2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)

Automatic Damage Detection of Fasteners in Overhaul Processes

Sajjad Taheritanjani *
Department of Informatics
Technical University of Munich
Garching, Germany
sajjad.taheri@tum.de

Ralf Schoenfeld *
Development Consultant IoT & Data Science
SAP Innovation Services
Walldorf/Baden, Germany
ralf.schoenfeld@sap.com

Bernd Bruegge
Department of Informatics
Technical University of Munich
Garching, Germany
bruegge@in.tum.de

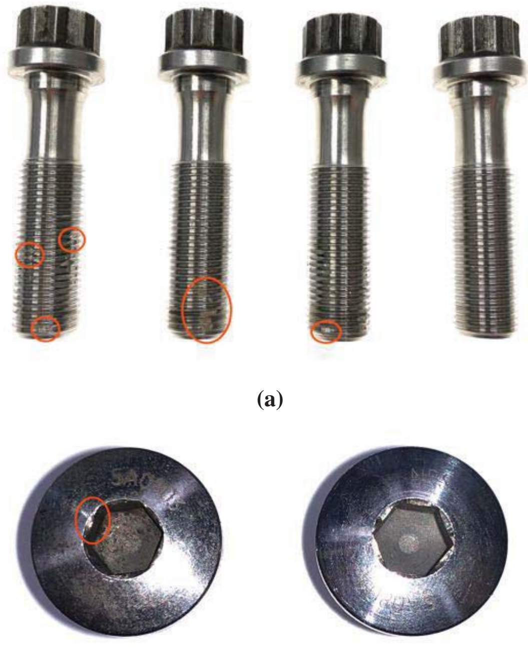
***Abstract*—Commercial aircraft engines have a maintenance process that includes overhauling approximately every six years. Hundreds of different components must be disassembled, checked, repaired (if necessary), and then reassembled. This includes undoing fasteners, cleaning, checking, refitting, and tightening them. Prior to refitting the fasteners, they must be checked for damages. In this paper, we propose an automatic damage inspection of the fasteners, using computer vision and machine learning. We built a setup to automatically record and preprocess the data and compared multiple supervised and unsupervised machine learning models for detecting damages of 12 different fasteners. Using our automatic approach, we can determine the type of fastener, its status (damaged or intact) and visualize the anomalies to aid the understanding of the decisions of the automatic detection. This can be the first step towards a fully automated fastener damage detection in overhaul processes.**

***Index Terms*—Computer Vision for Automation, Automated Damage Detection and Inspection, Fasteners Damage Detection, Overhaul Processes**

I. Introduction

The commercial airplane industry is under strict safety regulations to ensure the safety of all passengers. In 2002, the European Union established the European Aviation Safety Agency (**EASA**), which carries out certification, regulations and standardization within Europe. EASA also published a set of regulations in 2002 [1] including operational requirements for all aviation companies within Europe. The regulation states that an airplane must be in airworthy condition and the maintenance is performed in accordance with its maintenance program.

Overhauling is a part of the maintenance program that consists of disassembling, inspecting and reassembling the components to ensure that each part is in airworthy condition. Due to the fact that each company develops its own maintenance program, there is no single standard process for overhauling. Nevertheless, we can generalize how the main workflow of the overhaul process is conducted. During the overhaul process the components of an airplane are disassembled. The disassembled components consist of parts of various sizes such as panels from the wing or a propeller, as well as fasteners such as screws, bolts, washers, and nuts that hold different parts together.



(a)

(b)

Figure 1: Examples of intact and damaged bolts from a side view (a) and top view (b). The annotated damages show either scratches or dirt which could not be cleaned by washing.

The overhaul process is done manually by a human workforce. After disassembling all the fasteners from a component and washing off dirt and oil from normal operation, they can be inspected for damages. Afterwards, they are placed in a container and brought to special workstations in maintenance checking plants with access to magnifiers and illumination with which the technicians can inspect them for all kind of damages, including oxidation, scratches, missing or flaked off coating and erosion. Figure 1 shows examples of the damages on two sample bolts. There are no comprehensive defined rules to identify fasteners as damaged. Although

* Both authors have contributed equally to this work.

non-destructive methods like fluorescent penetrant testing or magnetic particle testing are used to identify cracks and voids in bigger components[1], applying them for all the fasteners could be expensive and unrealistic, considering their smaller size. Since most of the aero-engine fasteners are made of resistant alloys, like steel or titanium, and are manufactured under specific considerations [2], they can be expensive and technicians must check each of them by looking and touching, in which they may decide to keep or discard them. Typically, the technicians might need up to 15 seconds to inspect a fastener properly in their workstations. Furthermore, the nature of aero-engine fasteners introduces problems like reflection and shadow [3] which makes the task of damage detection challenging. Therefore, an automatic damage detection process would save time and reduce costs during overhaul processes.

