Towards automated aircraft maintenance inspection. A use case of detecting aircraft dents using mask r-cnn

Bouarfa, Soufiane; Doğru, Anıl; Arizar, Ridwan; Aydoğan, Reyhan; Serafico, Joselito

DOI
10.2514/6.2020-0389

Publication date
2020

Document Version
Final published version

Published in
AIAA Scitech 2020 Forum

AIAA SciTech Forum
6-10 January 2020, Orlando, FL
AIAA Scitech 2020 Forum

10.2514/6.2020-0389

Towards Automated Aircraft Maintenance Inspection. A use case of detecting aircraft dents using Mask R-CNN

Soufiane Bouarfa[1]
Abu Dhabi Polytechnic, Institute of Applied Technology, Al Ain, Emirate of Abu Dhabi, 66844, UAE

Anıl Doğru[2]
Department of Computer Science, Özyegin University, Istanbul, 34794, Turkey

Ridwan Arizar[3]
Singulair Solutions B.V., Data Analytics As a Service, Rotterdam, 3011BN, The Netherlands

Reyhan Aydoğan[4]
Department of Computer Science, Özyegin University, Istanbul, 34794, Turkey
Interactive Intelligence Group, Delft University of Technology, The Netherlands.

Joselito Serafico[5]
Abu Dhabi Polytechnic, Institute of Applied Technology, Al Ain, Emirate of Abu Dhabi, 66844, UAE

Deep learning can be used to automate aircraft maintenance visual inspection. This can help increase the accuracy of damage detection, reduce aircraft downtime, and help prevent inspection accidents. The objective of this paper is to demonstrate the potential of this method in supporting aircraft engineers to automatically detect aircraft dents. The novelty of the work lies in applying a recently developed neural network architecture know by Mask R-CNN, which enables the detection of objects in an image while simultaneously generating a segmentation mask for each instance. Despite the small dataset size used for training, the results are promising and demonstrate the potential of deep learning to automate aircraft maintenance inspection. The model can be trained to identify additional types of damage such as lightning strike entry and exit points, paint damage, cracks and holes, missing markings, and can therefore be a useful decision-support system for aircraft engineers.

Nomenclature

BB	=	Bounding Box
CNN	=	Convolutional Neural Network
FC	=	Fully Connected
FCN	=	Fully Convolutional Network
FN	=	False Negatives
FP	=	False Positives
L	=	Loss function
ROI	=	Regions of Interest
R-CNN	=	Region Convolutional Neural Network
RPN	=	Region Proposal Network
TP	=	True Positives

[1] Assistant Professor of Aviation, Abu Dhabi Polytechnic, Institute of Applied Technology.
[2] MSc student in Computer Science, Özyeğin University.
[3] Partner Consultant, Singulair Solutions B.V.
[4] Assistant Professor in Computer Science, Özyeğin University; and Researcher at the Interactive Intelligence Group, Delft University of Technology.
[5] Aircraft Engineer, Abu Dhabi Polytechnic, Institute of Applied Technology.

I. Introduction

DESPITE the rapid advances in technology, aircraft maintenance visual inspection has not evolved during the last 40 years. The inspection process is not only expensive and time consuming [26], but also subjective and in some cases dangerous. In [15], it was reported that automated inspection typically surpasses the standard of visual inspection by a human. Automating visual inspection can reduce Aircraft on Ground time by up to 90%, and introduce measurable efficiencies for both scheduled and unscheduled maintenance [5]. Especially the inspection of high waterline areas requires a long preparation time such as arranging work platforms, anti-fall straps, man-lift, and all the other associated ground support equipment therefore requiring many man hours. Through eliminating the manual process, inspection time can significantly be reduced leading to significant cost savings, and maintenance inspection accidents like falls [14] often due to time pressure or fatigue could be prevented. Consequently, this could also reduce flight delays and cancellations. Another benefit of automating visual inspection is preventing situations where inspectors fail to notice critical damage. For instance, the Australian Transport Safety Bureau (ATSB) recently reported a serious incident in which a significant damage to the horizontal stabilizer went undetected during a subsequent inspection, and was only identified 13 flights later [13]. Finally, automated aircraft inspection can also enable objective assessments as it is not uncommon to have different assessments by different inspectors for the same damage. In some cases, even the same inspector might disagree with himself at a different inspection time. To improve the quality of inspections, computer vision using deep learning methods can solve this problem for instance through using datasets labeled by a team of experts.

Computer vision is changing the field of visual assessment in nearly every domain. This is not surprising given the rapid advances in the field. For instance, the error in object detection by a machine decreased from 26% in 2011 to only 3% in 2016 which is less than human error reported to be 5% [12]. The main driver behind these improvements is deep learning which had a profound impact on robotic perception following the design of AlexNet in 2012. In **healthcare,** computer vision technology has become so good in medical imaging diagnosis that the FDA has recently approved many use cases [17, 18]. A recent study [16] compared the performance of disease detection from images by a machine versus specialists using a sample of studies between 2012 and 2019. The research found that computer vision has become more accurate in the last years in diagnosing diseases. The diagnostic performances by both humans and machine were found to be equivalent in 14 studies. In the **automotive** industry, companies such as Tesla and Waymo are working towards fully driverless cars enabled by computer vision technology that can detect various objects around the car. In **production and manufacturing** environment, computer vision is used for the external assessment of product quality and equipments such as tanks, pressure vessels, and pipes. In **agriculture**, various computer vision algorithms are integrated with drones that can scan large fields in a matter of minutes. Images are collected and processed to help farmers make informed decisions about their crops. The captured images include soil and crop conditions to monitor for any stress or disease [22]. Other domains include **retail** such as the new stores of amazon where customers can purchase products without being checked out by a cashier. The concept makes use of various technologies such as deep learning, and sensor fusion to automate much of the purchase, checkout, and payment [23].

Applications of computer vision in aircraft maintenance inspection remain very limited despite the impact this field is already making in many domains. Based on the literature and technology review performed by the authors, it was found that only a handful of organizations are working towards automating aircraft visual inspection. For instance Donecle [8], has developed a drone-based system that can inspect aircraft for different types of damage 10 times faster than current inspection methods. The inspection is carried out in three main steps mainly data collection, software-based diagnosis, and data storage for predictive maintenance. However, it was not reported which technology or algorithms are used in the diagnosis step. A similar automated inspection system has been developed by Ubisense and Mrodrone [5,6]. The system has been successfully tested by Easyjet [6], and is planned to be rolled out across Easyjet's european bases. However it is not clear whether the solution makes use of deep learning. Air-Cobot [24] is another example project to automate aircraft inspection. The research team makes use of image processing techniques to detect different types of objects on the aircraft such as damage caused by impact or cracks. However, to the best of the authors knowledge, no study before has applied Convolutional Neural Networks to automate the detection of aircraft dents.

Convolutional Neural Networks (CNNs) are considered to be the driver behind computer vision applications, and are evolving fast with advanced novel architectures. Several object detection techniques exist such as R-CNN, Fast R-CNN, Faster R-CNN, and Mask R-CNN [28]. The latter technique, the most advanced of the four, is chosen in this research because it allows identifying objects in the image at the pixel level (instance segmentation). Mask R-CNN is different from the classical models like Faster R-CNN because in addition to identifying the class and its bounding box location, it also colors the pixels in the bounding box that correspond to that class [30]. Such feature is very useful in detecting the exact pixel location of aircraft dents which do not have a clearly defined form. Mask R-CNN is considered to be a combination of Faster R-CNN which does the object detection and FCN which does the pixel wise boundary. The goal of object detection is to classify individual objects and localize each object instance using a bounding box. The goal of segmenting each instance is to classify each pixel into a fixed set of categories without differentiating object instances.

