COMPUTER VISION ALGORITHMS FOR THE IDENTIFICATION OF DAMAGES ON FULL-SCALE AIRCRAFT COMPONENTS

S. Merola[1], M. Guida[1], F. Marulo[1]

[1]Department of Industrial Engineering, University of Naples Federico II, Italy

Abstract

The objective of this study is to present a novel approach for airplane inspection to identify skin deterioration on the fuselage. Algorithms for computer vision were used as an instrument for automating the process of inspection and detection, decreasing human error, and increasing productivity and security. An overview of the problems, methods, and recent developments in the field of computer vision algorithms used for general damage detection on aircraft components is provided in this research work. Data were collected using a high-quality acquisition system. The data set was created by gathering photographs to highlight different sorts of defects and have the greatest possible variety of instances, images collected on two separate aeronautical demonstrations: a commercial partial full-scale aircraft fuselage section in primer paint and a general aviation aircraft fuselage white painted. In particular, 964 images and more than 6000 regions of interest were manually annotated. Datasets that accurately represent various types of damages can be limited, making it difficult to train accurate and reliable models. The Convolutional Neural Networks and machine learning models were trained on large datasets of annotated images, enabling them to learn complex patterns and features associated with different types of damage. Data augmentation techniques were adopted to add diversity to the training data. Transfer learning techniques, which leverage pre-trained models on large-scale image datasets, have also proved to be effective in achieving accurate and robust detection results.

Keywords: Aircraft Maintenance, Image Processing, Computer Vision, Convolutional Neural Network

1. Introduction and state of the art

The aviation industry is witnessing a transformative shift due to the integration of advanced technologies, which are fundamental for ensuring flight safety and operational efficiency. One crucial aspect of this change is aircraft maintenance, which is responsible for the prompt and precise identification of any aircraft defects or anomalies. In the realm of computer vision [1] and deep learning [2], the Convolutional Neural Network (CNN) [3] has shown remarkable potential in object identification and classification within images [4] [5]. However, applying CNN directly to aircraft maintenance may necessitate vast amounts of labeled data and substantial computational resources. Fine-tuning technique [6] arises as a promising solution in this context, enabling the utilization of knowledge gained from pre-trained neural networks on large, generic datasets to enhance the new architecture's performance in aircraft maintenance. By tuning the learned features from a pre-trained network to a specific problem, the fine-tuning technique reduces the need for extensive labeled datasets and improves training efficiency. This research employs two full-scale fuselage sections located in the San Giovanni campus laboratory of the Department of Industrial Engineering at the University of Naples Federico II. The ultimate objective was to develop an efficient, accurate, and neural network-based aircraft maintenance system capable of supporting operators in preventive and corrective maintenance processes.

1.1 Classical aircraft maintenance process

The aircraft maintenance programs (back in the 1960s) were based on conservative utilization and overhaul limits. This approach prioritized high safety margins but was financially costly. To address this issue, a more suitable maintenance logic and interval system was needed. As a result, Maintenance Steering Groups (MSG) [7] were formed, comprising operators and engineers who could develop MSG methodologies. These methodologies have evolved, from MSG-1 to MSG-3 [8] [9] [10]. Three significant methodologies have been established since the creation of MSG:

- MSG-1: This methodology was developed and implemented for the Boeing 747-100. It introduced three maintenance processes for various aircraft components and structures:
  - Hard Time (HT): A preventive maintenance process that necessitates the continuous inspection or removal of a component from service.
  - On Condition (OC): A preventive maintenance process that requires a component to be inspected according to specific standards. These standards can be adjusted based on experience or specific tests conducted, and they determine whether the component can remain in service or should be removed.
  - Condition Monitoring (CM): A maintenance process that uses appropriate means available to the operator to detect and resolve problems in certain areas of the inspected aircraft.

- MSG-2: This methodology was developed and applied to the DC-10. It is based on a more extensive and generic logic while still maintaining the division into Hard Time, On Condition, and Condition Monitoring.