The contributions of this paper is twofold: 1) we propose a system that employs vibrated conveyor belt and polarized lighting to create datasets for damages of fasteners, and 2) we compare different machine learning models for detecting damages of fasteners. To the best of our knowledge, there is no published study that detects damages of different kinds of fasteners in overhaul processes. Our study can be also applicable to other small reflective objects in other industrial fields.

The rest of this paper is structured as follows: Section II describes the related work; the dataset properties are stated in Section III; Section IV discusses the methods we used to obtain and preprocess the data and the algorithms used in training the damage detection models; our results are presented in Section V. Furthermore, Section VI presents the implications of our results and discussion, and finally, Section VII summarizes our contributions.

1

II. Related Work

Numerous research studies have been conducted to detect the defect and damages for both industrial and non-industrial purposes. In 2012 Masci et al. presented a convolutional neural network (CNN) approach for supervised steel defect classification in the steel strip market [4]. They classified 7 different defect types collected from a real production line, with an error rate of 7%, which outperformed a support vector machine (SVM) approach with various feature extractors. In 2016, Park et al. also used a CNN for automatic surface defect inspection with an accuracy near to the human inspection [5]. In 2018, Cha et al. performed an automated visual inspection on civil infrastructures to increase safety [6]. A Faster R-CNN was trained to identify objects with the following classes: medium steel corrosion, steel delamination, high steel corrosion, concrete cracks and bolt corrosion. The first features of the input image were extracted with a CNN, the object/region proposals were made from these features, and finally the proposals were classified and the bounding box was fitted using a regressor.

In 2015, Giben et al. described an approach to automatically monitor railway tracks to ensure passenger safety [7]. Cameras mounted on a wagon generate large volumes of images for visual inspection. The visual inspection was automated with a CNN for semantic segmentation. It could distinguish between the material categories ballast, wood, rough concrete, medium concrete, smooth concrete, crumbling concrete, chipped concrete, lubricator, rail, and fastener. With the resulting pixel-wise segmentation they could identify damages and also evaluate the number of damages occured. Semantic segmentation has also been used to detect defects in fabrics [8], fruits [9], and asphalt on roads [10].

In 2018, Ferguson et al. et al. trained a Mask R-CNN on the GDXray dataset [11] containing X-ray images with annotated casting defects [12]. The authors in [13] presented an approach to detect damages for product printings. It is worth noting that the algorithm is a combination of raw pixel comparison of neighboring pixels and edge detection. First, the edges are identified using a Laplacian edge detector, using the first and second derivative in one dimension finding contrast changes. With a threshold, a binary mask of edges is created. The difference in the binary mask, as well as the pixel difference in non-edge patches of pixels in the image for the input is compared to the reference image. A threshold determines if there is a defect.

In [14], the authors employed multiple variants of autoencoder architectures to detect anomalies from magnetic resonance images. The general idea is to train the autoencoder to reconstruct the input and detect anomalies by subtracting the reconstruction from the input. Unseen data cannot be reconstructed by the autoencoder because it can lead to a higher reconstruction error. Furthermore, the segmentation is achieved by thresholding the reconstruction error of the autoencoder in [14]. They used normal autoencoders, variational autoencoders, and variational autoencoders with the decoder part trained like an AnoGAN [15].

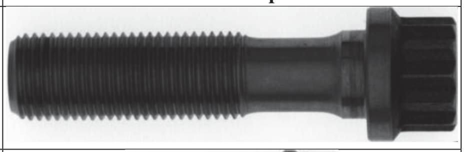
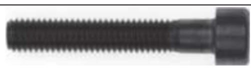
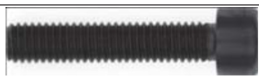
These studies inspire the current work; however, the characteristics of the fasteners [3] make this study different from others. In this work, we propose a system to address these characteristics and compare a set of damage detection models for fasteners using different approaches. This comparison can be used as a benchmark in training damage detection models for fasteners or other small reflective objects in industrial fields.

