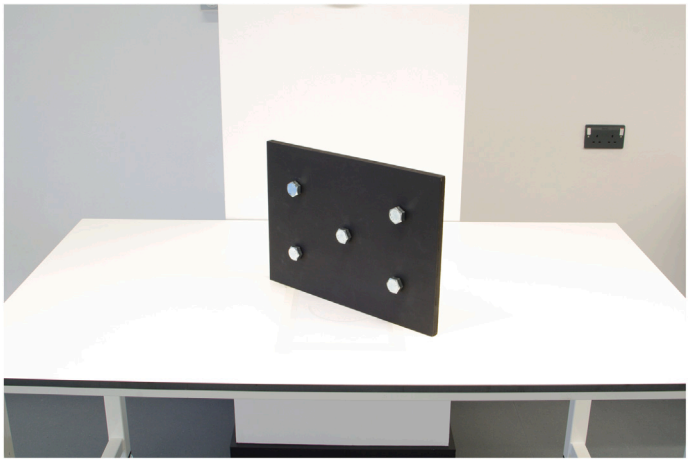
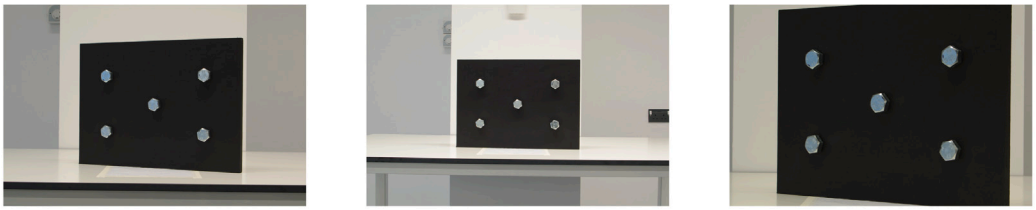


Fig. 4. A sample image from the bolt rotation dataset.



(a) Anchor (b) Positive (c) Negative

Fig. 5. An example training triplet.

two or three parallel feature extraction backbones with shared weights, embeddings are derived from input pairs or triplets of images. These fixed-sized encodings can then be used to compute similarity scores. These networks are robust; we are using weakly-labelled training data having a deliberately-introduced level of geometric noise to assess the models' suitability in a real-world application where we might not have data that is perfectly aligned.

By using an image of the bolted apparatus in a known state, with the bolts sufficiently tensioned, we establish an anchor. The positive image also depicts the apparatus in a known good state but from a different camera angle, deliberately introducing geometric noise. Finally, the negative image shows the apparatus having three of its five bolts deliberately loosened by a known amount. The images are labelled according to the deviation in the bolts' rotational angle — the camera angle is introduced at random.

Fig. 6 depicts the proposed architecture used in these experiments. Three convolutional neural network backbones are used in parallel to extract features from training data which is fed to the network in

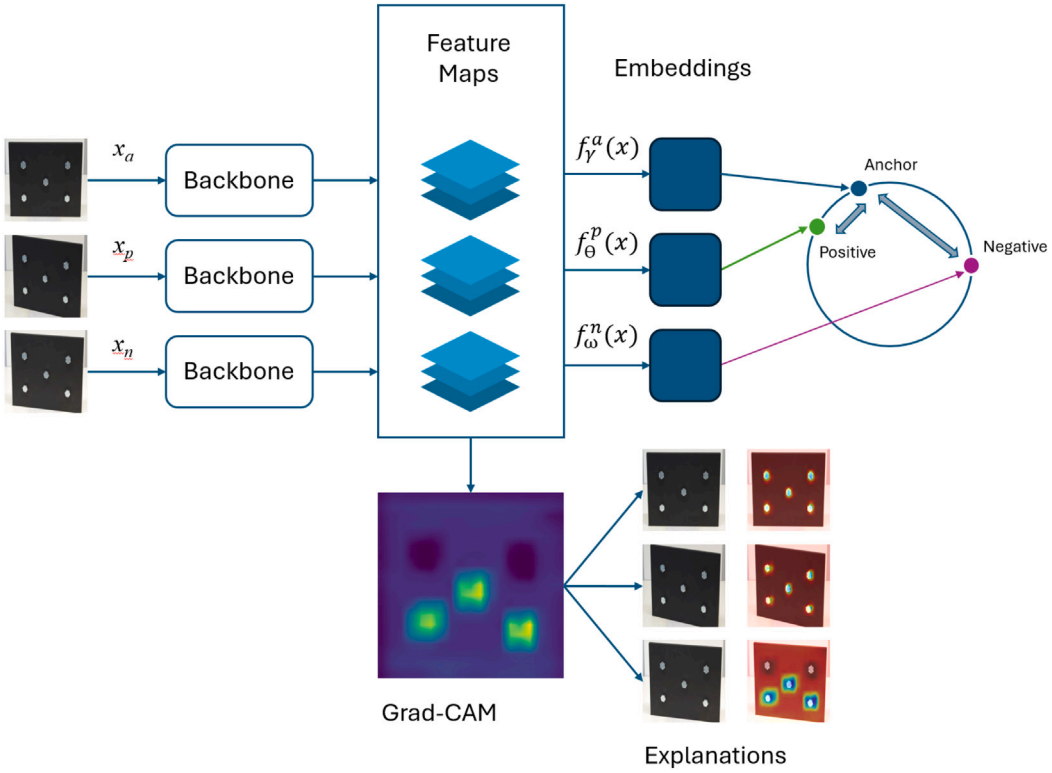


Fig. 6. Our proposed triplet loss architecture.

triplets — anchor, positive, and negative. The backbones' weights are shared; in practice, there is only one backbone into which the triplet images are fed using separate input layers. These features are then fed to a fully-connected classifier network, consisting of one or more layers. Finally, the resultant embeddings are passed to the triplet loss function.

Scroff et al. introduced the idea of a metric loss architecture based on their novel triplet loss function that they applied the idea to the problem of face recognition (Schroff et al., 2015a). The goal is to learn the loss function such that the Euclidean distance between the anchor embedding and the positive embedding is smaller than the distance between the anchor and negative embedding.

$$f_\gamma^a(x) \in \mathbb{R}^d \tag{1}$$

$$f_\theta^p(x) \in \mathbb{R}^d \tag{2}$$

$$f_\omega^n(x) \in \mathbb{R}^d \tag{3}$$

Per Fig. 6, Eqs. (1), (2), and (3) embed an image x into a d-dimensional Euclidean space resulting in output embeddings $f_\gamma^a(x)$, $f_\theta^p(x)$, and $f_\omega^n(x)$.

$$\left\| f_\gamma^a(x) \right\|_2 = 1 \tag{4}$$

$$\left\| f_\theta^p(x) \right\|_2 = 1 \tag{5}$$

$$\left\| f_\omega^n(x) \right\|_2 = 1 \tag{6}$$

In Eqs. (4), (5), and (6), we constrain these embeddings to exist on the d-dimensional hypersphere.

$$\left\| f_\gamma^a(x_i) - f_\theta^p(x_i) \right\|_2^2 + \alpha < \left\| f_\gamma^a(x_i) - f_\omega^n(x_i) \right\|_2^2, \tag{7}$$

$$\forall (f_\gamma^a(x_i), f_\theta^p(x_i), f_\omega^n(x_i)) \in T \tag{8}$$

As visualised in Fig. 6, we want to ensure that an image x_i (the *anchor*) of a bolt having a specific rotational angle is closer to all other images x_i of the bolt having that angle, and always further from any image x_i of a bolt with a different angle of rotation (Eq. (7)). The margin α is enforced between positive and negative pairs.