- MSG-3: This methodology is based on the specific function of aircraft components and the consequences of potential failures. It consists of several phases:
  - Determining if a component can be classified as "critical" (Maintenance Significant Item - MSI) and deciding the type of maintenance process to apply.
  - MSG-3 Analysis Level 1: Evaluating the functional failure and its potential consequences to assign each critical component a category suggesting the applicability of an intervention.
  - MSG-3 Analysis Level 2: Identifying possible maintenance interventions, which may include lubrication, visual inspections, or functional checks.

For structures, the logical path extends to the entire system to be inspected. Structural Significant Items (SSI) [11] [12] are identified, and categories are assigned based on the type of damage (fatigue, environmental deterioration, or accidental). The most effective inspection approach is then selected, which might be a general visual examination, detailed inspection, or particularly detailed inspection using specialized procedures and non-destructive testing (NDT) equipment.

1.2 New aircraft maintenance process

The methodology presented in this paper aims to create a technological solution that can identify and validate new procedures for inspecting aircraft fuselages using image processing technology [13]. Images are captured by various acquisition systems, including reflex cameras, drones, and commercial cameras. The initial trade-off phase involved taking images in different conditions and with different cameras to estimate prediction differences in the final testing phase. A small remote-controlled quadcopter drone can be used to access difficult-to-reach areas of the fuselage and take images of the interested areas, reducing maintenance operation time and enhancing operator safety. The combined use of artificial intelligence and cameras for aircraft fuselage inspections allows for near-real-time image analysis using machine learning and computer vision algorithms. Operators review images received from drones through a Graphical User Interface (GUI). The project has wide-ranging applications, from maintenance and quality checks to pre-flight inspections, with a focus on monitoring the health status of structures. The manual identification, mapping, analysis, and technical resolution of defects on the entire external surface of an aircraft is a time-consuming process for operators. Defects may or may not have a structural impact, so their identification and classification are crucial for ensuring the safety of the aircraft throughout its operational life. Development activities have focused on defects that can be found on the external surface of aircraft, particularly large ones. These defects present a significant challenge for maintenance personnel due to the size of the aircraft, which can be tens of meters in height and hundreds of meters in length. As a result, multiple operators are required to scan the entire surface.
To classify the damages, the manuals of the Airbus A320 family aircraft [14] were used as a reference.

In these manuals, the damages are categorized as allowable, repairable, or non-repairable, or by their nature, such as scratches, dents, cracks, grooves, distortions, thinning, abrasions, and delaminations. For each type of damage, the necessary measuring instrument was described to define the extent of the damage and assess its impact. Additionally, techniques for reporting damage were illustrated, which aim to record the location, possible cause of the damage, and its relative position to other structural elements, such as rivets and structural components. These elements contribute to improving the analysis of the defect, identifying its scope, and defining its resolution. An alternative method was then proposed to achieve the same results. The description of the current state of the art is useful for defining the method and capabilities that an automatic system should and can provide as a replacement for the usual techniques. Acceptance criteria for the damage were identified based on the structural analyses carried out by the aircraft manufacturer. These criteria generate graphical maps indicating the acceptable extent of damage for each area of the aircraft, without the need for repair, and may require frequent inspections of the damage at certain intervals. Given the wide range of damages that can be found on the external surface of an aircraft, the requirements necessary to validate the automatic inspection technology for the following types were defined, as they are the most frequent and significant in terms of workload:

- Missing rivet: This is one of the most recurring and structurally significant defects in maintenance inspections. Operators need to develop a system that can quickly and efficiently identify, map, and provide indications of such defects on the entire external surface of the aircraft, reducing the workload of the operators. This type of defect also represents the first level of software recognition validation, as it is relatively simple to determine the presence or absence of an object in a photo.
- Corroded rivet: The requirements necessary for the acceptability of the integrity level of a rivet head are reported in [15].
- General damage: Examples of flow charts are provided in [16], which guide corrective actions following the identification of scratches and dents that may be found on the analyzed aircraft.