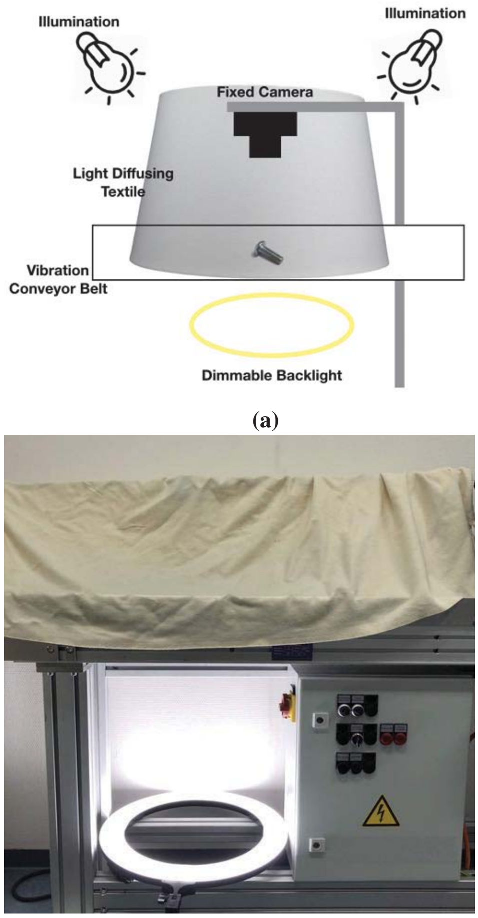
III. Dataset

Our fasteners dataset consists of 2019 images of 12 different bolt types each with intact and damaged samples. Table I shows examples of these different types of fasteners. The bolts of type AS 31532 and AS 21514 are specific airplane engine fasteners. The other bolts M5 12, M5 20, M5 30, M5 40, M5 50, M6 16, M6 20, M6 30, M6 40, and M6 50 are standard bolts. They have metric ISO-threads (M) with a diameter of 5 and 6 millimeters and a length ranging from 12 to 50 millimeters. The dataset consists of 4 different instances per status of each fastener for the standard types. The AS 31532 bolt is represented with 2 damaged and 2

[1]https://power.mtu.de/engineering-and-manufacturing/aero-solutions/special-processes/

Table I: Types of fasteners in the dataset

Type	Intact Sample
AS 31532	
AS 21514	
M5_12	
M5_20	
M5_30	
M5_40	
M5_50	
M6_16	
M6_20	
M6_30	
M6_40	
M6_50	



(a)

(b)

Figure 2: Sketch of the setup with a textile vibrating conveyor belt to enable backlighting and light diffusing textile above it (a) and its realization (b).

intact instances while the AS 21514 is presented in the dataset with 1 damaged instance and 2 intact instances. In total the dataset consists of 87 instances of fasteners.

To assure that the model generalizes to the all kinds of damages and not specific instances of them, the model is tested on a holdout set of damaged and intact instances that it has not seen before. For each of the standard types an instance of an intact and an additional damaged fastener - not in the training dataset - is used in the holdout set. The test dataset consists of 207 pictures of damaged fasteners and 213 intact fasteners. The rest of the images are used for training and validation, with a random split of 75% for training and 25% for validation.

For the instance segmentation of the damages, we manually created the fine-grained annotations. The instances of damages are annotated with a mask marking the damaged area. In total, we annotated 373 images of the fasteners with 2438 masks defining regions of damages (2065 masks) and the area of the fasteners (373 masks). The instance segmentation dataset is also split into a training dataset with 248 images and 1720 annotations as well as a validation dataset with 125 images and 718 annotations.

IV. Methods

Considering the challenges faced in the fine-grained visual categorization of fasteners [3], perspective, reflection, and shadow with normal ambient lighting as well as the shape, threads, and head features on which we want to focus, we propose a set of methods to preprocess the data, create damages, and train the models.

A. Dataset Creation and Preprocessing Steps

We automated the process of recording the images for the dataset by passing the fasteners on a backlight supported conveyor belt under the camera [3]. To ensure that one fastener arrives under the camera at a time, we separate them by vibration. We also preprocess the recorded images to improve the detection performance of the damage detection algorithms.