T is the set of all possible triplets (anchor, positive, negative) in the training dataset, that has cardinality N (Eq. (8)).

$$L = \sum_i^N [\left\| f_\gamma^a(x_i) - f_\theta^p(x_i) \right\|_2^2 - \left\| f_\gamma^a(x_i) - f_\omega^n(x_i) \right\|_2^2 + \alpha]_+ \cdot \tag{9}$$

The loss L that we wish to minimise is derived in Eq. (9) (Schroff et al., 2015a).

3.3. Explainability

Based on work by Livieris et al. (2023), our proposed architecture includes hooks for deriving visualisations of the gradients of a target class as sent through the last convolutional layer. This technique is called Grad-CAM and provides us with a class-discriminative localisation map (viewed as a heatmap) of activations in that layer which can then be enlarged and overlaid to fit the original image (Selvaraju et al., 2017).

$$\alpha_k = \frac{1}{z} \sum_i \sum_j \frac{\partial y}{\partial A_{ij}^2} \tag{10}$$

The localisation map $L_{Grad-CAM}$ is obtained by firstly calculating the neuron importance weights α_k using the gradient of the model's output y with respect to the kth map activations A^k of a specific convolutional layer — in our case, the final layer (Eq. (10)). These are propagated back and global average pooled over the image's width (i) and height (j) dimensions. Here, Z is the total number of spatial locations in the heatmap.

$$L_{Grad-CAM} = ReLU\left(\sum_k a_k A^k \right) \tag{11}$$

We can then derive a weighted combination of forward activation maps followed by ReLU activation to calculate $L_{Grad-CAM}$ (Eq. (11)).

The model's decision is visually indicated by using these gradients with respect to the model's internal feature maps to generate a heatmap.

By overlaying the resultant heatmap on the image, it is possible to see which regions of the anchor, positive, or negative caused activations and therefore which areas of the image are most contributing to reducing the anchor-positive distance, and which are enlarging the anchor-negative distance. It is these features that are learning the triplet loss function and therefore influencing the trained model's decisions; by ensuring that, for example, the bolts that have moved are contributing to the greater anchor-negative distance and not other background noise, we can be more certain of the model's usefulness.

Allowing stakeholders this transparency builds greater trust when the models are used as part of a continuous maintenance program.

3.4. Experiments

In order to answer the research questions posed in Section 1 - Introduction, we devised a series of experimental scenarios.

3.4.1. Comparing architecture backbones

To answer RQ1 ("Using a dataset depicting bolts in various degrees of rotation to train machine learning models, what level of accuracy is achievable in determining change in the rotational angle of bolts?") we devised a series of comparisons using a triplet loss architecture comprising a feature extraction backbone with three input layers, a fully-connected classifier, and a triplet loss function as described in Section 3.2 - Change Detection:

- *Feature extraction backbones:* ResNet-50, MobileNet - pretrained on ImageNet
- *Classifier:* fully-connected, two layer
- *Hyperparameters and optimiser:* batch size - 4, learning rate - 0.0001, epochs - 30, optimiser - Adam

Datasets of triplets were compiled using randomisation. The random selection was specifically used to make triplets that depict bolted connections with varying deviations in the bolt rotation angle; the randomisation also introduced geometric noise by varying the camera angle and height, as well as the focal length. By constructing the triplets in this manner, the common factor in the anchor and positive images was the rotational angle of the bolt; the differing factor in the anchor-negative pair was also the bolts' rotational angle. Geometric noise – deviation in the camera angle, and variation in the focal length – was introduced at random.

Triplets were constructed with deviations in bolt angle rotation limited to 5, 10, 15, 20, 25, and 30 degrees. The triplets' anchor and positive images had no rotation — all bolts were at the starting position. The negative image had the lower three bolts rotated by the given amount. In both positive and negative images, deviations in camera angle and focal length were introduced at random.

The following pseudocode gives a high-level overview of the initialisation and training/validation loops:

```
START:
Load dataset triplet file pointers from text file;
Load images into dataset;
Split into train/validation/test;

Load model architecture;

FOR number_of_epochs:
  Initialise loop and metric counters;

  # training
  FOR number_of_batches:
   Obtain embeddings of anchor, positive, negative;
   Calculate euclidean distance anchor-positive and anchor-negative;

  IF (anchor-negative - anchor-negative) > margin:
   Correct predictions ++;
   Total predictions ++;
  ELSE
   Total predictions ++;

  Backpropagate error;
  Update parameters;
 NEXT batch

 # validation
 FOR number_of_batches:
  Obtain embeddings of anchor, positive, negative;
  Calculate euclidean distance anchor-positive and anchor-negative;

  IF (anchor-negative - anchor-negative) > margin:
   Correct predictions ++;
   Total predictions ++;
  ELSE
   Total predictions ++;
 NEXT batch

 IF validation accuracy improved:
  Update saved model

NEXT epoch
END
```

3.4.2. Fine-tuning the feature extraction backbone

We used a fine-tuned ResNet-50 backbone, trained on ImageNet and then fine-tuned on the NPU-Bolt dataset (Zhao et al., 2022). This is a dataset of 337 images of mechanical fixings, in four classes. We used a ResNet-50 backbone that had shown the greatest accuracy in bolt detection in a comparison using a Faster R-CNN object detection model (Bolton et al., 2023).

- *Feature extraction backbone:* ResNet-50, pretrained on ImageNet and fine-tuned on NPU-Bolt
- *Classifier:* fully-connected, two layer
- *Hyperparameters and optimiser:* batch size - 4, learning rate - 0.0001, epochs - 30, optimiser - Adam

3.4.3. Exploring visualisation for explainability

To answer RQ2 ("is it possible to use a visualisation technique to explain the decisions the trained network is making?") and RQ3 ("is any achievable visualisation sufficient to enable a user to see where change in the image has occurred?"), we used Gradient-weighted Class Activation Maps (Grad-CAM) to obtain visual explanations of the activations within the last convolutional layer of the feature extraction backbones. The goal was to verify that the model was indeed learning the correct features; as we are using data that has variation in bolt rotation, as well as camera angle and focal length, we used Grad-CAM to identify coarse localisation maps showing the areas of the image that had led to the model's prediction.

A new model architecture was written, adapted by work from Livieris et al. whose research used Grad CAM within a contrastive loss network (Livieris et al., 2023). We added hooks to the triplet loss model enabling us to extract the information from all three embedding paths within the ResNet during backpropagation, allowing us to see activations within the anchor, the positive, and the negative image.

Table 1
Training results - triplet loss with ResNet-50 feature extractor pretrained on ImageNet.

	Rotation angle deviation	Training accuracy	Validation accuracy
ResNet-50	5	0.5919	0.2976
	10	0.7554	0.4286
	15	0.973	0.9286
	20	0.9662	0.9762
	25	0.9905	0.9405
	30	0.9554	0.9405

4. Results

Each of the models was implemented in code, using Python along with a variety of machine learning libraries. Discrete Python environments and their installed packages were managed with the Anaconda distribution of Python.

The following computer was used for training: a laptop equipped with an 11th generation Intel Core i7 CPU, 32 GB of RAM, and an RTX3080 GPU with 16 GB of VRAM.

4.1. Detecting bolt rotation with triplet loss

In this scenario, different triplet loss architectures with ResNet-50 and MobileNet feature extractors and two-layer classifiers were trained with different datasets having deviations in the rotation angle of three of five bolts. Both ResNet-50 and MobileNet feature extractors were pretrained on the ImageNet dataset (Deng et al., 2009).

The angle deviation increased by five degrees from the initial starting position. For each round of training, the dataset was shuffled before being split into training, validation, and test sets.

4.1.1. ResNet-50 - Pretrained on ImageNet

Six experiments were run on the training data using a triplet loss architecture with a ResNet-50 feature extraction backbone. Table 1 shows the training, validation, and test accuracy which are a function of the correct predictions as a proportion of the total number of predictions. These results consider training accuracy on samples from which the model learns, and validation accuracy on a test set which has no bearing on the model's weights. Validation accuracy increases broadly in line with greater deviation in bolt rotation.