A parallel study on error in decals on aircraft components was done, to enable maintenance operators to remove damaged decals and allow the crew to perform all maintenance processes and pre-flight operations as quickly and correctly as possible.

2. Methodology and Results

The methodology that allows to performing of visual inspection operations using computer vision algorithms is based on the YOLO (You Only Look Once) architecture [17], YOLO is a real-time object detection system that is designed to identify objects in images. The YOLO architecture treats object detection as a regression problem, which makes it different from other detection systems that use a two-step process of first identifying regions of interest and then classifying those regions. In particular, YOLO architecture is composed of the following main parts:

- Input Image: The process starts with an input image that is resized to a fixed size. The image is divided into a grid, where each grid cell is responsible for detecting objects if the center of the object falls into the grid cell.
- Feature Extraction: The image is passed through a Convolutional Neural Network (CNN) for feature extraction. The CNN can be a custom one or a pre-trained model like Darknet, ResNet, GoogleNet, etc. The output of the CNN is a feature map that encodes information about the objects in the image.
- Bounding Box Predictions: For each grid cell in the feature map, the model predicts several bounding boxes and their associated class probabilities. The number of bounding boxes depends on the version of YOLO. Each bounding box is represented by five values: the x and y coordinates of the box's center, its width and height, and a confidence score that indicates how confident the model is that the box contains an object.
- Non-Max Suppression: To eliminate duplicate detections, YOLO uses a method called non-max suppression. This method selects the bounding box with the highest confidence score and suppresses the others that have an Intersection over Union (IoU) greater than a certain threshold.
- Final Output: The final output is a set of bounding boxes with their associated class labels and confidence scores. The boxes are drawn on the original image, and the class labels and

confidence scores are displayed next to each box.

In particular, for this paper, YOLOv8 architecture was selected. YOLOv8 was released in January 2023 by Ultralytics [18] [19] and supports multiple vision tasks such as object detection, segmentation, pose estimation, tracking, and classification. This architecture uses a similar backbone as the previous YOLOv5 with some changes on the CSPLayer now called the C2f module; it is divided into three main parts:

- Backbone: The backbone is responsible for feature extraction from the input image. YOLOv8 uses a modified Cross Stage Partial Network 53 (CSPDarknet53) as its backbone. CSPDarknet53 is a variant of Darknet, which is designed to reduce computation while maintaining accuracy. It divides the input feature map into two parts, one going through a series of convolutional layers and the other being directly connected to the output. This architecture helps to reduce the computational cost and improve the model's ability to learn more complex features.
- Neck: The neck is the part of the architecture that is responsible for combining features from different layers of the backbone network. In YOLO, the neck typically consists of Path Aggregation Network (PANet) or Feature Pyramid Network (FPN) structures. These structures help in aggregating features from different scales and levels, allowing the model to detect objects of various sizes and resolutions.
- Head: The head is the final part of the YOLO architecture that is responsible for performing the actual object detection task. It takes the features extracted and fused by the backbone and neck and applies bounding box prediction and classification. In YOLO, the head predicts the objectness score, class probabilities, and bounding box coordinates for each object detected in the image. The head is designed to process the features from the neck and output the final detection results.

YOLOv8 utilizes an anchor-free architecture with a decoupled head to manage objectness, classification, and regression tasks independently. This structure enables each branch to concentrate on its specific task, thereby enhancing the overall accuracy of the model. In the output layer of YOLOv8, the sigmoid activation function is employed for the objectness score, which signifies the probability of a bounding box containing an object. The SoftMax function is used for class probabilities, indicating the likelihood of an object belonging to each possible class. YOLOv8 incorporates CIoU and DFL loss functions for bounding box loss and binary cross-entropy for classification loss. These loss functions have demonstrated improved object detection performance, especially when handling smaller objects.