Figure 2a and Figure 2b show the sketch of the setup and its realization. Figure 3 depicts the image preprocessing steps. First, to separate the captured image from the background, we apply either a simple threshold for a uniform background, i.e.,

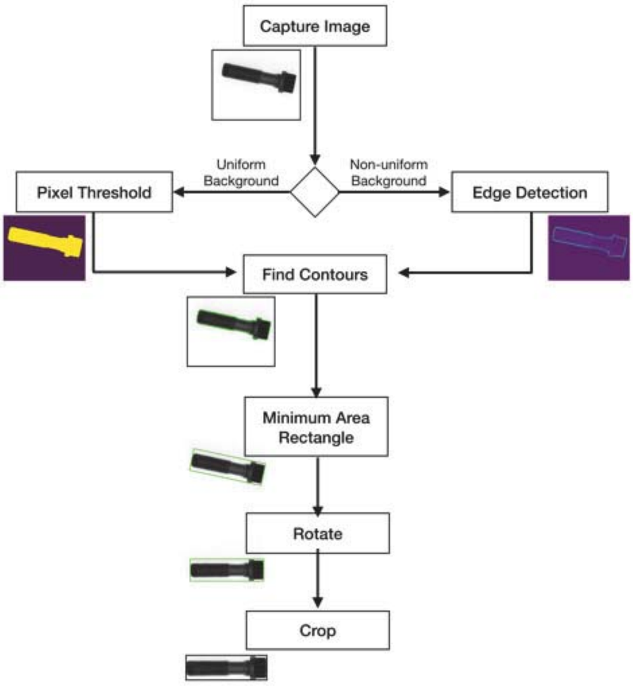


Figure 3: Preprocessing steps to obtain the unlabeled/non-annotated data. Depending on the background of the image, we perform pixel thresholding or edge detection to obtain the fastener mask. Subsequently, we find the contours, apply the minimum area rectangle, rotate and crop the image before saving it on the disk.

using a binary threshold for the background color in OpenCV, or an edge detection for a non-uniform background. In edge detection step, we blur the image using Gaussian Blur in OpenCV for noise robustness. Afterwards, using the Sobel filter in OpenCV, we apply a filter to detect edges. From the resulting pixel, the threshold mask or the edges of the outer contours are identified. To locate the fastener, we use the minimum area rectangle of the outer contour. Finally, the image of the fastener is rotated and cropped and the final result is a rectangular image just fitting the fastener.

B. *Algorithms*

We trained different machine learning models (both supervised and unsupervised) to detect the damages. Supervised tasks are the classification of the type of the fastener and the instance segmentation of damages, i.e., they determine if a fastener is intact or damaged. The unsupervised task is anomaly detection with anomalies representing damages. Figure 4 shows an overview of the supervised and unsupervised approaches in this study.

We also used the *Blackboard* pattern to combine multiple interdependent damage detection algorithms. *Blackboard* stores solutions and intermediate results that are accessed by the knowledge sources. Knowledge sources query the *Blackboard* for information that is provided by other knowledge sources. Table II shows the knowledge sources and Figure 5 depicts the decision process that models the interdependencies between the different knowledge sources and their corresponding inputs and outputs.

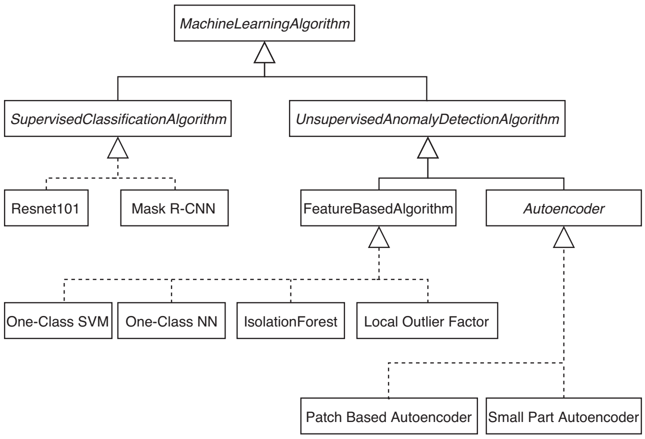


Figure 4: The overview of the algorithms which are used in this study.