This paper is organized as follows. Section 1 presents the introduction. Section 2 describes the use case selected to demonstrate the potential of Mask R-CNN in automating visual inspection. Section 3 explains the approach and techniques used together with the implementation steps. Section 4 presents the main results. The conclusion and future steps are provided in section 5.

II. Use Case

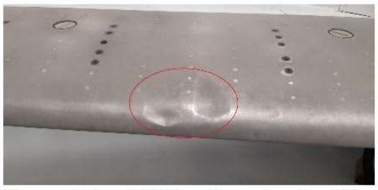
Computer vision can help aircraft engineers to automatically detect various types of external damage on the aircraft skin. The detected damage might include dents, scratches, cracks, abrasion, crease, corrosion, gouges, holes, nick, puncture, just to name a few. Examples of use cases in which computer vision can be leveraged to automate visual inspection are shown in Figure 1 and include:

- **Detection of aircraft dents:** This is usually caused by an impact from a smoothly contoured object. The affected area is pushed in from its normal shape. There is no change in the cross-sectional area of the material and no sharp edges. Widely separated dents are not easy to locate, and it could take inspectors a day or two to assess the aircraft damage depending on the airplane size, inspection area, and facility support. Automatic inspection instead can complete the damage assessment in less than an hour.

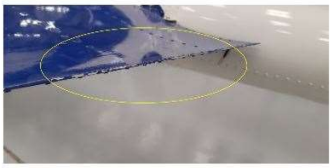
- **Detection of lightning strike damage:** Lightning damage to metal parts usually shows as pits, burn marks or small circular holes, which can be grouped in one area or spread over a large area. Composite damage is shown as discoloured paint, burned, punctured, or de-laminated skin plies [19]. Lightning enters and exits through the extremities of the airplane. The vertical fin is a likely point of entry. When this happens, it will not be easy for humans to inspect the area 63 feet high from the ground. Automatic inspection can assess the damage safely and much faster.

- **Inspection of aircraft markings and placards:** Invalid or missing markings on aircraft can lead to fines and even to aircraft grounding. The manual task of checking markings is a long process as the number of placards can be up to 300 on a narrowbody aircraft. Automatic inspection can quickly scan the aircraft and evaluate the quality of markings without the need of engineers to walk 30-50 feet high on work platforms.

- **Inspection of Paint Quality (paint flaking, wear, and degradation):** Typical paint and finishing problems are poor adhesion, blushing, pinholes, sags and/or runs "orange peel," fisheyes, sanding scratches, wrinkling, and spray dust [20]. Manual inspection of these paint defects is time consuming and sometimes hazardous. Also it's not uncommon to miss few defects by inspectors. This task consumes hundreds of man-hours when done by unaided human eyes and is prone to error. Defects can easily be missed out if inspectors are not close enough to the aircraft surface.

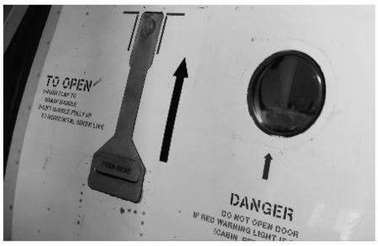


Dented Falcon 20 Wing Droop Leading Edge



Poor adhesion of paint finish on Bell Helicopter Tail Plane

Worn Paint Finish of a Bell Helicopter Cowling



Exterior marking on aircraft door [21]

Lightning damage [19]

Figure 1: Examples of various types of damages that can be automatically detected using computer vision

In this paper we will demonstrate the concept by focusing on the automatic detection of aircraft dents. According to Boeing 727 SRM ATA 51-10-1 [10], a dent is defined as a damaged area that is pushed in from its normal contour with no change in the cross-sectional area of the material. The edges of the damaged area are smooth. This damage is usually caused by an impact with a smoothly contoured object. The length of the dent is the longest distance from one end to the other end. The width of the dent is the second longest distance across the dent, measured at 90 degrees to the direction of the length. It should be noted that a dent-like form of damage to a panel area with a thick skin can be the possible result of the peening action of a smoothly contoured object. If the inner surface of skin shows no contour change, then the damage can be thought of as a local cross-sectional area change (Appendix A). In most of the cases, aircraft dents are detected through visual inspection and assessed by inspectors. Figure 2 shows a typical process following the detection of a dent on the aircraft wing.

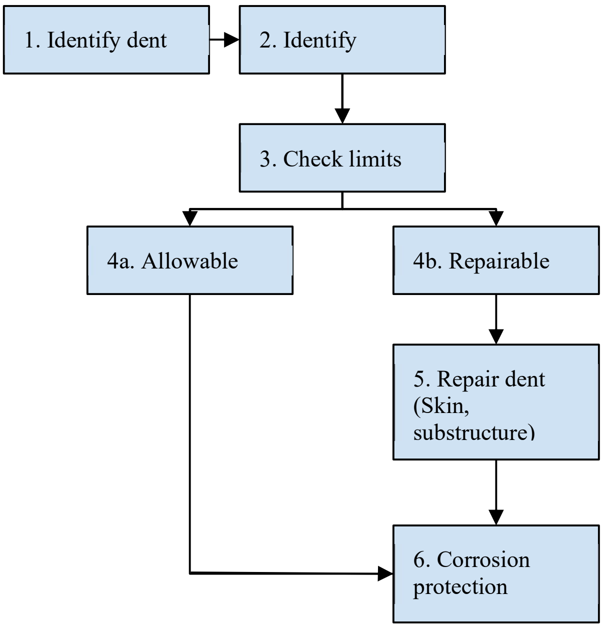


Figure 2: Decision-making process by an experienced maintenance inspector noticing an aircraft dent on the leading edge of the wing

1. **Identify Location:** Identifying the location of dent will be necessary to know which parts of the aircraft have been affected. In this case, the dented part is classified as Secondary Structure and is called Droop Leading Edge located on the LH Wing Assembly, SRM ATA 57-00-0 in [11]. See Appendix A Fig A03-01.
2. **Identify damaged parts:** Here the different parts of the Inboard Droop Leading Edge structure (Skin and Structural Members) are identified, SRM ATA 57-50-33 in [11]. See Appendix A Fig A03-01, Fig A03-02, and Fig A03-03.
3. **Check Limit:**. The parameters for negligible deformation are described in the SRM ATA 51-10-11 in [11]. This includes the importance of the damaged member, extent of the damage, appearance as evaluated by operators. SRM ATA 51-10-11 in [11] also provides the general information concerning repair work.
4. **a. Allowable Damage**: On vital members (spars, integral panels, main attachment fitting), no deformation of this type is to be considered negligible, SRM ATA 51-10-11 in [11]. Limits of skin deformation and underlying structure damage of Droop Leading Edge are described in SRM ATA 51-10-11 in [11] and the related sub assembly chapters.

   b. Repairable Damage: Identify the damaged parts. Droop leading edge Skin and Structural members are identified in SRM ATA 57-50-33 in [11]. See Appendix A Fig A03-01, Fig A03-02, and Fig A03-03.
5. **Repair:** Perform typical repair in accordance with SRM 51-10-21 [11] for the skin and SRM 51-10-22 [11] for the structural members. Classify and select the appropriate type of patch, determine the appropriate cut out, select appropriate splice plates, doublers, doors and fasteners. The repair will be adapted depending on whether it is over a structural member, on the edge of a panel, on a blind skin panel. If installing a repair is not economical, this would require complete replacement of the Skin panel. If no typical repairs are suitable, consult the manufacturer for Specific Repairs.
6. **Corrosion Protection and Refinish Paint:** Comply with the applicable anti-corrosion scheme in accordance with SRM ATA 51-20-4 in [11] .