Furthermore, YOLOv8 presents a semantic segmentation model, YOLOv8-Seg, which utilizes a CSPDarknet53 feature extractor as the backbone, followed by a C2f module in place of the conventional YOLO neck architecture. The C2f module is succeeded by two segmentation heads that learn to predict semantic segmentation masks for the input image. The YOLOv8-Seg model features detection heads similar to YOLOv8, consisting of five detection modules and a prediction layer. The YOLOv8-Seg model has achieved state-of-the-art results on various object detection and semantic segmentation benchmarks while maintaining high speed and efficiency.

YOLOv8 can be executed from the command line interface (CLI) or installed as a PIP package. Additionally, it offers multiple integrations for labeling, training, and deployment.

2.1 Damage detection algorithm set-up procedure

This paragraph presents the methodology for setting up the algorithm to identify, locate, and classify damage on skin fuselage panels. In particular, the process of building the database, the characteristics of the test facility equipped to collect photos of defects, the setting of the training parameters of the YOLO v8 architecture, and the results coming from the testing phase were illustrated.

2.1.1 Test facility and database description for damage detection

To build a useful database for the training phase of the selected architecture, a facility was made up in the laboratory of the Department of Industrial Engineering at the University of Naples Federico II. Specifically, a trade-off analysis (Table 1*)* was performed between three different acquisition systems:

- Smartphone (with a resolution of 12 MP)
- Drone (with a resolution of 4 MP)
- Reflex (with a resolution of 24 MP)

In particular, the three different acquisition systems were tested at various distances from the object to be inspected to simulate the behavior of maintenance operators on the maintenance lines during the visual inspection phase.

	Acquisition systems		
	Smartphone (12 MP)	Drone (4 MP)	Reflex (24 MP)
D = 65 cm			
D = 75 cm			
D = 85 cm			
D = 100 cm			

Table 1 - Performance of the acquisition systems during the trade-off phase. Green: Good number of classifications, Yellow: Mediocre number of classifications, Red: Poor number of classifications.

Despite the smartphone's better performance in the number of correct defect classifications, the reflex camera was still chosen to build the training database to best simulate environmental conditions in the hangars.

After the trade-off phase for choosing the acquisition system, two test articles present in the LIght saFe quiEt laboratory (LIFE Lab) of the Department of Industrial Engineering at the University of Naples were selected. In particular, the following test articles were used:

- A full-scale fuselage section of a commercial aviation aircraft in primer paint
- A full-scale fuselage section of a general aviation aircraft in white paint

Subsequently, a database (Table 2) was created containing approximately 1000 images taken under nominal light, low light, and blurred (to include the effects of drone propeller vibrations in the database, which make the images blurry) conditions.

Database	**Condition**	**Missing rivets (n. of images)**	**Corroded rivet (n. of images)**	**General damage (n. of images)**
Images from a full-scale fuselage section of a general aviation aircraft in white paint	Light	60	19	33
	Low light	51	20	23
	Blurred	80	79	47
Images from a full-scale fuselage section of a commercial aviation aircraft in primer paint	Light	74	34	12
	Low light	143	15	13
	Blurred	208	22	31
TOT. images for each defect		616	189	159
TOT. database's images		964		

Table 2 - Database for the training phase of the computer vision algorithm.

For each image, to correctly label every defect, bounding boxes were drawn around each encountered damage to train the neural network during the next training phase. In this way, the Regions Of Interest

(ROI) annotated during the database processing phase are approximately 6000.