The highest accuracy achieved was with a 20 degree deviation in the bolts' rotation angle, the lowest was with a five degree deviation. 30 epochs of training and validation were completed for each round, and took between 11 and 12 h per round.

Fig. 7 shows the test accuracy of the ResNet-50 triplet loss architecture on the test datasets across the range of bolt angle deviations. The graph depicts validation accuracy (y axis) for each epoch number (x axis). Each set of points represents validation accuracy for a given bolt rotation deviation. The headline accuracy given in Table 1 is the highest for all epochs at each bolt angle. It can be seen that, in each round of training, the architecture started reaching convergence at between five and 10 epochs, after which smaller gains were made.

4.1.2. MobileNet - Pretrained on ImageNet

Six further experiments were run on the training data using a triplet loss architecture with a MobileNet feature extraction backbone. Table 2 shows the training, validation, and test accuracy which are a function of the correct predictions as a proportion of the total number of predictions. Results consider training accuracy on samples from which the model learns, and validation accuracy on a test set which has no bearing on the model's weights. Validation accuracy increases broadly in line with greater deviation in bolt rotation.

The highest accuracy achieved was with a 30 degree deviation in the bolts' rotation angle, the lowest was with a five degree deviation. 30 epochs of training and validation were completed for each round, and took between 11 and 12 h per round.

Table 2
Training results - triplet loss with MobileNet feature extractor pretrained on ImageNet.

Bolt angle deviation	Training accuracy	Validation accuracy
5	0.6503	0.3949
10	0.8142	0.6087
15	0.8265	0.6775
20	0.8128	0.7065
25	0.8333	0.6902
30	0.8661	0.8098

Table 3
Training results - triplet loss with ResNet-50 feature extractor pretrained on ImageNet and then fine-tuned with NPU-Bolt (Zhao et al., 2022).

Rotation angle deviation	Training accuracy	Validation accuracy
5	0.6093	0.2681
10	0.7254	0.4384
15	0.9781	0.8261
20	0.9918	0.9620
25	0.9536	0.8043
30	0.9918	0.9293

Fig. 8 shows the test accuracy of the ResNet-50 triplet loss architecture on the test datasets across the range of bolt angle deviations. The graph depicts validation accuracy (y axis) for each epoch number (x axis). Each set of points represents validation accuracy for a given bolt rotation deviation. The headline accuracy given in Table 2 is the highest for all epochs at each bolt angle. It can be seen that, in each round of training, the architecture started reaching convergence after eight epochs, after which smaller gains were made.

4.2. Fine-tuning the feature extraction backbone

In this scenario, a triplet loss architecture with a ResNet-50 feature extractor and two-layer classifiers was trained with different datasets having deviations in the rotation angle of three of five bolts. The ResNet-50 feature extraction backbone was pretrained on the ImageNet dataset (Deng et al., 2009). The ResNet was then fine-tuned on the NPU-Bolt dataset, a collection of images of bolts and other fixings (Zhao et al., 2022). The feature extractor was trained within a Faster R-CNN model as part of a series of experiments to determine optimal hyperparameter settings for bolt detection (Bolton et al., 2023).

The angle deviation increased by five degrees from the initial starting position. For each round of training, the dataset was shuffled before being split into training, validation, and test sets.

Six further experiments were run on the training data using a triplet loss architecture with a fine-tuned ResNet feature extraction backbone. Table 3 shows the training, validation, and test accuracy which are a function of the correct predictions as a proportion of the total number of predictions. Above 10 degrees, the validation accuracy shows no discernible pattern with respect to deviation in bolt rotation.

The highest accuracy achieved was with a 20 degree deviation in the bolts' rotation angle, the lowest was with a five degree deviation. 30 epochs of training and validation were completed for each round, and took between 11 and 12 h per round.

Fig. 9 shows the test accuracy of the ResNet-50 triplet loss architecture on the test datasets across the range of bolt angle deviations. The graph depicts validation accuracy (y axis) for each epoch number (x axis). Each set of points represents validation accuracy for a given bolt rotation deviation. The headline accuracy given in Table 3 is the highest for all epochs at each bolt angle. It can be seen that, in each round of training, the architecture started reaching convergence after between seven and 10 epochs, after which smaller gains were made.

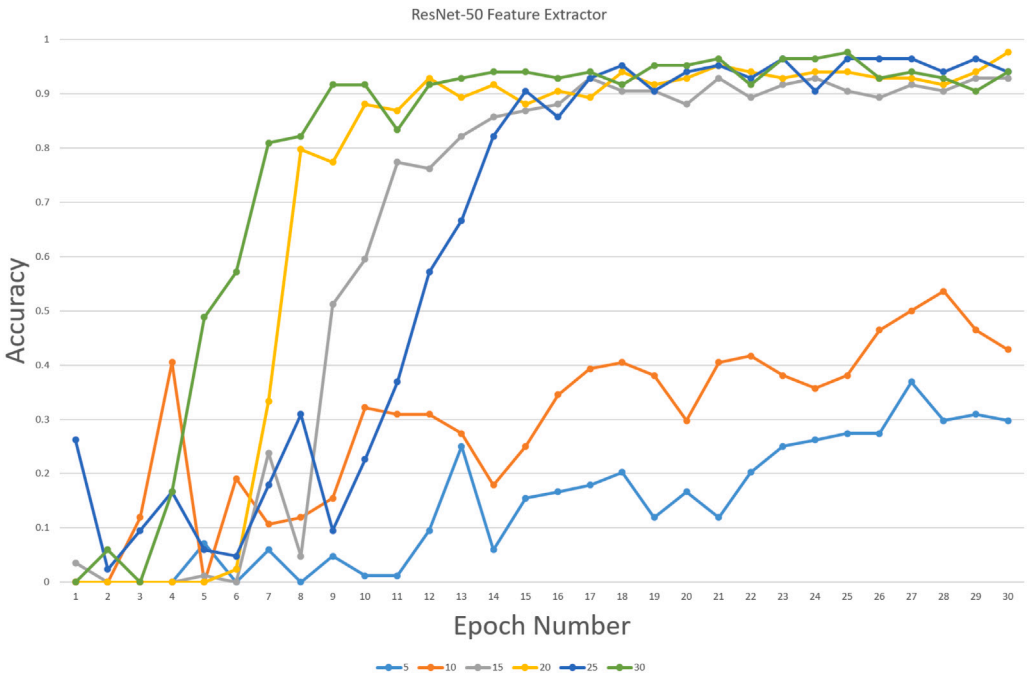


Fig. 7. Results - triplet loss with ResNet-50 feature extractor.

