Chinese Journal of Aeronautics, (2020), xxx(xx): xxx–xxx

Chinese Society of Aeronautics and Astronautics
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn
www.sciencedirect.com

A mask R-CNN based method for inspecting cable brackets in aircraft

Gang ZHAO [a,b], **Jingyu HU** [a], **Wenlei XIAO** [a,b,*], **Jie ZOU** [a]

[a] *School of Mechanical Engineering and Automation, Beihang University, Beijing 100191, China*
[b] *MIIT Key Laboratory of Aeronautics Intelligent Manufacturing, Beihang University, Beijing 100191, China*

Received 16 June 2020; revised 18 September 2020; accepted 18 September 2020

KEYWORDS

Aircraft assembly;
Augmented reality;
Cable bracket;
Mask R-CNN;
Synthetic dataset;
Template matching;
Visual inspection

Abstract In the aviation industry, cable bracket is one of the most common parts. The traditional assembly state inspection method of cable bracket is to manually compare by viewing 3D models. The purpose of this paper is to address the problem of inefficiency of traditional inspection method. In order to solve the problem that machine learning algorithm requires large dataset and manually labeling of dataset is a laborious and time-consuming task, a simulation platform is developed to automatically generate synthetic realistic brackets images with pixel-level annotations based on 3D digital mock-up. In order to obtain accurate shapes of brackets from 2D image, a brackets recognizer based on Mask R-CNN is trained. In addition, a semi-automatic cable bracket inspection method is proposed. With this method, the inspector can easily obtain the inspection result only by taking a picture with a portable device, such as augmented reality (AR) glasses. The inspection task will be automatically executed via bracket recognition and matching. The experimental result shows that the proposed method for automatically labeling dataset is valid and the proposed cable bracket inspection method can effectively inspect cable bracket in the aircraft. Finally, a prototype system based on client-server framework has been developed for validation purpose.

1. Introduction

The quality control is of great importance in the aviation industry.[1] At the same time, the market puts higher requirements on the assembly efficiency of commercial aircraft with the increasing demand. Take the large-scale commercial aircraft C919 made in China as an example, there are about 30,000 cable brackets per C919. However, the traditional assembly state inspection method is to manually compare by viewing 3D models. Therefore, the assembly states inspection of so many brackets is a laborious and time-consuming task,

* Corresponding author at: School of Mechanical Engineering and Automation, Beihang University, Beijing 100191, China.
E-mail address: xiaowenlei@buaa.edu.cn (W. XIAO).

Peer review under responsibility of Editorial Committee of CJA.

Production and hosting by Elsevier

https://doi.org/10.1016/j.cja.2020.09.024
1000-9361 © 2020 Production and hosting by Elsevier Ltd. on behalf of Chinese Society of Aeronautics and Astronautics.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Please cite this article in press as: ZHAO G et al. A mask R-CNN based method for inspecting cable brackets in aircraft, *Chin J Aeronaut* (2020), https://doi.org/10.1016/j.cja.2020.09.024

and smart devices to assist the manual inspection are desired to improve inspection efficiency and quality.

Computer vision has developed for so many years. In particular, the rapid development of deep learning has brought computer vision to a new level.[2] At the same time, visual inspection is widely used in many industrial systems.[3] In this work, a semi-automatic assembly states inspection method for aircraft cable brackets is proposed. With this method, the worker only needs to take a picture with a camera-equipped mobile device to get the inspection results. The whole proposed approach consists of several steps: firstly, the synthetic realistic dataset with pixel-level annotation is automatically generated by our simulation platform; secondly, a bracket recognizer based on Mask R-CNN is trained and applied to segment brackets from image to be inspected (target image)[4]; thirdly, image registration between the target image and the standard global image is conducted to get the corresponding standard partial image; then, assembly states inspection for brackets is executed by automatically comparing the shapes of brackets in the target image with the shapes of corresponding brackets in the standard image; finally, the inspection result is visualized based on augmented reality (AR).

The rest of the paper is organized as follows. In Section 2, related works are reviewed. The proposed method, including dataset generation platform, Mask R-CNN model training and bracket inspection pipeline are described in Section 3. Results and discussion are shown in Section 4. Then, a prototype system based on client–server framework is depicted in Section 5. Finally, the paper is concluded in Section 6.

2. Related works

2.1. Automated visual inspection

Automatic visual inspection (AVI) has been widely used in industrial fields, such as inspection of airplane exterior[5] and structural damage.[6] However, application of AVI about assembly state inspection in the literature is rare. Airbus Innovation Group put forward Smart Augmented Reality Tool (SART) to reduce the time of brackets inspection by superimposing Digital Mock-Up over real scene based on marker technology associated to SLAM algorithms.[7] But it still needs manual comparison and labeling. Biagio, et al.[8] proposed a multi-camera system to address the problem of model checking using support vector machine. But the system is very bulky, which consists of nine cameras.