2.1.2 Training phase for the damage detection algorithm

The training phase is crucial for accurately classifying defects in the testing images. It involves the optimization of two functions [20] [21]:

- Accuracy function: metric used to evaluate the performance of a model during the training process. It measures the proportion of correct predictions made by the model compared to the total number of predictions. The accuracy function is usually calculated as the number of true positives (correctly predicted positive instances) and true negatives (correctly predicted negative instances) divided by the total number of instances in the dataset. The value of the accuracy function ranges between 0 and 1, where 1 indicates a perfect model that correctly classifies all instances, and 0 indicates a model that does not make any correct predictions.
- Loss function: the main purpose is to measure the discrepancy between the model's predicted output and the actual ground-truth output (ROI). By quantifying the difference, the loss function provides a signal that guides the optimization process, allowing the model to iteratively adjust its parameters and improve its performance. The loss function represents the objective that the deep learning model aims to minimize during the training phase.

To achieve the optimal results, several parameters (Table 3) need to be appropriately set during this phase, including:

N. of classes	3 (eroded_rivet, no_rivet, general_damage)
Training algorithm	ADAM (Adaptive Moment Estimation)
Batch size	16
Initial Learning Rate	10^{-6}
Epochs	50

Table 3 - Training algorithm parameters.

1. N. of classes: number of damages' typology to identify, classify, and localize.
2. Training algorithm: in this case, the Adaptive moment estimation (ADAM) algorithm [22] is chosen. In this training algorithm, there is an adaptive learning rate, which is individually adjusted for each weight in the model. This is achieved by computing and maintaining two-moment estimates: the first moment (mean) and the second moment (uncentered variance) of the gradients. These moment estimates are then used to scale the learning rate and prevent it from being too large or too small.
3. Batch size: To efficiently train the convolutional neural network, the database is divided into smaller "mini-databases" called batches [23]. Each batch consists of a limited number of input images, and the loss factor calculated for each minibatch is an approximation of the loss factor for the entire database.
4. Initial learning rate: This is the starting value of the learning rate parameter. It should not be too aggressive, as it could lead to errors in training the convolutional neural network.
5. Epochs: This is the number of times the convolutional neural network analyzes the entire database during the training phase.

The graphs of the accuracy function (specifically, the precision function) (Figure 1) and the loss function (Figure 2) related to the neural network training phase are reported:

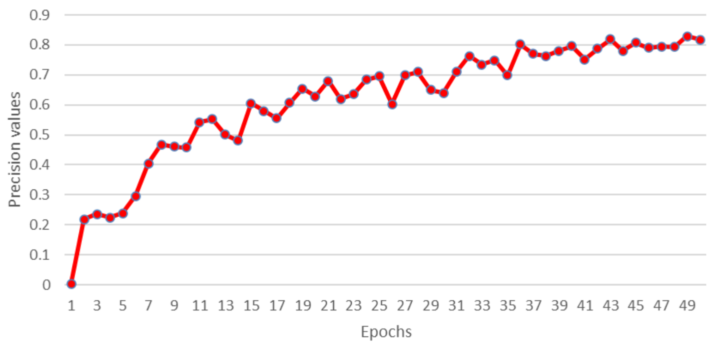

Figure 1 - Accuracy function from the training phase.

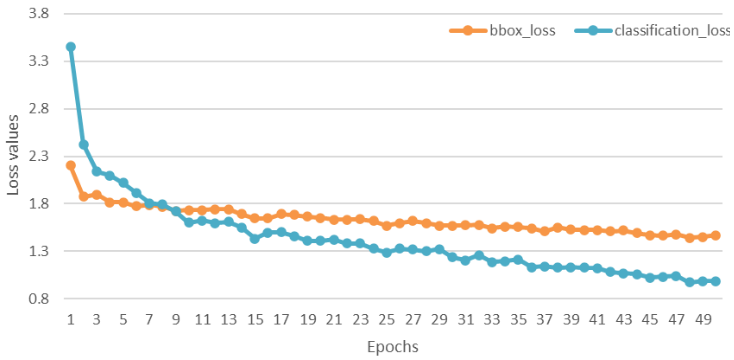

Figure 2 - Loss function from the training phase.