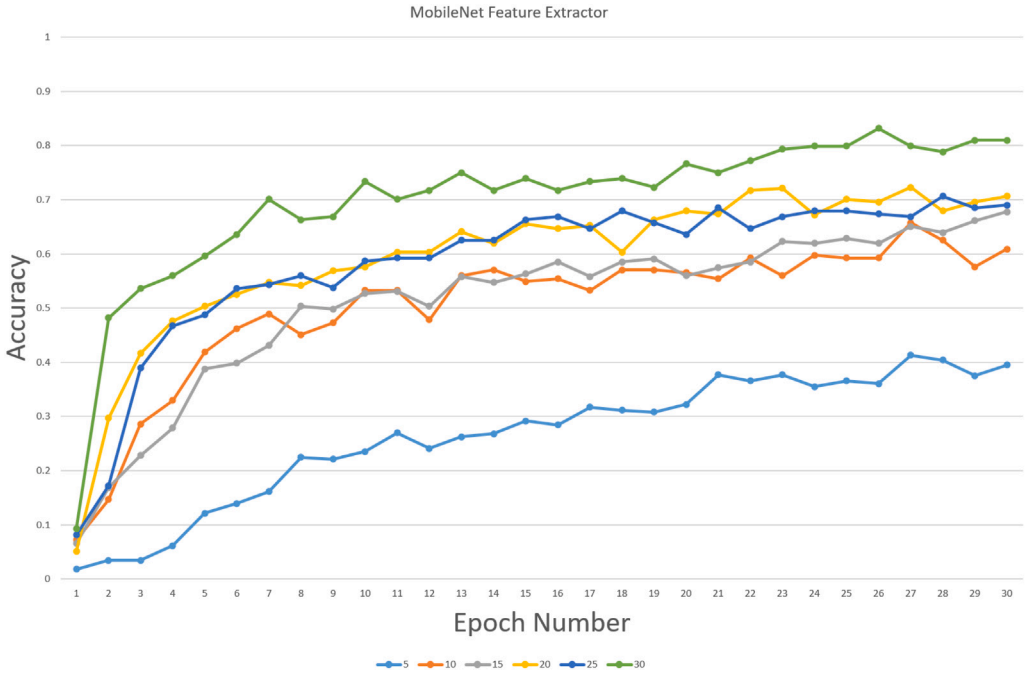


Fig. 8. Results - MobileNet.

4.3. Analysing triplet loss with Grad-CAM

Using our proposed framework, we can examine visualisations generated by the trained models.

4.3.1. ResNet and MobileNet

For each of these architectures, we present the Grad-CAM visualisations for the most and least accurate models.

In six experiments using a ResNet-50 feature extractor pretrained on ImageNet, the model reached its highest accuracy level on bolts with 20 degrees deviation.

The sample images on which the model made predictions are overlaid with a heatmap. The heatmap is a coarse localisation depicting areas of importance in the final convolutional layer of the feature extraction backbone, and tells us the features that the model has learned to prioritise. It is generated by the gradients of our desired target concept flowing into that convolutional layer. It can be seen from Fig. 10a (anchor) and (c) (negative) that the model has clearly delineated the regions of interest in the anchor and, in the negative, the three bolts that have moved.

This model was least accurate with bolts at five degrees deviation.

In Fig. 11, it can be seen that there are no useful areas delineated by the heatmaps overlaying any of the three images. Again, the heatmap

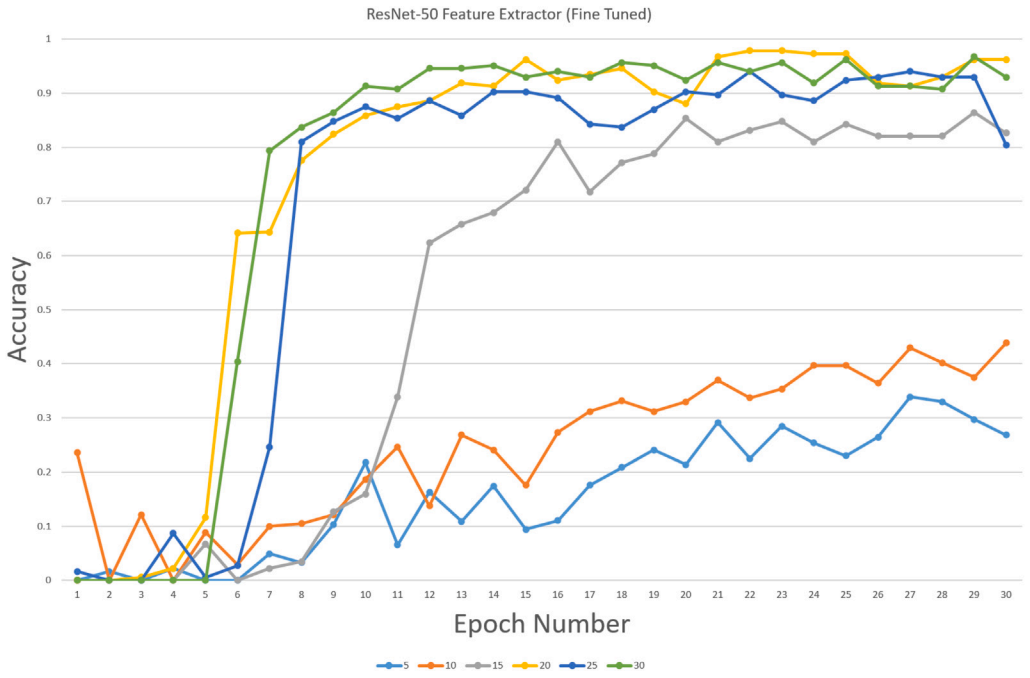


Fig. 9. Results - ResNet-50 (fine-tuned).

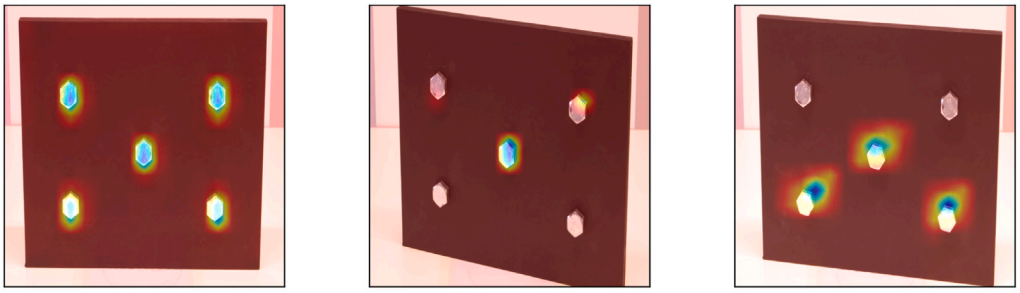


Fig. 10. ImageNet Grad-CAM output for 20 degrees bolt angle deviation.

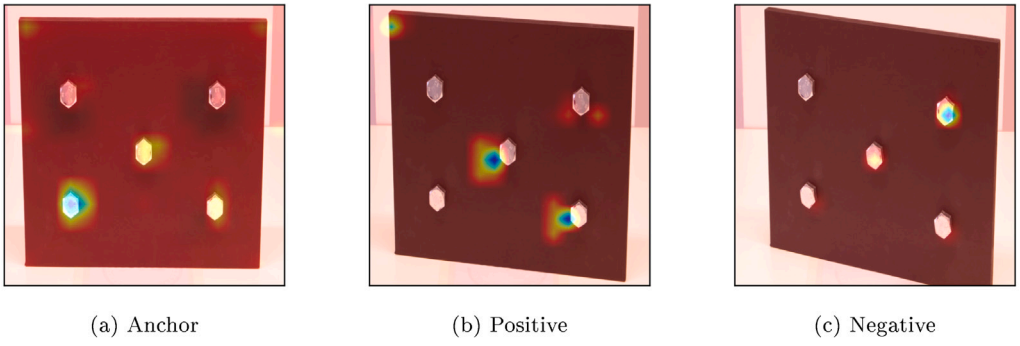
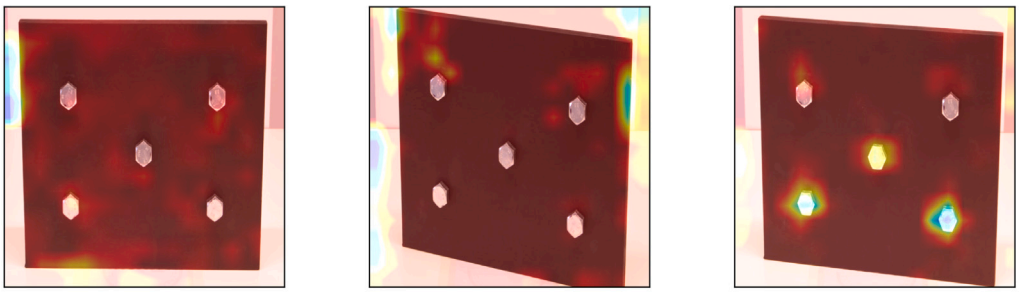


Fig. 11. ImageNet Grad-CAM output for 5 degrees bolt angle deviation.

is a coarse localisation depicting areas of importance in the final convolutional layer of the feature extraction backbone.