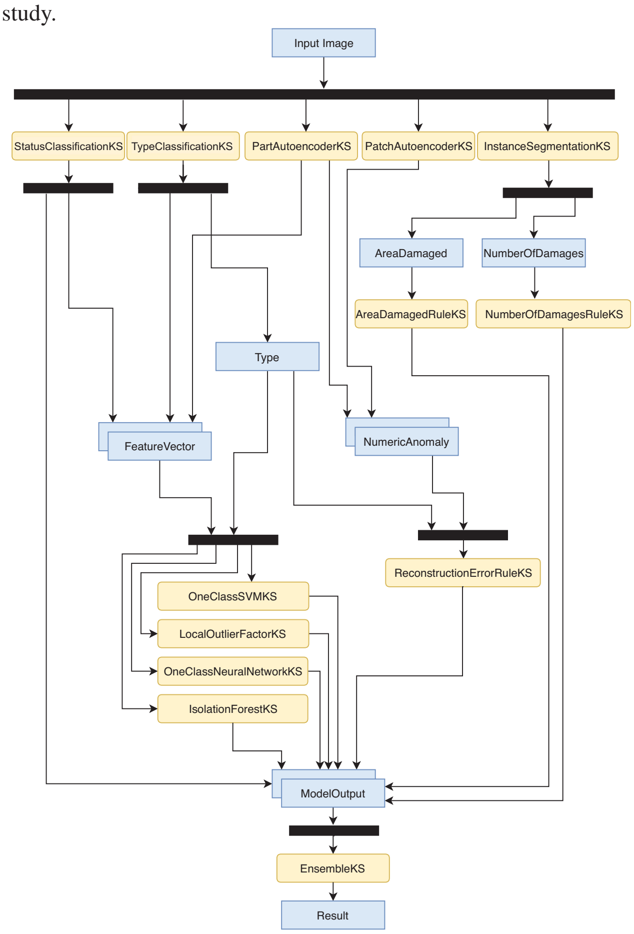


Figure 5: The decision process workflow using different knowledge sources (in yellow) and the inputs/outputs (in blue).

V. Results

The models are trained using a Titan Xp GPU donated by the NVIDIA Corporation. The results are divided into supervised and unsupervised approaches. Supervised approaches are the classification of the type of the fastener and the

Table II: Knowledge sources that are used for damage detection with their input and output sources. Feature Vector: lower dimensional representation, Area Damaged: percentage of the fastener which is damaged, Numeric Anomaly: numeric value representing how anomalous a fastener is according to a damage detection algorithm, Model Output: output of a model determining whether the fastener is damaged or intact, Result: output of the blackboard representing the final decision.

Knowledge Source	Input	Output	Algorithm
TypeClassificationKS	Input Image	Type, Feature Vector	Resnet101
StatusClassificationKS	Input Image	Model Output, Feature Vector	Resnet101
InstanceSegmentationKS	Input Image	Number of Damages, Area Damaged	Mask R-CNN
PatchAutoencoderKS	Input Image	Numeric Anomaly	Patch-based Autoencoder
FastenerAutoencoderKS	Input Image	Numeric Anomaly, Feature Vector	Autoencoder
OneClassSVMKS	Feature Vector, Type	Model Output	One-class SVM
IsolationForestKS	Feature Vector, Type	Model Output	Isolation Forest
LocalOutlierFactorKS	Feature Vector, Type	Model Output	Local Outlier Factor
OneClassNeuralNetworkKS	Feature Vector, Type	Model Output	One-class Neural Network
NumberOfDamagesRuleKS	Number of Damages	Model Output	Rule-based
AreaDamagedRuleKS	Area Damaged	Model Output	Rule-based
ReconstructionErrorRuleKS	Numeric Anomaly, Type	Model Output	Rule-based
EnsembleKS	Model Output	Result	Ensemble

Table III: Supervised classification performance metrics test dataset

Algorithm	Accuracy	Precision	Recall	F1-Score	Training Details
Resnet101	0.99	0.99	0.99	0.99	epochs=8, batch_size=8, lr=0.001
Mask R-CNN (Number of Damages Rule)	0.743	0.99	0.483	0.649	epochs=32, batch_size=2, lr=0.001
Mask R-CNN (1% Area Rule)	0.743	0.99	0.483	0.649	epochs=32, batch_size=2, lr=0.001
Mask R-CNN + Resnet101	0.988	0.986	0.99	0.988	-

instance segmentation of damages, i.e., if a fastener is intact or damaged, while the unsupervised approaches are based on anomaly detection.

A. Supervised Approaches

The first supervised approach is the classification that determines if a fastener is damaged or intact - damaged/intact classification. Using a Resnet101, all damaged and intact fasteners are correctly classified for the training dataset. In the test dataset, consisting of the damages the model has not seen before, the performance drops slightly. 99% of the damaged fasteners are correctly identified as damaged as well as 99% of the intact fasteners are correctly classified as intact.