2.1.3 *Results of damage detection algorithm*

After the training phase of YOLO architecture, the results (Figure 3 to Figure 6) obtained during the testing phase were evaluated. In particular, the YOLO architecture is capable of classifying missing rivets with an accuracy of 75%, eroded rivets with an accuracy of 63%, and general damages with an accuracy of 87%.

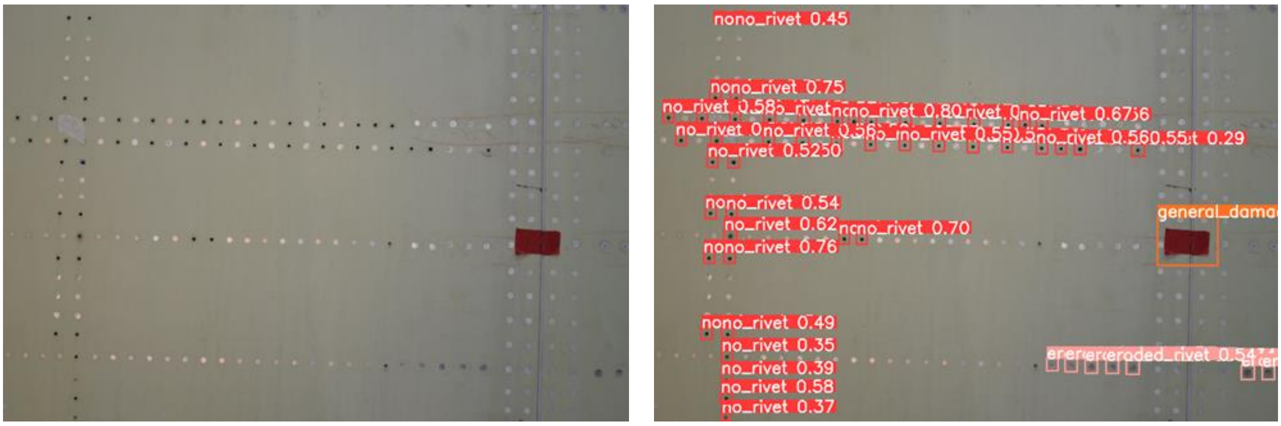

Figure 3 - Test image 1 before (left) and post (right) processing

Figure 4 - Test image 2 before (left) and post (right) processing.

Figure 5 - Test image 3 before (left) and post (right) processing

Figure 6 - Test image 4 before (left) and post (right) processing

2.2 Errors in decal detection algorithm set-up procedure

The methodology for setting up the algorithm to identify, locate, and classify errors in decals on aircraft panels is reported. In particular, the process of building the database, the characteristics of the test facility equipped to collect photos of defects, the setting of the training parameters of the YOLO v8 architecture, and the results coming from the testing phase were illustrated.

2.2.1 Test facility and database description for errors in decal

A database (Table 4) containing approximately 100 images was created, taken under nominal light, low light, and blurred (to include the effects of drone propeller vibrations in the database, which make the images blurry) conditions.

Database	Condition	General damage (n. of images)
Images from a full-scale aircraft panel of a general aviation aircraft in white paint	Light	34
	Low light	33
	Blurred	30
TOT. database's images		97

Table 4 - Database for the training phase of the computer vision algorithm.

Even in this case study for each image, to correctly label every defect, bounding boxes were drawn around each encountered damage to train the neural network during the next training phase. In this way, the Regions Of Interest (ROI) annotated during the database processing phase are approximately 600.

2.2.2 Training phase for errors in decal algorithm

To have the best results in the testing phase, parameters (Table 5) were set for the training phase of the YOLO architecture for detecting errors in decals.

N. of classes	1 (decal_error)
Training algorithm	ADAM (Adaptive Moment Estimation)
Batch size	16
Initial Learning Rate	10^{-6}
Epochs	100

Table 5 - Training algorithm parameters.