While the various aircraft manuals contain the instructions to prepare and fix the aircraft, the use of automated inspection can be used to quickly pre-assess the material condition of the aircraft while the examiner is outlining a work plan that will utilize the data streamlined by the cameras (e.g. a drone or smart hangar equipped with cameras). Thus, timely briefing the pilot, operator or management of the operational impact of the required repair work. A set of maintenance requirements will be established earlier than normal as the engineer combines collected data with his experience and other conventional resources. Data collected can be utilized for inspection of similar defects. Also, data from this scheme can be used for trend analysis purposes.

To have an idea of the extent and depth of preparations, the following is a list of some items typically considered during the first two hours of planning for the job at hand:

1. Should the aircraft be placed in the hangar with the required maintenance platforms?
2. Would fixing the airplane have adverse impact on current shop workload?
3. Would jacks or trestles be necessary?
4. Would paint stripping be required?
5. What NDT would be required to examine and confirm the extent of damage?
6. What engineering supports and how many would be required (mechanical, electrical, other specialists)?
7. What repair materials would be required?
8. How long will the repair take to complete?
9. What parts removal would be required to gain access to the repair area?
10. What aircraft systems would be disturbed? Example: Anti-Icing System, Flight Control System
11. What tools/special tools or support equipment would be needed?
12. What documents (Technical Manuals, Logbooks, Technical Directives, etc…) would be required?
13. Cost estimate of man-hours, materials, and facility required.

Actual inspection of a large airplane for paint condition, lightning damage, dents or impact damage at heights (such as the empennage surfaces or top surfaces of the fuselage) can take 2-3 days and hundreds of man-hours of preparations. Automated inspection could cut the preparation time to less than a day and with the aircraft engineers spending less than 30 man-hours. Also, data collected are automatically saved for much deeper analysis or communicated to remote engineering support where preparations to receive the airplane can be done saving valuable resources.

III. Methodology

This paper uses Mask Region Convolutional Neural Networks (Mask R-CNN) to automatically detect aircraft dents. Mask R-CNN is a deep learning algorithm for computer vision that can identify multiple objects in one image. It goes beyond a plain vanilla CNN in that it allows the identification of the exact location of objects within their bounding boxes, so not only limited to object identification in the image. This functionality is relevant for detecting aircraft dents which don't have a clearly defined shape. However, Mask R-CNN comes at a computational cost. For example, YOLO [25], a popular object detection algorithm is much faster if all what needed is bounding boxes. Another drawback of Mask R-CNN is labelling the masks. Annotating data for masks is a cumbersome and tedious process as the data labeler needs to draw a polygon for each of the objects in an image.

A. Object Detection

As with every object detection task, there exist three subtasks [4] (see also figure 3):

1. **Extracting Regions of Interest (ROI):** The image is passed to a ConvNet which returns the Region of Interests (RoIs) based on methods like selective search (R-CNN) or RPN (Region Proposal Network for faster R-CNN). Then, a pooling layer is extracted from the ROI to ensure all regions have the same size.
2. **Classification Task:** Regions are passed on to a fully connected network which classifies them into different image classes. In our case study, the classes are dent 'Damage' or background 'aircraft skin without damage.'
3. **Regression Task:** A bounding box (BB) regression is used to predict the bounding boxes for each identified region for tightening the bounding boxes.

Since aircraft dents don't have a clearly defined shape, arriving at square/rectangular shaped BBs is not sufficient. it's important to identify the exact pixels in the bounding box that corresponds to the class damage. Exact pixel location of the dent will help to identify the location and quantify the damage. An additional step is needed: semantic segmentation (pixel-wise shading of the class of interest) into the entire pipeline for which we will use Masked Region based CNN (Mask R-CNN) architecture.

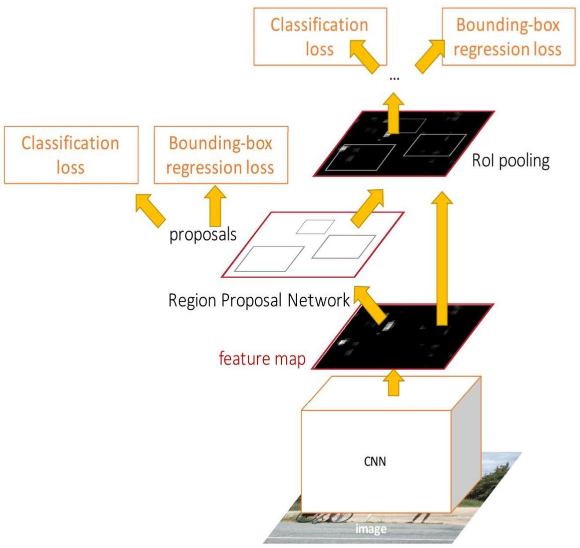


Figure 3: Faster R-CNN Architecture based on [32]

B. Mask R-CNN

Mask R-CNN is an instance segmentation model which enables the identification of pixel-wise delineation of the object class of interest. In order to get instance segmentation for a particular image, two main tasks are required (See figure 4):

1. **BB based object detection (Localization task):** uses similar architecture as faster R-CNN. The only difference in Mask R-CNN is the ROI step. Instead of using ROI pooling, it uses ROI align to allow the pixel to pixel preserve of ROIs and prevent information loss.
2. **Semantic segmentation:** which allows segmenting individual objects at pixel within a scene, irrespective of the shapes. Semantic segmentation uses a Fully Convolutional Network (FCN) which creates binary masks around the BB objects through creating pixel-wise classification of each region. Hence, Mask R-CNN minimizes the total loss.

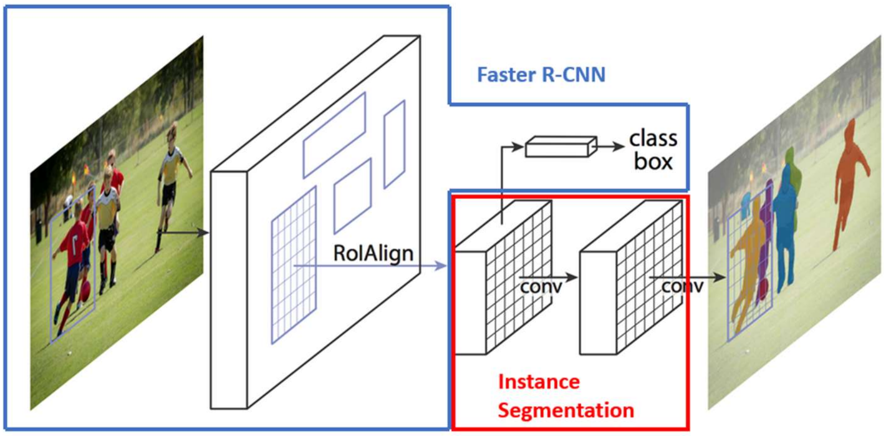


Figure 4: Mask R-CNN framework for instance segmentation [2]

C. Implementation

This section discusses the data preparation and the implementation of the concept on real-life aircraft images using Mask R-CNN. The authors have adopted the code taken from [4] such that it can be used to identify dents on aircraft structures. In order to reduce the computational time to train the Mask R-CNN, we have taken as initial weights the ones from [34]. This concept is also known as transfer learning [31] with a warm restart as shown in figure 5. By pre-training the neural network on the COCO dataset, we then re-use it on our target data set as the lower layers are already trained on recognizing shapes and sizes. In this way we refine the upper layers for our target data set (aircraft structures with dents).