The Mask R-CNN trained on the detection of damage instances achieved an mAP at an IoU of 50 of 0.5563 for the task of object detection on the validation dataset with fine-grained annotations. Combining the results from Mask R-CNN with simple rules (damaged area covers more than 1% of the whole image and number of damaged objects is greater than zero) can also output a damage detection model. Table III shows the performance metrics for the supervised classification algorithms. The rules have the same precision with lower recall values, due to a lower number of true positives. Combining the Resnet101 result with Mask R-CNN (considers a fastener to be damaged if either the Resnet101 or the Mask R-CNN determines it to be damaged) decreases the true negatives and precision but increases the generalizability since damage objects of unknown fasteners can also be identified.

Figure 6 visualizes the Resnet101 output for a sample damaged fastener using Grad-CAM [16] and Figure 7 shows an example of the Mask R-CNN result.

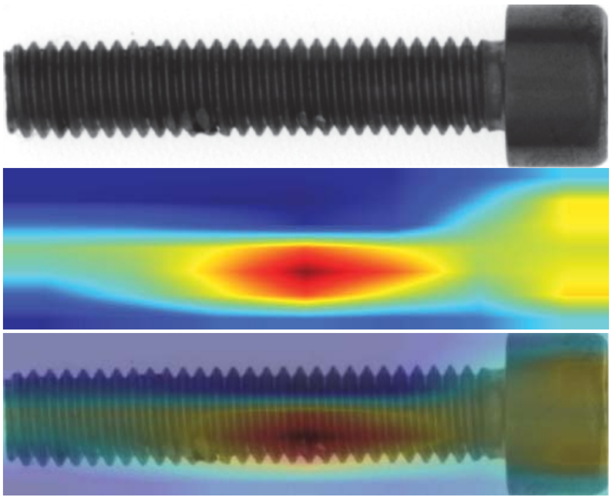


Figure 6: An example of the heatmap visualization for a damaged fastener. Low activation is shown with cold colors (namely blue). The colder area indicates less anomalies in that part of the image.

Figure 7: An example of Mask R-CNN result for the damages including their masks and bounding boxes.

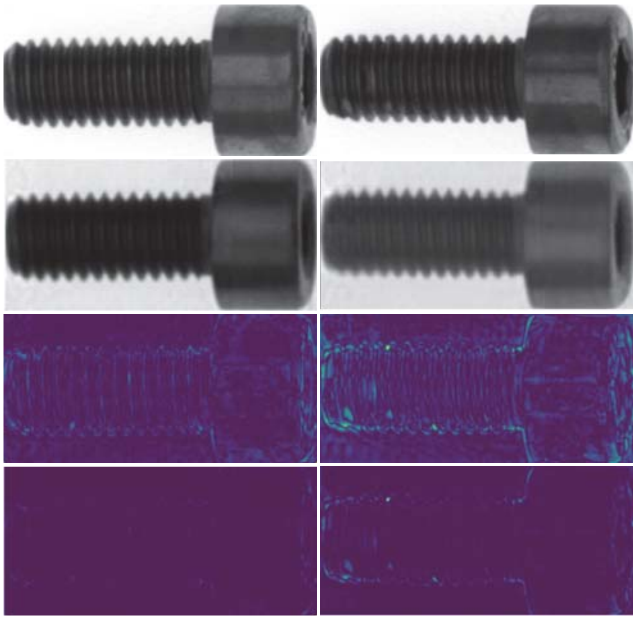
Table IV: Unsupervised classification performance metrics test dataset.

Algorithm	Feature Extractor	Accuracy	Precision	Recall	F1-Score	Training Details
One-class Support Vector Machines	ResNet101	0.63	0.59	0.82	0.69	kernel="rbf", nu=0.1, gamma=1/8192
One-class Support Vector Machines	Autoencoder	0.55	0.57	0.35	0.44	kernel="rbf", nu=0.01, gamma=1/8192
Isolation Forests	ResNet101	0.63	0.59	0.83	0.69	n_estimators=100, bootstrap=False
Isolation Forests	Autoencoder	0.55	0.57	0.35	0.44	n_estimators=100, bootstrap=False
Local outlier Factors	ResNet101	0.66	0.60	0.89	0.72	-
Local outlier Factors	Autoencoder	0.53	0.54	0.24	0.33	-
One-class Neural Networks	ResNet101	0.65	0.70	0.49	0.58	epochs=50, hidden_nodes_number= 32
One-class Neural Networks	Autoencoder	0.50	0.45	0.04	0.08	epochs=50, hidden_nodes_number= 32
Autoencoder Reconstruction Rule	-	0.78	0.96	0.57	0.70	epochs=40, batch_size=4, lr=0.00005
Patch Autoencoder Reconstruction Rule	-	0.84	0.97	0.64	0.77	epochs=40, batch_size=4, lr=0.00005