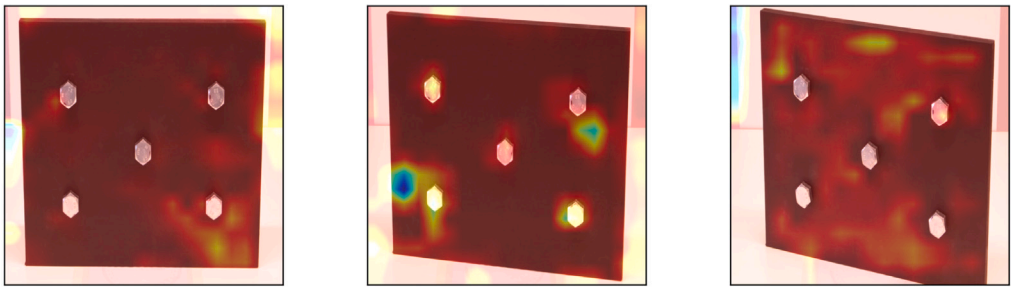
In our six experiments with a MobileNet feature extractor, we found that the triplet loss architecture was most accurate when trained with bolts having a 30 degree deviation in rotation angle; it was – as hypothesised – least accurate when trained with data showing bolts with the smallest 5 degree deviation in rotation angle.

The MobileNet network trained on 30 degree rotation deviations, whose accuracy was 81%, is shown in Fig. 9; the Grad-CAM visualisations from the last convolutional layer of the extraction backbone for each of the anchor, positive, and negative inputs are overlaid. We can interpret areas of the heatmap tending towards blue in colour as representing stronger activations in the convolutional layer.



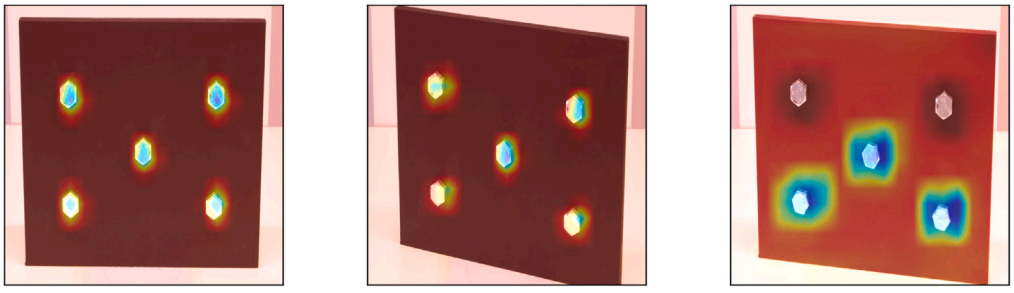
(a) Anchor (b) Positive (c) Negative

Fig. 12. MobileNet Grad-CAM output for 30 degrees bolt angle deviation.



(a) Anchor (b) Positive (c) Negative

Fig. 13. MobileNet Grad-CAM output for 5 degrees bolt angle deviation.



(a) Anchor (b) Positive (c) Negative

Fig. 14. Fine-tuned ResNet-50 Grad-CAM output for 20 degrees bolt angle deviation.

Fig. 12(a) and (b) depict the anchor and positive inputs; the activations are focused around the horizontal edges of the apparatus. Fig. 12(c) depicts the negative, the image in which we are keen to see if the model has learned the change in the bolts' rotational angle; there are three strong areas of activation around the lower three bolts suggesting that this is the case.

The MobileNet-based architecture was least accurate with data depicting a five degree deviation, reaching 39%.

Fig. 13 depicts the visualisations for this network. There is little definition in the heatmaps for any of the images; crucially, Fig. 13(c) – the negative in which we would expect to see activations centred on the lower three bolts – shows little, if anything that could be construed as such.

4.3.2. Fine-tuned ResNet

Of the six experiments that used a ResNet-50 feature extraction backbone, the model trained with data showing 20 degrees deviation was most accurate, and the model trained on five degrees deviation least accurate.

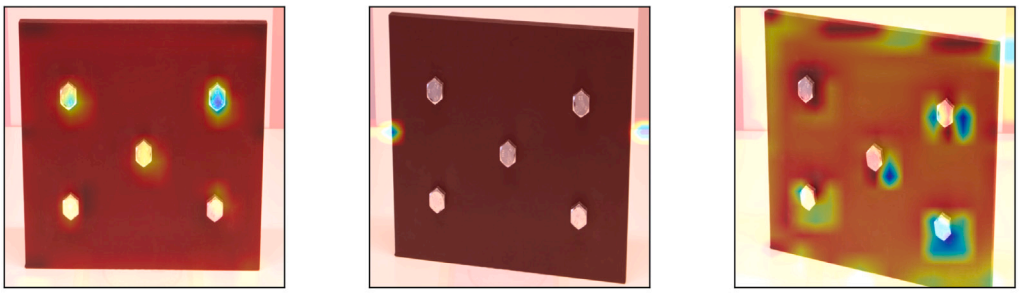
Firstly, the model trained on 20 degrees data that reached 96% accuracy:

Fig. 14 depicts the Grad-CAM activations for this model. It can be seen that Fig. 14(a) and (b) – the anchor and positive – are showing very clearly areas of activation in the images that were learned as a result of the triplet loss function penalising the network if the anchor-positive distance was large. In both of these images, all five bolts are clearly showing a concentration of activations.

Conversely, we can see in Fig. 14(c) - the negative - a very clear concentration around the lower three bolts showing fairly conclusively the regions of the image where bolt rotation has occurred.

In contrast to the high accuracy of the 20 degree model, the five degree model reached a poor accuracy level of 27%.

Fig. 15 shows the three outputs from this five degree model. The results are less conclusive; the anchor (Fig. 15(a)) and positive (Fig. 15(b)) show activations suggesting the learned similarities were not necessarily the bolts. The negative (Fig. 15(C)) does show some activations around the lower three bolts, however these are not concentrated solely on the lower three bolts as we would like. Rather, the top-right bolt has caused some activations which are, for this purpose, incorrect.



(a) Anchor (b) Positive (c) Negative

Fig. 15. Fine-tuned ResNet-50 Grad-CAM output for 5 degrees bolt angle deviation.

Table 4
A comparison of our proposed network and other published work

Studies	Key ideas	Minimum rotation	Extra equipment	Manual correction
Wang and Song (2020), Yuan et al. (2022), An and Sohn (2012) and Zhang et al. (2019b)	Signal applied to bolted connection and result analysed	Can detect change in preload force with no visible rotation	Yes - waveform generators, digitizers, hammer	No
Gong et al. (2022), Ramana et al. (2019), Cha et al. (2016), Yuan et al. (2021) and Zhang et al. (2020)	A model is trained to recognise two classes — loose bolt and tight bolt. The difference lies in the amount of visible thread.	4mm visible thread (630 degrees)	None	No
Pan and Yang (2023) and Pan et al. (2023)	LiDAR cameras are used to analyse bolted connections and detect differences in the Z-plane	Less than 1 mm visible thread (225 degrees)	Yes - specialist LiDAR cameras	No
Zhao et al. (2019)	A model is trained to recognise patterns on a bolt head relative to the bolt head itself. The relationship is analysed to derive rotational angle	10 degrees	Yes - a specific type of bolt with a known pattern	No
Luo et al. (2024), Huynh et al. (2019) and Luo et al. (2023)	Canny edge detection and hough line transforms are used to derive angles of a bolt head relative to the horizontal	10 degrees	In one study, a square gasket for each bolt	Yes - bolts must be pictured from directly in front, and any deviation must be corrected
Ours	A triplet loss architecture is trained to recognise bolt rotation with respect to a known ground truth	10 degrees	None	No

5. Discussion

Having trained a series of models on a dataset that showed progressive amounts of bolt rotation, from five degrees to 30 degrees, we find that our initial hypotheses have, to a certain extent, been validated:

- Excessive deviation in the camera angle makes the bolts' rotational angle harder to estimate
- When the camera angle's deviation reaches 90 degrees, the apparatus is side-on to the camera and it is very difficult to identify any deviation in the bolts' rotational angle
- The most obvious deviation in the bolts' rotational angle is 30 degrees; the least obvious is 5 degrees

Training models using data having variations in camera angle, from straight-on to the apparatus to a side view, we find that those models showing greater degrees of accuracy are invariant to this noise. Likewise, the deviation in the bolts' rotational angle would seem to be not only more obvious to a human the greater it becomes, but also to a neural network.