In the following figures, the graphs of the accuracy function (specifically, the precision function) (Figure 7) and the loss function (Figure 8) related to the neural network training phase are reported:

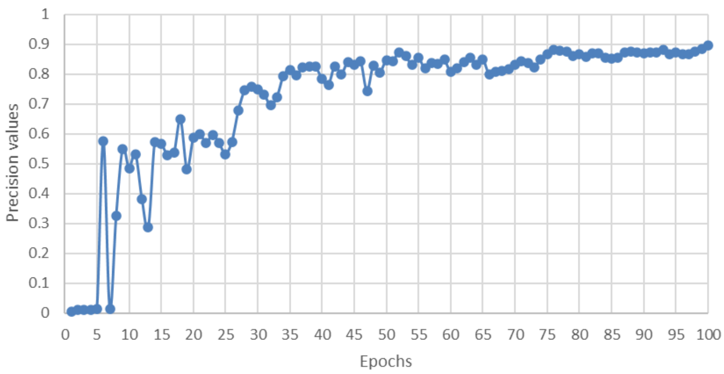

Figure 7 - Accuracy function from the training phase.

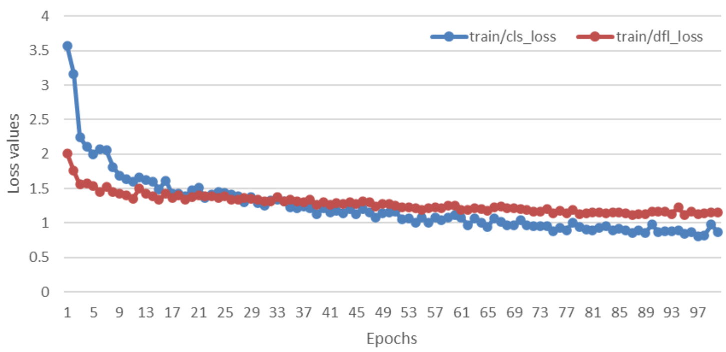

Figure 8 - Loss function from the training phase.

2.2.3 Results for errors in decal algorithm

The results obtained during the testing phase are schematically reported here. In detail, the procedure is demonstrated to be valid on four image sets, as shown in Figure 9 to Figure 12. The YOLO architecture is capable of classifying errors in decals with an accuracy of 82%.

Figure 9 - Results from errors in decals' first test image set.

Figure 10 - Results from errors in decals' second test image set.

Figure 11 - Results from errors in decals' third test image set.

Figure 12 - Results from errors in decals' fourth test image set.

3. Conclusions

In this paper, a new method for aircraft maintenance has been outlined. This procedure allows for the use of a small quadcopter drone to reach remote areas on the fuselage of aircraft and take photos which are then post-processed by software that uses machine learning and computer vision algorithms. The process began with a trade-off phase to select a high-resolution acquisition system, to better simulate the behavior of the drone when taking photos in an industrial environment (inside and outside the hangar). This information was then used to build an image database for damage detection on full-scale aircraft skin fuselage panels and errors in decals on full-scale aircraft panels. The database was then used to train the YOLO v8 network to identify, locate, and classify damages on aircraft and errors in decals on aircraft panels. As reported in the previous paragraphs, the YOLO v8 architecture is capable of classifying missing rivets with an accuracy of 75%, eroded rivets with an accuracy of 63%, general damages with an accuracy of 87%, and errors in decals with an accuracy of 82%. The network is to be optimized to identify and classify a greater number of defects with higher accuracy. The database for the training phase will be expanded with images from an aircraft maintenance line to experiment with this new approach in an industrial environment, using the semantic segmentation module of the YOLO v8 architecture. This approach allows operators to perform near real-time maintenance under safer conditions, both during scheduled maintenance and pre-flight checks.