B. Unsupervised Approaches

The unsupervised approaches are anomaly detection with anomalies that represent damages. Table IV shows the performance metrics for the unsupervised anomaly detection methods. The anomaly detection methods operate on different extracted features from the image (Resnet101 features and autoencoder bottleneck features) and apply one-class SVMs, one-class neural networks [17], isolation forests, and local outlier factors to determine if the set of features is abnormal. Rules based on the reconstruction error of autoencoders (patch-based and for the whole fastener) trained on images of intact fasteners use a threshold to determine whether a fastener is considered to be damaged. The patch-based autoencoder rule has the highest accuracy among the unsupervised approaches with an accuracy of 84%, the true negatives of 98%, precision 97%, and F1-Score of 77%.

Figure 8a and Figure 8b show the output of the patch based autoencoder to the corresponding input as well as the reconstruction error (absolute value for subtracting the reconstruction from the input image) and the squared reconstruction error. The squared error reduces the noise and visualizes only larger damages. The patch-based autoencoder does the reconstruction on small image patches identifying errors in the texture of the fasteners.



(a) Intact fastener **(b)** Damaged fastener

Figure 8: An example of the autoencoder input (first row), its reconstruction (second row), reconstruction error (third row), and square of reconstruction error (last row) for an intact and a damaged fastener.

VI. Result Implications and Discussion

The results for the damaged/intact classification, with the precision and recall of 99%, also imply a possible usage in the overhaul process. The mAP at an IoU of 50 of 0.5563 for the damage object detection is comparable to other object detectors on the COCO dataset [18]. With the specificity of 0.995 and precision of 0.99, the instance segmentation can complement other damage detectors and provide visual insights about the damages. For a complete evaluation, the damaged/intact classification must be scaled with more samples recorded with an automated system, allowing automating the whole classification and sorting process in an overhaul plant.

Without the knowledge of damages, the task of unsupervised anomaly detection on images can be challenging. The distinction between damaged and intact fasteners is often fine-grained and affected by noise, light, the position of the fastener or other minor changes. The combination of feature extraction with one-class classifiers did not lead to satisfying results. We consider that the extracted features, especially from the autoencoders, cannot represent the damages very well. We can determine the notion of normality using autoencoders. However, their performance is still lower than the supervised methods - with an accuracy of 84% and a recall of 64% for the patch-based autoencoder paired with a rule for the reconstruction error, which could be due to the distinguishing of the concept of normality from everything else. However, a visualization of the reconstruction error provides valuable insights for human operators showing where exactly the damage might be located.

The drawback of the supervised approaches is that we need to train the models with both intact and damaged instances of the fasteners. While they offer higher accuracy, the efforts for the dataset collection and the preprocessing steps are also high and the generalizability could be affected with introducing new unseen damages. However, unsupervised approaches need only the intact instances for the training phase that can reduce the efforts for the dataset collection and the preprocessing steps.

To improve the results and the generalization of the solutions, more data and more different damages are needed. Furthermore, since the noise is an extra anomaly, constraining the data collection environment and reducing the noise can also increase the accuracy of the results. Moreover, the current images of the fasteners are from one top-down perspective. To detect the damages on all sides of the fasteners, they must automatically be rotated using vibration or another setup that uses multi-view cameras. In addition, the current dataset must be scaled to a larger number of classes. Some damages are not detectable optically and, therefore, cannot be identified using computer vision. To detect these types of damages, we can use for example other sensors and methods, as described in [19], [20] and [21].

Our dataset might not represent all the classes in the target environment. Therefore, to ensure consistent performance, more data preferably with more classes have to be recorded. Another threat to validity is the damages of the fastener in the dataset that might not represent all the cases of real damaged fasteners. To solve this issue, the data has to be collected at the target location. Currently, images are only recorded with one specific type of camera at a fixed distance. Using image processing and passing the distance/scale to the models can remediate this limitation.

VII. Conclusion

In this paper, we proposed a polarized backlighting supported system with vibration conveyor belt to automatically detect damages of fasteners in overhaul processes and recorded datasets for fastener damage detection (for 12 different bolts). Since the input size is reduced and targets are more similar using our preprocessing steps, we required less data and fewer training for the damage detection algorithms.