In line with one of our study's aims – that the method should involve no specialist apparatus, and be useable on any site – our proposed architecture demonstrates advantages over the state of the art. Table 4 details the studies and overarching methodologies that were grouped in Section 2 - Related Work. We can compare these methods by the minimum bolt rotation in degrees they can detect, whether they require any specialist equipment besides a standard video camera, and whether any human intervention is required to manipulate the images.

Our method requires no apparatus other than a camera to capture images; sensor-based methods need often complicated equipment to transmit and capture signals for analysis, something that is not necessarily transferable to new worksites (Wang and Song, 2020; Yuan et al., 2022; An and Sohn, 2012; Zhang et al., 2019b). Our method requires no particular type of bolt or any markings to be made upon existing bolts (Yang et al., 2022; Zhao et al., 2019). Finally, we require no specialist cameras for capturing 3D points clouds (Pan et al., 2023; Pan and Yang, 2023). Of the studies that use image data, we have shown results comparable to those that can detect bolt rotations as low as ten degrees with good accuracy (Luo et al., 2024; Huynh et al., 2019).

Safety-critical industrial environments are, by their nature, dangerous; access to such sites is necessarily limited. By using a robot – Spot the quadruped dog – we have gone some way towards simulating a maintenance inspection. In doing so, we found that whilst clear, non-blurry frames were obtainable from the video, geometric noise in the form of variations in camera angle and focal length were always present. Our dataset focussed on these geometric aberrations.

One aspect that should be discussed is the amount of time required to train these models; between 11 and 12 h were required to complete the 30 epochs used in our experiments. It is likely that a dedicated server-based GPU would be more powerful and free of any thermal

throttling, thereby considerably reduce training times. Even so, this does indicate that the computational load required indicates that this task is not suitable for edge computing on – for example – low-powered, low-cost Arm devices such as Android tablets.

5.1. RQ1: using a dataset depicting bolts in various degrees of rotation to train machine learning models, what level of accuracy is achievable in determining change in the rotational angle of bolts?

We found that good levels of accuracy were achievable — headline figures of more than 98% do not, however, tell the full story. The accuracy was dependent on the rotational angle of the bolts. Perhaps not surprisingly, all of the architectures tested did struggle with data showing little deviation.

Achievable accuracy rose approximately with increasing bolt angle deviation, although there was no strict correlation with this hypothesis throughout the three model architectures. Models using an ImageNet-pretrained extraction backbone, both ResNet-50 and MobileNet, struggled with data showing only five degrees of deviation. Whilst showing higher accuracy at greater rotation degrees, the ImageNet-trained ResNet-50 backbone also struggled to delineate bolt rotation at 10 degrees.

The architecture with a ResNet-50 backbone that we fine-tuned using a static bolt detection dataset did not result in greater accuracy levels. In fact, accuracy levels for both the five degree data and the 10 degree data performed poorly (27% and 44% respectively). Moving to the 15 degree data, there was a large jump to 83% and, after that, no discernible pattern as the bolt angle widened.

Taken at face value, these results are promising — the instinct in a production scenario would be to choose the models showing the greatest accuracy. However, we have used feature extraction backbones that have been pretrained on a dataset containing millions of images covering 1000 classes. Furthermore, we introduced known deviations in camera angle and focal length to simulate the sort of geometric noise that might be found in an industrial setting.

In short, we do not know for certain that the models have learned the features of the data we would like, whether they are invariant to the features we are not interested in, and how much has been skewed by the pretraining on many classes and shapes of object. This highlights the need for a degree of explainability — safety in maintenance is crucial and AI making black-box decisions, however high the reported accuracy, cannot be trusted until it has been at least visualised if not explained.

5.2. RQ2: is it possible to use a visualisation technique to explain the decisions the trained network is making?

We introduced Grad-CAM to a triplet loss architecture in order to show the activations relating to the images that have been learned to force the anchor-positive distance smaller than the anchor-negative. We could see that, in the models displaying good test accuracy, the negative visualisations were indeed clustered on the bolts that had moved, and not on the geometric noise. At this point it appears that the reported accuracy of the models in scenario one – where we were essentially 'driving blind' – can be visually tested by the Grad CAM interpretation.

There is an inherent degree of interpretation in assessing a heatmap; qualitatively assessing whether or not change has occurred – the bolt has rotated – becomes harder where the model's accuracy was low and the visual output is less clear. This is, perhaps, where the extra fine-tuning of the ResNet backbone has helped; even with very poor accuracy, the five degree fine-tuned model showed more promise in the visualisation. Three of the rotated bolts showed activations, with an added false positive in a fourth bolt that had not moved.

In comparison, as the training accuracy of the MobileNet-based architecture fell, the visual output became of little use despite showing 39% accuracy compared with the fine-tuned ResNet's 27% at five degrees. We find that this proves the requirement for explanation of machine learning — without this insight, instinct would be to select the more accurate model.

5.3. RQ3: is any achievable visualisation sufficient to enable a user to see where change in the image has occurred?

The visualisations that we were able to extract from our triplet loss network showed very clearly where change in the image has occurred. This is, again, contingent on the accuracy of the model in the first instance. This is not necessarily a shortcoming with the Grad-CAM technique itself as any explanation technique would simply report the inherent uncertainty in a model whose training accuracy was poor.

6. Conclusion and future work

We have demonstrated that good levels of accuracy are achievable using our proposed triplet loss architecture to analyse image data and detect bolt rotation. The accuracy of the models was, perhaps unsurprisingly, poorer when asked to learn data showing only minimal differences. We further demonstrated that it is possible to use Grad-CAM to visualise a triplet loss network, and see straight away where similarities and differences in the temporal triplets have been learned by the feature extraction backbone.

Whilst accuracy levels were good for detecting bolt deviation of more than 10 degrees, it would benefit the work to develop a more fine-tunable apparatus to determine the point below which the proposed model was no longer able to accurately detect rotation. The five degree increments used in these experiments reflect the fact that the apparatus was constructed from wood, and accuracy when setting the rotational angle of the bolts could not be guaranteed to a single degree. An apparatus machined from metal, perhaps with CNC, would enable us to take these measurements.

The amount of data is, as discussed, problematic. There are ways in which the data could be augmented, and more random triplets generated. Use of a generative adversarial network (GAN) is tempting, but the images generated could not be guaranteed to represent their class label. Simply put, more data is needed. In constructing our dataset, we used laboratory conditions to establish baseline data with known geometric noise. Further work is required to simulate further real-world conditions and should introduce gaussian noise and lighting and contrast variations.

A potential source of training data is through the use of newer vision foundation models such as Segment Anything Model (SAM) (Kirillov et al., 2023). These models are capable of multi-modal prompting – text, image – and are trained on enormous amounts of data. The output of such a model could generate annotation boxes or masks for data scraped from the internet which could, in turn, be used to train the feature extractors within our triplet loss architecture. This would likely increase the overall computational load requirement which would have to be taken into consideration within a production environment.