Figure 5: The concept of transfer learning with a warm restart

1. Data Collection

Although this research uses transfer learning from a pre-trained(weight) CNN architecture, a key first step is to customize the network for our case in order to reduce the application specific loss. This loss results from a mismatch of damage location in pixel level between the ground truth and predicted. In our experiment, we have run and trained the network on 90% of images of aircraft dents, collected from the internet and our own hangar, and used the remaining photos for testing purposes. This experiment was done in 10 different combinations of testing and training photos (10 folds). The pictures used for training the model were different in terms of:

- Background color of the dent (e.g. white, blue, gray, etc.)
- Size of dents (small, medium, large)
- Location of dents (e.g. leading edge of wing, radome, engine, horizontal stabilizers)
- Causes of dents (e.g. hail, bird strike, etc.)
- Source (online photos, own photos made by different cameras)
- Lighting conditions
- Resolution (low resolution, high resolution)
- Distance and angle from which the picture is taken

2. Data Annotation

Since we are training the computer how to detect aircraft dents, this concept falls under supervised learning. Therefore we need to label the data. In computer vision object detection, this labeling is referred to annotation. In this application, it means precisely identifying the region of damage in a photo and marking the boundary around the aircraft dent that is not allowed. To perform this annotation, we make use of VGG Image Annotator [3] shown in figure 6 which is an open source tool developed by the Visual Geometry Group of Oxford University. Using this tool, we uploaded all the images and drew polygon masks along the dents for each image as follows.

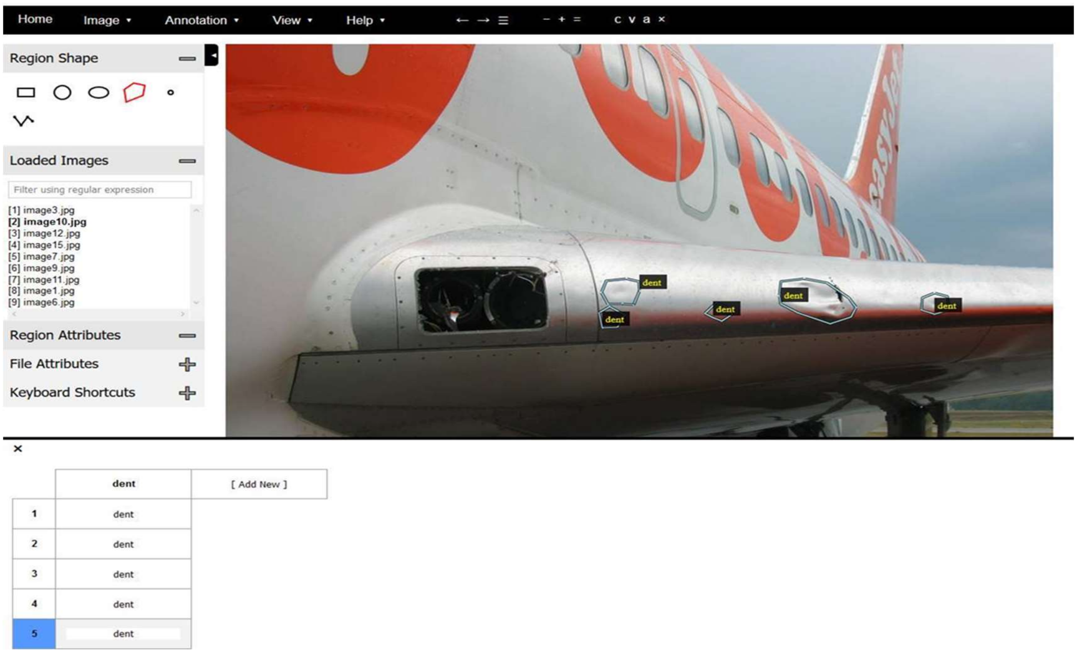


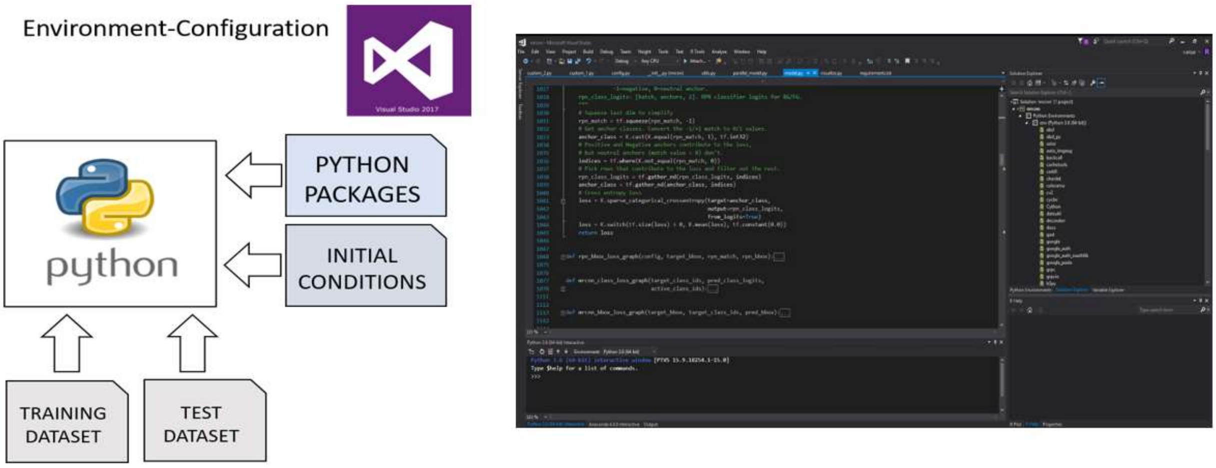
Figure 6: Examples of data annotation using the VGG Image Annotator (VIA) [3]

3. Environment Set-Up

The most crucial element before training the model is setting up a proper environment, where the core computations are performed. Here we choose to use Python since it offers a wide range and fully tested open-source packages /

libraries that cuts down the source code development time. As a so-called Integrated Development Environment (IDE) we have used Microsoft Visual Studio 2017 (MS VS2017), since this is also freely available and integrates Python sources smoothly. Figure 7 shows the elements used in setting up the computational environment which consists of:

- Python
- Python packages: {numpy, Scipy, Pillow, Cython, Matplotlib, Scikit-image, tensorflow>=1.3.0, keras>=2.0.8, Opencv-python, H5py, Imgaug, IPython}
- Initial conditions: these are the pre-trained weights of the Mask R-CNN network and the tuned hyper parameters. The pre-trained weights are taken for a deep neural network that is trained using the so-called COCO data. The link to the pre-trained weight can be find at https://github.com/matterport/Mask_RCNN/releases/download/v2.0/mask_rcnn_coco.h5
- The training and the test data consisting of images of dents detected on airplanes fuselage / wings.
- The IDE (MS VS2017)



(a) (b)

Figure 7.a: The core elements in the environment-setup, **7.b:** screenshot of IDE MS VS2017 with in right upper window some of the used (sub)-libraries in the project.

4. Loading Dataset

For training the Mask R-CNN network we have used approximately 50 pictures of airplanes with dents, which seems at first hand a small number. However, we have used for the training initial weights that have been used to train dents on cars so it is to be expected that only a relatively small amount of input data is needed to train the Mask R-CNN to classify dents on airplane structures. Some examples of training data is shown below in figure 8. It should be noted that this training data is accompanied by JSON files containing coordinates of the bounding boxes and the masks formed by polygons. Both training sets of images and JSON files are fed into the Mask R-CNN code.

Figure 8: Sample of training data corresponding to dents of aircraft in Abu Dhabi Polytechnic Hangar

5. Network Training

Details on the structure used in the Mask R-CNN is depicted below in Figure 9. The process flow of the Mask R-CNN can be divided in two stages. In the first stage the location of the possible object of interest is detected on the input image (region proposal), using a fully connected CNN. In the second stage, it predicts the class of the object, refines the bounding box and generates a mask in pixel level of the object based on the first stage region proposal.