2.2. Deep learning algorithm

Accurate recognition of bracket is the prerequisite of inspection. Fortunately, algorithms based on deep learning has promoted the development of object detection,[9] semantic segmentation[10] and instance segmentation. At the same time, deep learning is also used to solve problems in the industrial field.[11–15] Object detection algorithm generates bounding boxes around targets, such as Faster R-CNN.[16] Semantic segmentation algorithm classifies every pixel into different class without distinguishing different instance in same class, such as Fully Convolutional Network (FCN),[17] which is mainly used for automated driving.[18] However, instance segmentation algorithm detects every instance and segments them pixel-wise, such as Mask R-CNN.[4]

2.3. Synthetic datasets

Deep learning requires large datasets for training. But, pixel-level annotation is a laborious task. Some researcher has proposed simulation method used to synthesize realistic datasets. Tobin, et al.[19] proposed a domain randomization method to bridge the 'reality gap'.[19] Ros, et al.[20] presented SYNTHIA, a new synthetic dataset for semantic segmentation of driving scenes in a virtual city. Gaidon, et al.[21] presented an efficient real-to-virtual world cloning method and validated their approach by building a new dataset, called 'Virtual KITTI'. In addition, Wang, et al.[22] proposed a synthetic dataset used for visual SLAM evaluation. However, these methods are not used for synthesizing dataset for instance segmentation.

3. Material and methods

In this section, dataset generation platform, Mask R-CNN model training process and bracket inspection pipeline are introduced.

3.1. Dataset generation platform

As other deep learning frameworks, Mask R-CNN requires a large number of images with pixel-level annotation for model training and parameter optimization, but pixel-level annotation of training dataset is a laborious task. Fortunately, there are ready-made 3D digital models in the aircraft manufacturing process. In virtue of simulation technology based on Open Scene Graph (OSG),[23] a platform is developed to automatically generate synthetic realistic images with pixel-level annotations based on 3D digital model. Killing two birds with one stone, the platform can also be used to generate standard global image used for image registration.

3.1.1. Viewpoint sampling

To capture pictures of brackets from various positions and angles, viewpoints are sampled. In the world coordinate system, the ranges of camera position in the x_w, y_w, and z_w directions are given according to the size of product respectively, then the camera position is traversed in the order of x_w, y_w, z_w by setting reasonable Δx, Δy and Δz. At each camera position, the ranges of camera pose in the x_c, y_c, and z_c directions are given respectively in the camera coordinate system, then the camera pose is traversed in the order of α, β, γ by setting reasonable $\Delta\alpha$, $\Delta\beta$ and $\Delta\gamma$. The schematic diagram of viewpoint sampling is shown in Fig. 1.

3.1.2. Pixel-level annotation

At each viewpoint, the realistic RGB image is rendered by setting reasonable materials and light sources. The corresponding mask is generated by rendering background with grayscale value of 0 and rendering brackets with different grayscale value from 1 to N (N is the number of brackets at current viewpoint). In order to make it easier for people to see, different gray values are mapped to various colors by setting a color map. In addition, the related label information is saved as

Please cite this article in press as: ZHAO G et al. A mask R-CNN based method for inspecting cable brackets in aircraft, *Chin J Aeronaut* (2020), https://doi.org/10.1016/j.cja.2020.09.024

yaml format file. In this yaml file, the second line is the background label, and from the third line, the *n*th line is the label of the object whose gray value is *n*-2. Example of synthetic dataset is shown in Fig. 2.

3.1.3. Standard global image generation

In the same way, standard global image used for image registration and its corresponding pixel-level annotation mask can also be rendered by fixing camera in front of the scene. Since the standard global image contains more brackets, a higher image resolution is adopted to ensure that each bracket has enough pixels to make its contour accurate. Example of standard global image and corresponding annotation result is shown in Fig. 3.

3.2. Mask R-CNN model training

In order to obtain the accurate instance segmentation results, a brackets recognizer based on Mask R-CNN is trained.[4] The framework of brackets recognizer based on Mask R-CNN is illustrated in Fig. 4. It consists mainly of two parts: backbone for feature extraction and head for object detection (location and classification) and mask prediction. Region proposal network (RPN)[16] is used to generate candidate bounding box for further instance segmentation. In addition, to obtain the accurate object mask, ROI Align layer is used to map feature map of region of interest (ROI) to a fixed size.[4]

Before model training, transfer learning is introduced by using a pre-trained model based on the COCO dataset to reduce convergence time.[24] To obtain higher recognition accuracy, the deeper network ResNet-101[25] combined with Feature Pyramid Network (FPN)[26] is used as the backbone for feature extraction. Since the bracket is the only thing we care about in the scene, all objects except the brackets are classified as background. Furthermore, according to the number of brackets in each image, 100 regions of interest (ROI) per image are trained to keep a positive : negative ratio of 1 : 3. In addition, to balance the performance and speed, the resolution of input image is 1280×720.