Finally, there is potential for more informative explainability. Grad-CAM has worked with some degree of success here, but offers only a post-hoc explanation of an already-trained model. A means of explaining not only these decisions, but the features that were extracted, could be of use. Mansouri et al. propose Gumbel-Sigmoid eXplanator (GSX) which demonstrates a quantitative evaluation of features that contribute to a model's decision during training (Mansouri and Vadera, 2022). This could be adapted for use in a vision model, bringing an ante-hoc explainability layer which would strengthen our xAI metrics.

7. Abbreviations

- **AI** - Artificial Intelligence
- **CNC** - Computer Numerically Controlled
- **CNN** - Convolutional Neural Network
- **CPU** - Central Processing Unit
- **GAN** - Generative Adversarial Network
- **GPU** - Graphical Processing Unit
- **Grad-CAM** - Gradient-weighted Class Activation Mapping

- **GSX** - Gumbel-Sigmoid eXplanator
- **LiDAR** - Light Detection And Ranging
- **RAM** - Random Access Memory
- **R-CNN** - Region-based Convolutional Neural Network
- **ResNet** - Residual Neural Network
- **SAM** - Segment Anything Model
- **SA-Siam** - Semantic and Appearance Features in Siamese network
- **VRAM** - Video-reserved Random Access Memory
- **xAI** - eXplainable Artificial Intelligence
- **YOLO** - You Only Look Once

CRediT authorship contribution statement

Tom Bolton: Writing – original draft, Visualization, Validation, Resources, Methodology, Data curation, Conceptualization. **Julian Bass:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization. **Tarek Gaber:** Writing – review & editing, Supervision, Project administration, Conceptualization. **Taha Mansouri:** Writing – review & editing, Project administration, Conceptualization. **Peter Adam:** Resources, Project administration, Funding acquisition, Conceptualization. **Hossein Ghavimi:** Resources, Project administration, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Tom Bolton reports financial support was provided by ABL Group Limited. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by ABL Group Limited, Aberdeen, United Kingdom; and The University of Salford, Salford, United Kingdom.

Data availability

Data will be made available on request.

References

An, Y.-K., Sohn, H., 2012. Integrated impedance and guided wave based damage detection. Mech. Syst. Signal Process. 28, 50–62. http://dx.doi.org/10.1016/j.ymssp.2011.11.016, URL https://www.sciencedirect.com/science/article/pii/S0888327011005085 Interdisciplinary and Integration Aspects in Structural Health Monitoring.

Bastian, B.T., N, J., Ranjith, S.K., Jijia, C., 2019. Visual inspection and characterization of external corrosion in pipelines using deep neural network. NDT E Int. 107.

Bianchi, E.L., Sakib, N., Woolsey, C., Hebdon, M., 2023. Bridge inspection component registration for damage evolution. Struct. Heal. Monit. 22 (1), 472–495. http://dx.doi.org/10.1177/14759217221083647.

Bolton, T., Bass, J., Gaber, T., Mansouri, T., 2023. Comparing object recognition models and studying hyperparameter selection for the detection of bolts. In: 28th International Conference on Natural Language Processing and Information Systems. Springer-Verlag, Berlin, Heidelberg, pp. 186–200. http://dx.doi.org/10.1007/978-3-031-35320-8_13.

Cha, Y.-J., You, K., Choi, W., 2016. Vision-based detection of loosened bolts using the hough transform and support vector machines. Autom. Constr. 71, 181–188. http://dx.doi.org/10.1016/j.autcon.2016.06.008, URL https://www.sciencedirect.com/science/article/pii/S0926580516301297.

Chen, L., Chen, J., Hajimirsadeghi, H., Mori, G., 2020. Adapting grad-CAM for embedding networks. In: 2020 IEEE Winter Conference on Applications of Computer Vision. WACV, IEEE Computer Society, Los Alamitos, CA, USA, pp. 2783–2792. http://dx.doi.org/10.1109/WACV45572.2020.9093461.

Chen, T., Lu, Z., Yang, Y., Zhang, Y., Du, B., Plaza, A., 2022. A siamese network based U-net for change detection in high resolution remote sensing images. IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens. 15, 2357–2369. http://dx.doi.org/10.1109/JSTARS.2022.3157648.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255. http://dx.doi.org/10.1109/CVPR.2009.5206848.

Drysdale, D.D., Sylvester-Evans, R., 1998. The explosion and fire on the piper alpha platform, 6 july 1988. a case study. Philos. Trans.: Math. Phys. Eng. Sci. 356 (1748), 2929–2951, URL http://www.jstor.org/stable/55055.

Gong, H., Deng, X., Liu, J., Huang, J., 2022. Quantitative loosening detection of threaded fasteners using vision-based deep learning and geometric imaging theory. Autom. Constr. 133, 104009. http://dx.doi.org/10.1016/j.autcon.2021.104009, URL https://www.sciencedirect.com/science/article/pii/S092658052100460X.

Gopalakrishnan, K., Khaitan, S., Choudhary, A., Agrawala, A., 2017. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. Constr. Build. Mater. 157, 322–330.

He, A., Luo, C., Tian, X., Zeng, W., 2018. A twofold siamese network for real-time object tracking. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4834–4843. http://dx.doi.org/10.1109/CVPR.2018.00508.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 770–778. http://dx.doi.org/10.1109/CVPR.2016.90.

Hu, X., Assaad, R.H., 2023. The use of unmanned ground vehicles (mobile robots) and unmanned aerial vehicles (drones) in the civil infrastructure asset management sector: Applications, robotic platforms, sensors, and algorithms. Expert Syst. Appl. 232, 120897. http://dx.doi.org/10.1016/j.eswa.2023.120897.

Huynh, T.-C., Park, J.-H., Jung, H.-J., Kim, J.-T., 2019. Quasi-autonomous bolt-loosening detection method using vision-based deep learning and image processing. Autom. Constr. 105, 102844. http://dx.doi.org/10.1016/j.autcon.2019.102844, URL https://www.sciencedirect.com/science/article/pii/S092658051930250X.

Huynh, H.C.P., Ta, Q.-B., Kim, J.-T., Ho, D.-D., Tran, X.-L., Thanh-Canh, 2020. Bolt-loosening monitoring framework using an image-based deep learning and graphical model. Sensors URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7349298/.

Ji, Q.Y., Xiaodong, 2021. Automatic pixel-level crack detection for civil infrastructure using unet++ and deep transfer learning. IEEE Sensors 21 (17), 19165–19175.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.-Y., Dollár, P., Girshick, R., 2023. Segment anything. In: 2023 IEEE/CVF International Conference on Computer Vision. ICCV, pp. 3992–4003. http://dx.doi.org/10.1109/ICCV51070.2023.00371.

Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324. http://dx.doi.org/10.1109/5.726791.

Li, Z., He, Q., Li, J., 2024. A survey of deep learning-driven architecture for predictive maintenance. Eng. Appl. Artif. Intell. 133, 108285. http://dx.doi.org/10.1016/j.engappai.2024.108285.

Liu, Y., Yeoh, J.K.W., 2021. Robust pixel-wise concrete crack segmentation and properties retrieval using image patches. Autom. Constr. 123, 103535. http://dx.doi.org/10.1016/j.autcon.2020.103535.

Livieris, I.E., Pintelas, E., Kiriakidou, N., Pintelas, P., 2023. Explainable image similarity: Integrating siamese networks and grad-CAM. J. Imaging 9 (10), http://dx.doi.org/10.3390/jimaging9100224, URL https://www.mdpi.com/2313-433X/9/10/224.

Luo, J., Zhao, J., Sun, Y., Liu, X., Yan, Z., 2023. Bolt-loosening detection using vision technique based on a gray gradient enhancement method. Adv. Struct. Eng. 26 (4), 668–678. http://dx.doi.org/10.1177/13694332221122950.