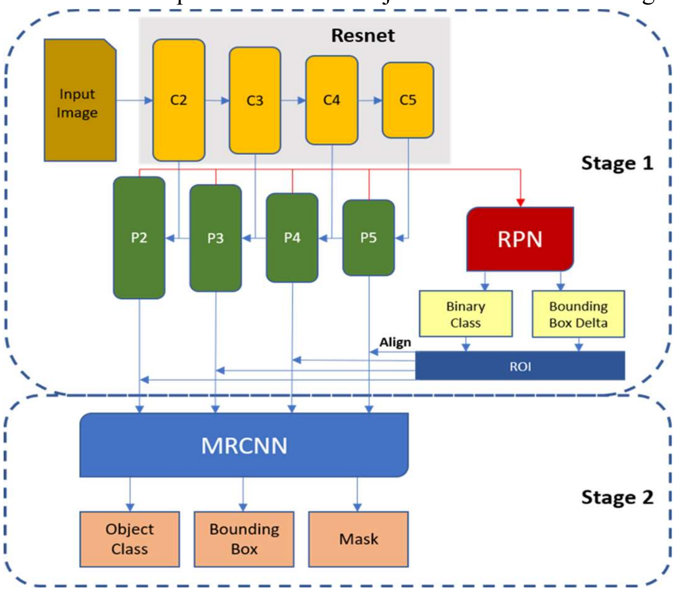


Figure 9: Detailed component view of the Mask R-CNN and the associated information flow

Stage 1 components:

- Input image
- C2 up to C5: Convolutional Layers (4 layers with sizes [64, 64, 256], [128, 128, 512], [256, 256, 1024], [512, 512, 2048]) which forms the Resnet layer. This is a standard convolutional neural network ResNet101 that serves as a feature extractor. The early layers detect low level features (edges and corners), and later layers successively detect higher level features (airplane, sky, person).
- P2 up to P5: Pooling Layer (4 layers with sizes [64, 64, 256], [128, 128, 512], [256, 256, 1024])
- RPN: Regional Purpose Network is a lightweight neural network that scans the image in a sliding-window fashion and finds areas that contain objects.

- Binary Classification
- BBox Delta: These are basically the N number boxes around the dents on the input image and defined with a center (x, y) coordinate and associated width and height.
- ROI Align

Stage 2 components:

- Class: Classifies the type of object
- Bbox: Creates a bounding box around the region of interest
- Mask: Within the bounding box a mask is put on the region of interest

Loss functions optimization:

An indicator for evaluating how well a specific deep learning algorithm models the given data is the so-called loss function L. This is basically an objective function that needs to be minimized. When the predictions deviate too much from the actual results, the loss function would generate a large number. In the field of machine learning there is unfortunately not one size fits all loss functions. Each algorithm needs a different form and adaptation of the loss function and the more complex the algorithm (meaning: consisting of several sub algorithms) the more loss functions one needs to define as training indicator for these different parts. In case of Mask R-CNN the loss functions can be divided in sub loss functions as follows [1]:

$$L_{total-loss} = L_{rpn-class-loss} + L_{rpn-bbox-loss} + L_{mrcnn-class-loss} + L_{mrcnn-bbox-loss} + L_{mrcnn-mask-loss}$$

Where loss functions for object classification (both for RPN and Mask R-CNN) and mask detection are defined by certain cross entropy functions having functional forms of :

$$-\sum_{c=1}^{C} q(y_c).\, log[p(y_c)],$$

with $\{q, p\}$being distinct distributions, y_c represent the dataset belonging to either input or predictions. The bounding box loss functions (for RPN and Mask RCNN) on the other hand are represented by so-called smooth L1 functions

$$\frac{K}{2}(y - \hat{y})^2 + (1 - K) * \left(|y - \hat{y}| - \frac{1}{2}\right)$$

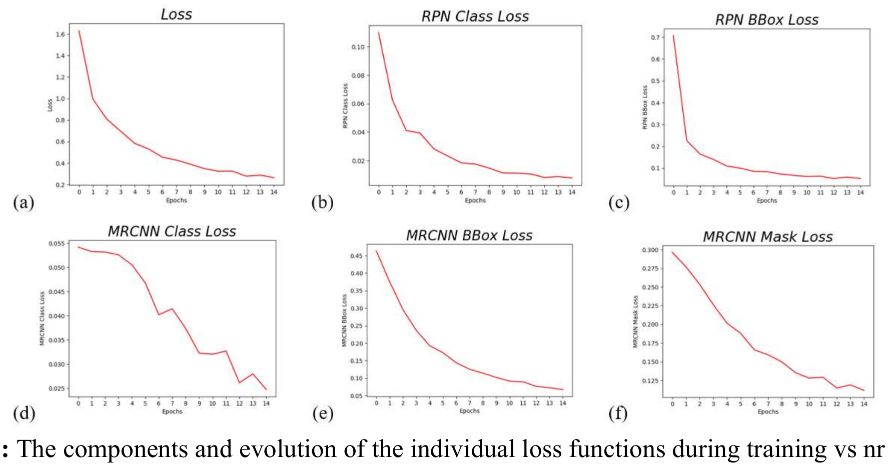


Figure 10: The components and evolution of the individual loss functions during training vs nr of epochs

Above in figure 10 are the evolutions of above loss functions plotted when training the Mask R-CNN. The first graph (a) represents the total loss, while the following graphs (b-f) are respectively the individual loss functions for RPN object classification (b), RPN for bounding box (c), Mask R-CNN for object classification (d), Mask R-CNN for bounding box (e) and Mask R-CNN for mask localization (f)

1. RPN for object classification
2. RPN for bounding box localization
3. Mask R-CNN for object classification
4. Mask R-CNN for bounding box localization
5. Mask R-CNN for mask localization

IV. Results

As explained above, Mask R-CNN is used to detect the area of the dents (i.e., damages) on the given aircraft pictures. From the point of view of the decision makers utilizing the proposed approach, detecting the dents are more important than how precise the area of the dents is calculated. Therefore, this work focuses on detecting the dents correctly and measure the performance by considering how well the dent predictions are made. For this purpose, ***precision*** and ***recall*** are used as our performance metrics. In our case, precision measures the percentage of truly dents classification among all dent predictions made by the network model while recall measures what percentage of the dents are detected by the model. Formally, we calculate the precision and recall as follows:

$$Precision = TP \,/\, (TP + FP)$$
$$Recall = TP \,/\, (TP + FN)$$

Where:

- ***TP*** denotes the true positives and is equal to the number of truly detected dents (i.e., the number of dent predictions, which is correct according to the labeled data).
- ***FP*** denotes the false positives and is equal to the number of falsely detected dents (i.e., the number of dent predictions, which are not correct accordingly to the labeled data).
- ***FN*** denotes the false negatives and is equal to the number of dents, which are not detected by the model (i.e., the number of dents labelled in the original data but the model could not detect them).

Since the number of training samples are not large enough, we use 10-fold cross-validation [27] when we evaluate the performance of the network model in terms of precision and recall. In 10-fold cross validation, we split the original dataset into 10 equal sized parts. To generate training and test datasets, one of this part is taken as test dataset while the rest is used as the training set. We repeat this process 10 times so we create ten different training and test pairs in order to evaluate the performance of the model more robustly.

After training the network model on the training set of each fold and testing on the associated test sets separately, an expert checked and compared the predictions with the labeled data for each fold and calculate the true positives *TP*, false negatives *FN*, and false positives *FP*. Figure 11 shows an example result where the model prediction is shown on the right-hand side, and the manually labelled correspondence on the left-hand side. For this example, the expert observed three dents in the manually labeled picture while two of them were predicted correctly by the model (i.e. 2 true positives) and one of the predictions was not really a dent (i.e. one false positive). Moreover, one of the dents was not detected; therefore it is counted as false negative.