In addition, we trained different models using both supervised and unsupervised machine learning methods to detect the damages and compared the results. The best supervised and unsupervised accuracies were 99% and 84% respectively.

To the best of our knowledge, this is the first reported study to apply damage identification for different kinds of fasteners in overhaul processes. Our trials showed that it is possible to automate the data collection and preprocessing, and the damage detection of fasteners with a reasonable error rate and feasible hardware setup.

References

[1] European Aviation Safety Agency. "REGULATION (EC) No 1592/2002 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL." 2002.

[2] D. G. Teer and F. B. Salem. "The formation of low friction wear-resistant surfaces on titanium by ion plating." Thin Solid Films 45, no. 3 (1977): 583-589.

[3] S. Taheritanjani, J. Haladjian, B. Bruegge. "Fine-grained Visual Categorization of Fasteners in Overhaul Processes." 5th International Conference on Control, Automation and Robotics (ICCAR), 2019.

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

[5] J.K. Park, B.K. Kwon, J.H. Park, and D.J. Kang. "Machine learning-based imaging system for surface defect inspection." International Journal of Precision Engineering and Manufacturing-Green Technology, volume 3, 303–310, 2016.

[6] Y.J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Bueyuekoeztuerk. "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types." Autonomous structural visual inspection using region?based deep learning for detecting multiple damage types. Computer?Aided Civil and Infrastructure Engineering, 33(9), 731-747, 2018.

[7] X. Giben, V.M. Patel, and R. Chellappa. "Material classification and semantic segmentation of railway track images with deep convolutional neural networks." IEEE International Conference in Image Processing (ICIP), 621-625, 2015.

[8] L. Kara and P. Prabhu. "A review of human error in aviation maintenance and inspection." International Journal of industrial ergonomics. volume 26, 133-161, 2000.

[9] Q. Li, M. Wang, and W. Gu. "Computer vision based system for apple surface defect detection." Computers and electronics in agriculture, volume 36, 215-223, 2002.

[10] S. Sudakov, O. Barinova, A. Velizhev, and A. Konushin. "Semantic segmentation of road images based on cascade classifiers." Proceedings of the ISPRS XXI Congress, 601-604, 2008.

[11] D. Mery, V. Riffo, U. Zscherpel, G. Mondragon, I. Lillo, I. Zuccar, H. Lobel, and M. Carrasco. "GDXray: The database of X-ray images for nondestructive testing." Journal of Nondestructive Evaluation, volume=34, p. 42, 2015.

[12] M. Ferguson, R. Ak, Y.T.T. Lee, and K.H. Law. "Detection and Segmentation of Manufacturing Defects with Convolutional Neural Networks and Transfer Learning." arXiv preprint arXiv:1808.02518, 2018.

[13] W. Yangping, X. Shaowei, Z. Zhengping, S. Yue, and Z. Zhenghai. "Real-time Defect Detection Method for Printed Images Based on Grayscale and Gradient Differences." Journal of Engineering Science & Technology Review, volume 11, 2018.

[14] C. Baur, B. Wiestler, S. Albarqouni, and N. Navab. "Deep Autoencoding Models for Unsupervised Anomaly Segmentation in Brain MR Images." arXiv preprint arXiv:1804.04488, 2018.

[15] T. Schlegl, P. Seeböck, S.M. Waldstein, U. Schmidt-Erfurth, and G. Langs. "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery." International Conference on Information Processing in Medical Imaging, 146-157, 2017.

[16] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. "Grad-cam: Visual explanations from deep networks via gradient-based localization." International Conference on Computer Vision (ICCV), 618?626, 2017.

[17] R. Chalapathy, A.K. Menon, and S. Chawla. "Anomaly Detection using One-Class Neural Networks." arXiv preprint arXiv:1802.06360, 2018.

[18] J. Redmon, Joseph and A. Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767, 2018.

[19] J. E. Michaels. "Detection, localization and characterization of damage in plates with an in situ array of spatially distributed ultrasonic sensors." The journal of Smart Materials and Structures, volume 17, 035035, 2008.

[20] M.L. Fugate, H. Sohn, and C.R. Farrar. "Vibration-based damage detection using statistical process control" The journal of Mechanical Systems and Signal Processing, volume 15, 707-721, 2001.

[21] C.C. Ciang, J.R. Lee, and H.J. Bang. "Structural health monitoring for a wind turbine system: a review of damage detection methods." The journal of Measurement Science and Technology, volume 19, 122001, 2008.