4. Contact Author's Email Address

salvatore.merola@unina.it (corresponding author);
michele.guida@unina.it;
francesco.marulo@unina.it

5. Copyright Statement

References

[1] Shuyuan et al. Computer vision techniques in construction: a critical review, Advanced Machine Learning Technologies, and Applications: Proceedings of AMLTA 2020, pp. 3383–3397, 2021.

[2] Amitha Mathew, P Amudha, and S Sivakumari. Deep learning techniques: an overview., Archives of Computational Methods in Engineering 28, pp. 599-608, 2021.

[3] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects, IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 12, pp. 6999-7019, Dec. 2022.

[4] Ajeet Ram Pathak, Manjusha Pandey, and Siddharth Rautaray. Application of deep learning for object detection. Procedia computer science 132, pp. 1706–1717, 2018.

[5] Wilson Ricardo Leal da Silva and Diogo Schwerz de Lucena. Concrete crack detection based on deep learning image classification. Proceedings. Vol. 2.8 MDPI., p. 489, 2018.

[6] Taormina Vincenzo, et al. Performance of fine-tuning convolutional neural networks for HEP-2 image classification. Applied Sciences 10.19, p. 6940, 2020.

[7] Maintenance Steering Group et al. Handbook: Maintenance Evaluation and Program Development (MSG-1). Air Transport Association, Washington D. C. 10, 1968.

[8] D. Nakata. An introduction to MSG-3 (Maintenance Steering Group). Society of Automotive Engineers, Aerospace Congress and Exposition, Long Beach, CA. Vol. 15. 18. 1984.

[9] Gregory Sweers and John Fielding. Recent Developments of MSG-3 (Maintenance Steering Group) Guidance for the Continued Airworthiness of Lightning/ HIRF (L/HIRF) Protection and Benefits to New Airplane Design. The 26th Congress of ICAS and 8th AIAA ATIO, p. 8964, 2008

[10] Alan Hobbs. An overview of human factors in aviation maintenance. In: ATSB Safety Report, Aviation Research and Analysis Report AR 55, p. 2008, 2008.

[11] Khairy AH Kobbacy et al. “Reliability centred maintenance”. In: Complex system maintenance handbook, pp. 79–108, 2008

[12] Yang Tang et al. A framework for identification of maintenance significant items in reliability centered maintenance. Energy 118, pp. 1295–1303, 2017.

[13] B. Chitradevi and P. Srimathi. An overview on image processing techniques. International Journal of Innovative Research in Computer and Communication Engineering 2.11, pp. 6466–6472, 2014.

[14] Airbus. ref 51-11-00 Description - DAMAGE CLASSIFICATION. Tech. rep. Airbus Group, 2003.

[15] Airbus. ref 51-11-13 - Description - DAMAGE/DEFECT REPORTING. Tech. rep. Airbus Group, 2003.

[16] Airbus. ref 51-11-11 Description - Allowable Damage. Tech. rep. Airbus Group, 2003.

[17] Redmon, Joseph and Divvala, Santosh and Girshick, Ross and Farhadi, Ali. You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016

[18] Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLO (Version 8.0.0) [Computer software]. https://github.com/ultralytics/ultralytics.

[19]Terven J, Córdova-Esparza D-M, Romero-González J-A. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Machine Learning and Knowledge Extraction. 2023

[20] Katarzyna Janocha and Wojciech Marian Czarnecki. On loss functions for deep neural networks in classification”. arXiv preprint arXiv:1702.05659, 2017.

[21] Terven, J., Cordova-Esparza, D. M., Ramirez-Pedraza, A., Chavez-Urbiola. Loss functions and metrics in deep learning. A review. arXiv preprint arXiv:2307.02694. 2023.

[22] Zhang, Z. Improved adam optimizer for deep neural networks. IEEE/ACM 26th international symposium on quality of service (IWQoS) (pp. 1-2). 2018.

[23] Radiuk, P. M. Impact of training set batch size on the performance of convolutional neural networks for diverse datasets. Information Technology and Management Science, 20(1), 20-24. 2017.