Figure 11: Labeled and predicted aircraft dents

Accordingly, precision and recall values are calculated for each fold as their averages and listed in Table 1. It can be seen that the precision and recall values vary for each fold. Recall the training samples and test samples are different for each fold. Training instances are used for finding the right parameters of the network model while test instances - which are not used during the training of the model, are used to measure the performance of the model. On average, the precision and recall of the model are about 54 per cent and 46 percent respectively. Those results may seem low to the reader; however, considering the number of training instances and the variety of training pictures, they are reasonably acceptable.

Table 1. Precision and Recall Values for 10-Folds

	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	Fold8	Fold9	Fold10	Avg
Train Size	50	50	50	50	50	49	49	49	49	49	49.5
Test Size	5	5	5	5	5	6	6	6	6	6	5.5
TP	2	3	5	5	1	21	4	4	4	8	5.7
FP	2	3	2	3	6	2	4	5	5	6	3.8
FN	3	2	12	10	6	10	3	2	7	6	6.1
Precision	50.0%	50.0%	71.4%	62.5%	14.3%	91.3%	50.0%	44.4%	44.4%	57.1%	**53.6%**
Recall	40.0%	60.0%	29.4%	33.3%	14.3%	67.7%	57.1%	66.7%	36.4%	57.1%	**46.2%**

The performance of the model can be influenced by some factors (also known as hyperparameters) such as number of epochs[6] (i.e., the number of iteration over the training set) in the training phase, the number of training samples as well as their variety. During the training, we take the number of epoch as 15 due to the time limitation. Training session for each fold took about 8-10 hours. In total, it took about 72-96 hours to train the ten folds. 15 epochs may not sound enough to train the entire model. Recall that the pre-trained weight values with the COCO dataset are used

[6] One epoch is when an entire dataset is passed forward and backward through the neural network once to update the Mask R-CNN weights. Since we applied transfer learning with pre-trained weights, there was no need to divide the relatively small dataset in batches and therefore the nr of epochs equals here the nr of iterations. For example: in case we would use 1000 figures and divide that in 4 batches of size 250 then performing 4 iterations to pass 4*250 figures once through the Mask R-CNN would be called one epoch.

as the initial weights instead of random weights. Therefore, even so low number of epochs, 54 percent precision on average was gained. In order to increase the precision and recall, we need to increase the number of training samples as well as their variety. One way is to utilize data augmentation methods [29] and generate new instances from the existing ones by rotating, changing their brightness, etc.

During a detailed analysis on the test dataset, we observed that some environmental factors such as the lightning, reflection, raindrops, rivets may mislead the learning model. For instance, consider the picture on the left hand side in Figure 12, where the dents are manually labeled by an expert. The prediction of our model is shown on the right hand side in Figure 12. As seen, some raindrops or rivets are mistakenly detected as dents.

Another important observation is that in some cases the learning model detects the dents more precisely than the human experts as shown in Figure 13. The picture on the left hand side is the manually labeled whereas the one on the right hand side is labelled by the neural network model. While the human expert labelled single large area on the back wing as a detect, our model indicates several small detects in that region.

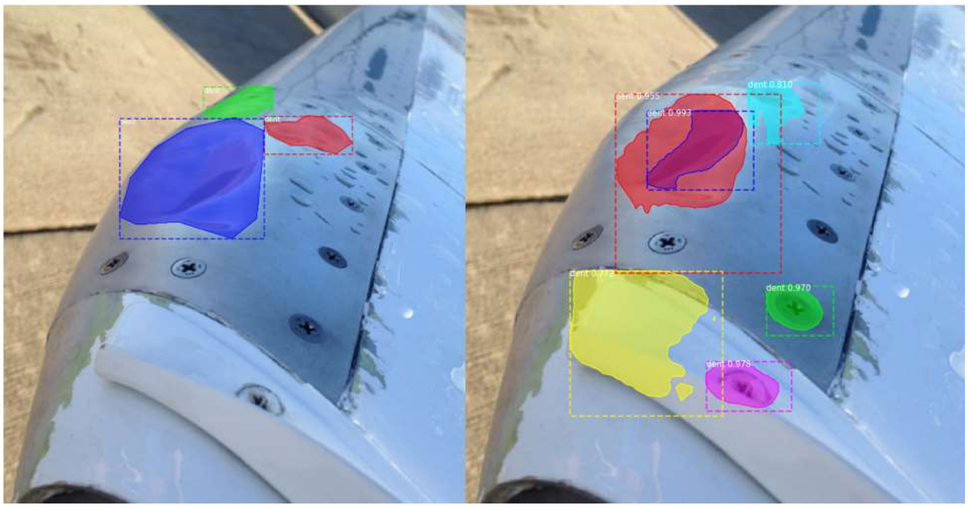


Figure 12: Manually labeled and predicted dents for a false positive sample on Fold 10 Test Set. Note that raindrops are present in the (left / right) picture at the right hand side of the (red / light blue) mask.

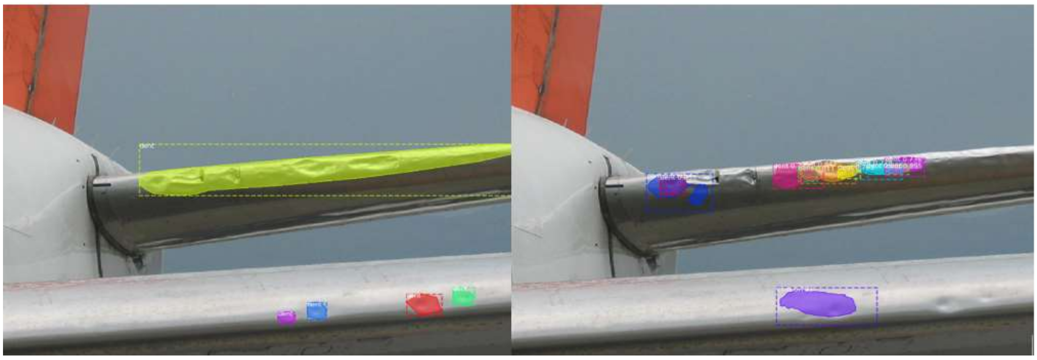


Figure 13: Manually labeled and predicted dents on Fold 10 Test Set.

As explained, we use a pre-trained convolutional neural network by coco domain, and we only updated some of the weights while fixing the others during the training. That is, the RESNET part is fixed while updating the parameters for the rest of the network architecture. After our first training where the results are given in Table 1, we continued the training 5 more epoch by updating the weights of the RESNET during the training. Those results are shown in Table 2. It can be clearly seen the precision and recall values increased significantly after this additional training. On average, we gained 69 percent precision while 57 percent recall. We noticed that the precision calculated for the test set on Fold 5 increased from 14 percent to 80 percent . Figure 14 shows the manually labelled picture (left hand side), the prediction of first model (in the center), and the prediction after the additional training (right hand side). It can be seen that the model achieved to detect the dents after additional 5 epoch training while also updating the weights of RESNET.