Luo, J., Zhao, J., Xie, C., Sun, Y., Liu, X., Yan, Z., 2024. Image-based bolt-loosening detection using an improved homography-based perspective rectification method. J. Civ. Struct. Heal. Monit. 14 (3), 513–526. http://dx.doi.org/10.1007/s13349-023-00722-4.

Mansouri, T., Vadera, S., 2022. A deep explainable model for fault prediction using IoT sensors. IEEE Access 10, 66933–66942. http://dx.doi.org/10.1109/ACCESS.2022.3184693.

Mushtaq, F., Ramesh, K., Deshmukh, S., Ray, T., Parimi, C., Tandon, P., Jha, P.K., 2023. Nuts and bolts: YOLO-v5 and image processing based component identification system. Eng. Appl. Artif. Intell. 118, 105665. http://dx.doi.org/10.1016/j.engappai.2022.105665.

Nash, W., Powell, C., Drummond, T., Birbilis, N., 2020. Automated corrosion detection using crowdsourced training for deep learning. Corros. 76.

Pan, X., Tavasoli, S., Yang, T.Y., 2023. Autonomous 3D vision-based bolt loosening assessment using micro aerial vehicles. Computer- Aided Civ. Infrastruct. Eng. 38 (17), 2443–2454. http://dx.doi.org/10.1111/mice.13023.

Pan, X., Yang, T., 2023. 3D vision-based bolt loosening assessment using photogrammetry, deep neural networks, and 3D point-cloud processing. J. Build. Eng. 70, 106326. http://dx.doi.org/10.1016/j.jobe.2023.106326, URL https://www.sciencedirect.com/science/article/pii/S2352710223005053.

Pitts, W., McCulloch, W., 1947. How we know universals the perception of auditory and visual forms. Bull. Math. Biophys. Vol. 9, 127–147.

Ramana, L., Choi, W., Cha, Y.-J., 2019. Fully automated vision-based loosened bolt detection using the viola–jones algorithm. Struct. Heal. Monit. 18 (2), 422–434. http://dx.doi.org/10.1177/1475921718757459.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, IEEE Computer Society, Los Alamitos, CA, USA, pp. 779–788. http://dx.doi.org/10.1109/CVPR.2016.91, URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.91.

Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39 (06), 1137–1149. http://dx.doi.org/10.1109/TPAMI.2016.2577031.

Retzlaff, C.O., Angerschmid, A., Saranti, A., Schneeberger, D., Röttger, R., Müller, H., Holzinger, A., 2024. Post-hoc vs ante-hoc explanations: xAI design guidelines for data scientists. Cogn. Syst. Res. 86, 101243. http://dx.doi.org/10.1016/j.cogsys.2024.101243, URL https://www.sciencedirect.com/science/article/pii/S1389041724000378.

Rodríguez-Rodríguez, J.A., López-Rubio, E., Ángel-Ruiz, J.A., Molina-Cabello, M.A., 2024. The impact of noise and brightness on object detection methods. Sensors 24 (3), http://dx.doi.org/10.3390/s24030821.

Schroff, F., Kalenichenko, D., Philbin, J., 2015a. FaceNet: A unified embedding for face recognition and clustering. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, IEEE, http://dx.doi.org/10.1109/cvpr.2015.7298682.

Schroff, F., Kalenichenko, D., Philbin, J., 2015b. FaceNet: A unified embedding for face recognition and clustering. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 815–823. http://dx.doi.org/10.1109/CVPR.2015.7298682.

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: 2017 IEEE International Conference on Computer Vision. ICCV, pp. 618–626. http://dx.doi.org/10.1109/ICCV.2017.74.

Sinha, D., El-Sharkawy, M., 2019. Thin MobileNet: An enhanced MobileNet architecture. In: 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference. UEMCON, pp. 0280–0285. http://dx.doi.org/10.1109/UEMCON47517.2019.8993089.

Wang, F., Song, G., 2020. Monitoring of multi-bolt connection looseness using a novel vibro-acoustic method. Nonlinear Dynam. 100 (1), 243–254. http://dx.doi.org/10.1007/s11071-020-05508-7.

Wang, T., Song, G., Liu, S., Li, Y., Xiao, H., 2013. Review of bolted connection monitoring. Int. J. Distrib. Sens. Networks 9 (12), 871213. http://dx.doi.org/10.1155/2013/871213.

Wang, X.Z., Zhang, Y., Niannian, 2018. Bolt loosening angle detection technology using deep learning. Struct. Control. Heal. Monit. URL https://onlinelibrary.wiley.com/doi/full/10.1002/stc.2292.

Weiss, K., Khoshgoftaar, T., Wang, D., 2016. A survey of transfer learning. J. Big Data 3.

Wu, Z.-Q.Z., Zheng, P., Xu, S.-T., Xindong, 2019. Object detection with deep learning: A review. IEEE Trans. Neural Networks Learn. Syst. 30 (11), 3212–3232.

Yang, X., Gao, Y., Fang, C., Zheng, Y., Wang, W., 2022. Deep learning-based bolt loosening detection for wind turbine towers. Struct. Control. Heal. Monit. 29 (6), e2943. http://dx.doi.org/10.1002/stc.2943.

Yu, L., Yang, E., Luo, C., Ren, P., 2021. AMCD: an accurate deep learning-based metallic corrosion detector for MAV-based real-time visual inspection. J. Ambient. Intell. Humaniz. Comput..

Yuan, C., Chen, W., Hao, H., Kong, Q., 2021. Near real-time bolt-loosening detection using mask and region-based convolutional neural network. Struct. Control. Heal. Monit. 28 (7), e2741. http://dx.doi.org/10.1002/stc.2741.

Yuan, C., Wang, S., Qi, Y., Kong, Q., 2022. Automated structural bolt looseness detection using deep learning-based prediction model. Struct. Control. Heal. Monit. 29 (3), e2899. http://dx.doi.org/10.1002/stc.2899.

Zhang, Y., Li, W., Wang, Y., Wang, Z., Li, H., 2022. Beyond classifiers: Remote sensing change detection with metric learning. Remote. Sens. 14 (18), URL https://www.mdpi.com/2072-4292/14/18/4478.

Zhang, Y., Sun, X., Loh, K.J., Su, W., Xue, Z., Zhao, X., 2019a. Autonomous bolt loosening detection using deep learning. Struct. Heal. Monit..

Zhang, Y., Sun, X., Loh, K.J., Su, W., Xue, Z., Zhao, X., 2020. Autonomous bolt loosening detection using deep learning. Struct. Heal. Monit. 19 (1), 105–122. http://dx.doi.org/10.1177/1475921719837509.

Zhang, Y., Zhao, X., Sun, X., Su, W., Xue, Z., 2019b. Bolt loosening detection based on audio classification. Adv. Struct. Eng. 22 (13), 2882–2891. http://dx.doi.org/10.1177/1369433219852565.

Zhao, Y., Yang, Z., Xu, C., 2022. Bolt loosening detection for a steel frame multi-story structure based on deep learning and digital image processing. In: ASME International Mechanical Engineering Congress and Exposition, Volume 3: Advanced Materials: Design, Processing, Characterization and Applications; Advances in Aerospace Technology, V003T04A017. http://dx.doi.org/10.1115/IMECE2022-94786.

Zhao, X., Zhang, Y., Wang, N., 2019. Bolt loosening angle detection technology using deep learning. Struct. Control. Heal. Monit. 26 (1), e2292. http://dx.doi.org/10.1002/stc.2292.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A., 2016. Learning deep features for discriminative localization. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, IEEE Computer Society, Los Alamitos, CA, USA, pp. 2921–2929. http://dx.doi.org/10.1109/CVPR.2016.319, URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.319.