Table 2. Precision and Recall Values for 10-Folds when also updating RESNET

	Fold1	**Fold2**	**Fold3**	**Fold4**	**Fold5**	**Fold6**	**Fold7**	**Fold8**	**Fold9**	**Fold10**	**Avg**
Train Size	50	50	50	50	50	49	49	49	49	49	49.5
Test Size	5	5	5	5	5	6	6	6	6	6	5.5
TP	2	3	9	6	4	20	7	4	5	9	6.9
FP	1	1	5	1	1	2	3	6	3	7	3.0
FN	3	2	8	11	2	12	1	2	7	6	5.4
Precision	66.7%	75.0%	64.3%	85.7%	80.0%	90.9%	70.0%	40.0%	62.5%	56.3%	**69.13%**
Recall	40.0%	60.0%	52.9%	35.3%	66.7%	62.5%	87.5%	66.7%	41.7%	60.0%	**57.32%**

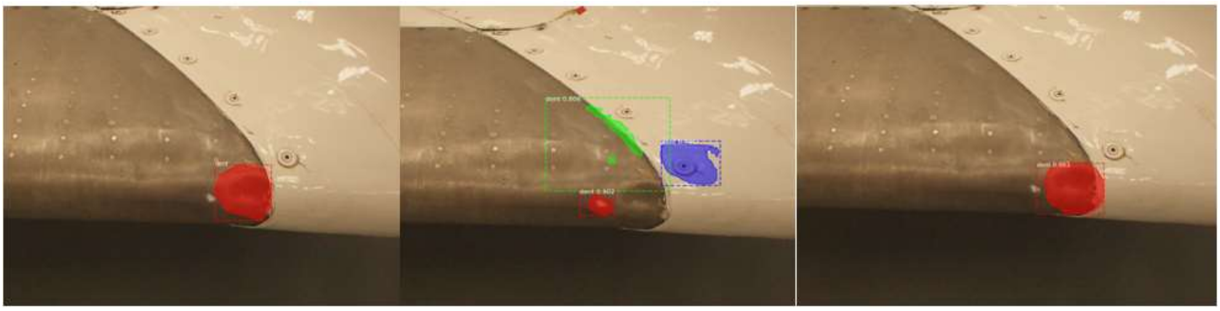


Figure 14: The prediction improvement of a test sample on Fold 5 after additional training

V. Conclusion

This paper applied Mask Region Convolutional Neural Networks to automatically detect aircraft dents. The neural network has been trained with various photos with varying background colors, size, location, and resolution. The photos were taken from the Abu Dhabi Polytechnic hangar and also collected from internet and then manually annotated for dents. In total, 55 photos have been collected, 90% of which were used for training the neural network and the remaining to test the model. Since the number of photos found during the online search was small, it was decided to use the same initial weights for training as in [4] which also used mask R-CNN to detect car dents. This way the amount of input data needed to train the neural network to detect dents on aircraft is minimum. Another complementary approach used to address the small data problem was to use a 10-fold cross validation [27]. Using this approach, the original dataset was split into 10 equally sized parts 9 of which are used for training and 1 for testing.

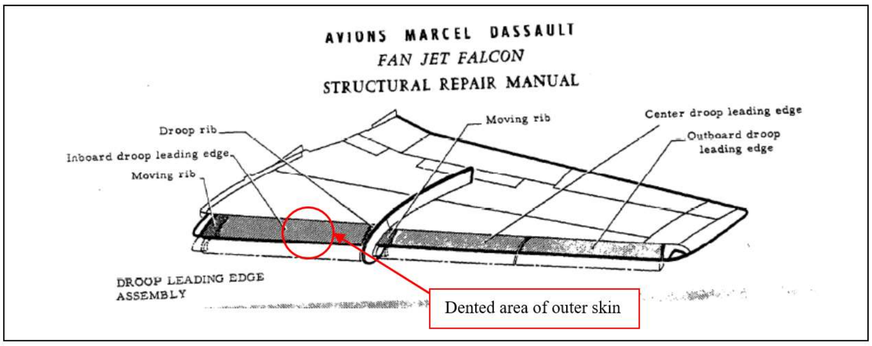
The process was repeated 10 times in order to create 10 different combinations of training and test pairs, and evaluate model performance more robustly.

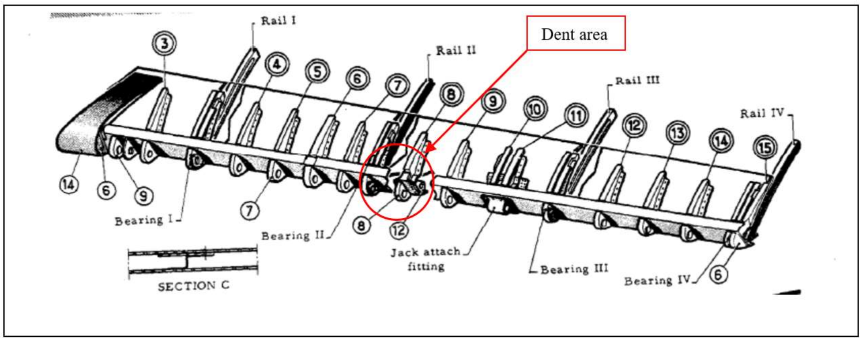
The performance of each fold was evaluated using two main metrics namely precision and recall which are function of true positives, false positives, and false negatives. The precision and recall values corresponding to the 10 folds were different. On average, the precision was 54% with a noticeable low corresponding to fold 5 with 14%, and a maximum value of 91% corresponding to fold 6. The recall value had an average of 46% and followed the same trend with a minimum of 14% corresponding to fold 5 and a maximum of 68% corresponding to fold 6. The averages may seem low but considering the small number of training instances and the variety of pictures used, the results are reasonably acceptable. While analyzing the predictions, two interesting observations were made. 1) rivets and rain drops may mislead the model as they are sometimes detected as dents. Creating new classes for these objects during the annotation could solve this issue. 2) Another observation was that the model sometimes detect dents more elegantly than experts. For instance, in one photo, the model accurately predicted the exact locations of multiple small dents, while in the original photo the expert had only labelled one large area of dents.

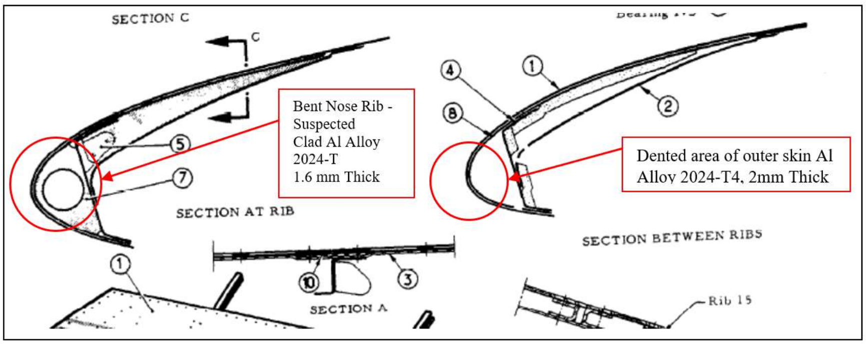
As discussed in the results section, the RESNET layer shown in figure 9 was not included during the initial 15 epochs when transfer learning is applied. During this stage A, the RESNET weights are kept constant, while the layers including RPN, Masking, Bounding Boxes etc., basically the so-called head of the Mask R-CNN structure were trained and fine tuned. Then another 5 epochs were used to continue training the head of the Mask R-CNN structure including the RESNET layer (stage B). In this way the head has been trained for 20 epochs, while the RESNET layer 5 epochs. A considerable improvement in the performance is observed, particularly in the 5-fold cross-validation case, while certain cases (like 10-fold or 6-fold) show minor decreases. However, the overall trend shows an increase in precision and recall. As explained in the methodology section - network training, the RESNET layer functions as a feature extractor. By training / fine tuning it, even though for only 5 epochs, the Mask R-CNN is able to increase the true positives (***TP***) while decreasing the false positives / negatives (***FN***, ***FP***).

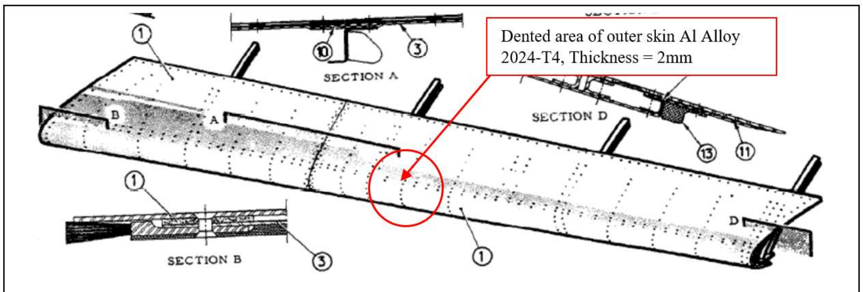
This research can be extended in five different ways: 1) The model can be further trained since this study only used 20 epochs because of computational resources constraints. It would be interesting to see how increasing the number of epochs would affect the precision and recall for both initial training stages A and B. 2) The quality of training data can be further improved through involving at least 3 experienced aircraft engineers during the annotation process. This is important since what represents a damage to one inspector might be acceptable to another inspector. Consensus needs to be reached in order to annotate the photos more accurately. Another measure is to not only annotate dents, but also other objects in the same photo such as rivets and rain drops to increase model performance. 3) Automatic detection of additional types of damage such as holes, cracks, lightning strike entry points, markings, etc. 4) Using Augmentation methods [29] or Generative Adversarial Networks [33][35] (GANs) to generate new data with the same statistics as the training set because of the lack of pixel data, and 5) Physical testing of the technology in the lab. E.g. by using a drone, or equipping a hangar with multiple sensors. The research will determine the right set-up of sensors to improve detection performance.

Appendix A: Falcon 20 Inboard Droop Leading edge









Inboard droop leading edge (Structure)

Acknowledgments

The authors would like to thank Dr. Beshara Sholy from Abu Dhabi Polytechnic for making the aircraft hangar facilities available to us during the experiments, and Youssef Zmarou for the discussions we had during the research project.

References

[1] Girshick, R. "Fast R-CNN," 2015. arXiv:1504.08083v2 https://arxiv.org/pdf/1504.08083.pdf
[2] He, K., Gkioxari, G., Dollar, P., Girshick, R., "Mask R-CNN," 2018. arXiv:1703.06870v3 https://arxiv.org/pdf/1703.06870.pdf
[3] VGG Image Annotator version 1.0.6 http://www.robots.ox.ac.uk/~vgg/software/via/via-1.0.6.html
[4] CNN Application-Detecting Car Exterior Damage(full implementable code) https://towardsdatascience.com/cnn-application-detecting-car-exterior-damage-full-implementable-code-1b205e3cb48c
[5] Drone-based aircraft damage inspection system https://www.mrodrone.net/
[6] Ubisense, 2018, "Ubisense and MRO drone launch world's first smart hangar solution," 11 april. https://www.ubisense.net/news/ubisense-and-mro-drone-launch-worlds-first-smart-hangar-solution
[7] Graham Warwick, Aircraft Inspection drones entering service with airline MROs https://www.mro-network.com/technology/aircraft-inspection-drones-entering-service-airline-mros 4 april 2018.
[8] https://www.donecle.com/
[9] https://www.mro-network.com/technology/aircraft-inspection-drones-entering-service-airline-mros
[10] Boeing 727 Structure Repair Manual ATA 51-10-1 Page 1, May 25/00, "Inspection and Removal of Damage"
[11] Falcon 20 Structure Repair Manual, DMD 11753, November 1964, Revision 31: October 2007
[12] Gina, S., 2017. "Google Brain chief: AI tops humans in computer vision, and healthcare will never be the same," https://siliconangle.com/2017/09/27/google-brain-chief-jeff-dean-ai-beats-humans-computer-vision-healthcare-will-never/
[13] Aerossurance, 2019, "ATR72 Missed Damage: Maintenance Lessons," October 12. http://aerossurance.com/safety-management/atr72-missed-damage/
[14] Aerossurance, 2019, "Hurried Door Inspection Results in Fall From Aircraft," Aug 2019. http://aerossurance.com/safety-management/hurried-door-inspection-fall/
[15] Khan, J,. "Everything you need to know about visual inspection with AI," https://nanonets.com/blog/ai-visual-inspection/
[16] Liu, X, et al., 2019, "A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis," The lancet digital health, Volume 1, ISSUE 6, Pe271-e297, October 01, 2019.
[17] Bai, N., 2019, "Artificial Intelligence That Reads Chest X-Rays Is Approved by FDA," September 12. https://www.ucsf.edu/news/2019/09/415406/artificial-intelligence-reads-chest-x-rays-approved-fda
[18] Carfagno, J., 2019, "5 FDA Approved Uses of AI in Healthcare," July 18. https://www.docwirenews.com/docwire-pick/future-of-medicine-picks/fda-approved-uses-of-ai-in-healthcare/
[19] U.S. Air National Guard photo by Terry Atwell, KC-135 Stratotanker at Selfridge Air National Guard Base, Mich., https://www.127wg.ang.af.mil/News/Article-Display/Article/930979/191st-maintenance-team-keeps-lightning-struck-tanker-flying/
[20] Flight Mechanic, http://www.flight-mechanic.com/common-paint-troubles/
[21] formulanone, 2019, "Airbus A320 Exit/Entry Door," Creative Commons Attribution-Share Alike 2.0 Generic license, retrieved 22/11/2019. https://commons.wikimedia.org/wiki/File:Airbus_A320_Exit_Entry_Door_(40156967861).jpg
[22] Precisionhawk,"Drone-based Aerial Intelligence in Precision Agriculture," https://www.precisionhawk.com/hubfs/PrecisionHawk%20PrecisionAnalytics%20Agriculture%20Solution%20Brief%202019.pdf
[23] Gross, R., "How the Amazon Go Store's AI Works," https://towardsdatascience.com/how-the-amazon-go-store-works-a-deep-dive-3fde9d9939e9
[24] Jovancevic, I, Larnier, S, Orteu, JJ, Sentenac, T,, 2015 "Automated exterior inspection of an aircraft with a pan-tilt-zoom camera mounted on a mobile robot" . Journal of Electronic Imaging, Society of Photo-optical Instrumentation Engineers, 24 (6), pp.061110. <10.1117/1.JEI.24.6.061110>. <hal-01351008>
[25] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.,, 2016 YouOnlyLookOnce: Unified,Real-TimeObjectDetection https://arxiv.org/pdf/1506.02640v5.pdf
[26] Uniting Aviation, 2019, "The future of MRO: emerging technologies in aircraft maintenance," August 1st, https://www.unitingaviation.com/strategic-objective/capacity-efficiency/the-future-of-mro-emerging-technologies-in-aircraft-maintenance/
[27] Alpaydin,, 2010 "Introduction to Machine Learning", Thee MIT Press, https://dl.acm.org/citation.cfm?id=1734076
[28] Brownlee, J, 2019, "A gentle introduction to object recognition with deep learning", in deep learning for computer vision. https://machinelearningmastery.com/object-recognition-with-deep-learning/
[29] Shorten, C. Khoshgoftaar, T. M., 2019, "Survey on Image Data Augmentation for Deep Learning", Journal of Big Data, https://link.springer.com/content/pdf/10.1186%2Fs40537-019-0197-0.pdf

[30] Analyticsvidhya, 2018, “Ultimate guide: building a mask R-CNN model for detecting car damage (with python codes)”, July 19. https://www.analyticsvidhya.com/blog/2018/07/building-mask-r-cnn-model-detecting-damage-cars-python/
[31] Pan, Yang, 2010, “A survey on Transfer Learning”, IEEE Transactions on Knowledge and Data Engineering, https://doi.org/10.1109/TKDE.2009.191
[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, 2016, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks” https://arxiv.org/pdf/1506.01497.pdf
[33] Creswell, A. White, T, Dumoulin V., Arulkumaran, K. Sengupta, B., Bharath, A. “Generative Adversarial Networks: An Overview”, IEEE Signal Processing Magazine Volume: 35 , Issue 1.
[34] Github, “Mask R-CNN 2.0,” Releases, mask_r_cnn_coco.h5, retrieved December 1st, 2019 https://github.com/matterport/Mask_RCNN/releases/download/v2.0/mask_rcnn_coco.h5
[35] Foster, D. 2019, “Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play“, OReilly Media