3.3. Bracket inspection pipeline

Once Mask R-CNN model training is completed, brackets inspection task can be executed. With our method, the inspector can get the inspection result only by taking a picture with a portable device, such as AR glasses. The schematic diagram of brackets inspection pipeline is shown in Fig. 5. Firstly, brackets are recognized from images to be inspected (target image) by Mask R-CNN. Secondly, the corresponding standard partial image is obtained by image registration from standard global image to target image. Thirdly, the inspection result will be obtained by bracket matching between brackets in the target image and brackets in the corresponding standard partial image. Finally, the inspection result is visualized based on AR technology.

3.3.1. Brackets recognition

Accurate recognition of brackets is a prerequisite for inspection. The Mask R-CNN model trained on our synthetic realistic dataset is used to recognize brackets in the target image and get the target masks.

3.3.2. Image registration

After recognizing brackets in the target image, the corresponding standard partial image is expected. In the field of computer vision, image registration is a fundamental task for matching two pictures taken from different viewpoints, including feature based method and template based method.[27] However, featured based method, such as scale-invariant feature transform (SIFT), is more suitable for images with rich textures.[28] Therefore, template based image registration is executed from standard global image in the images library to target image by following steps:

Firstly, as a result of different scale between standard global image and target image, a multi-scale transformation of integrated standard global mask is executed.[29] Then, template matching between integrated target mask and integrated standard global mask with different scale is used to determine the

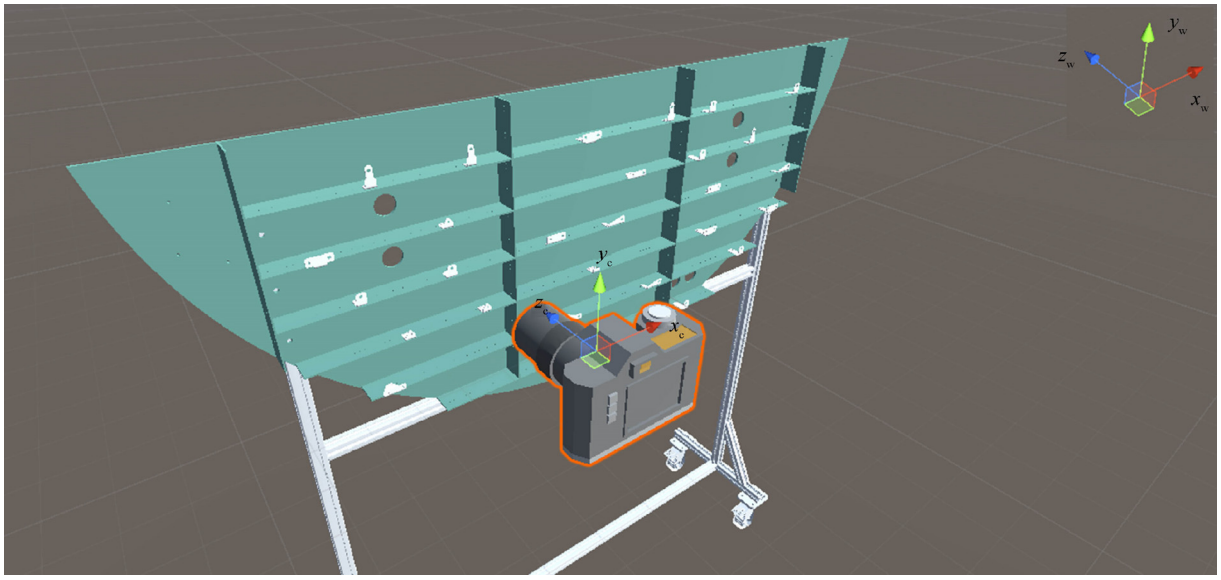
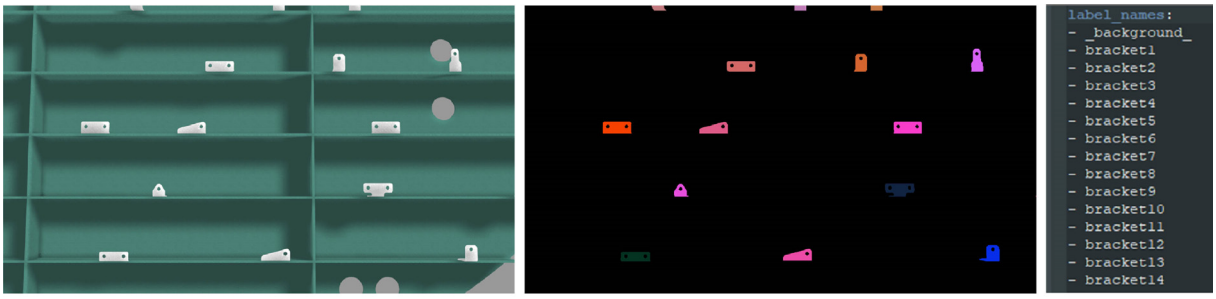
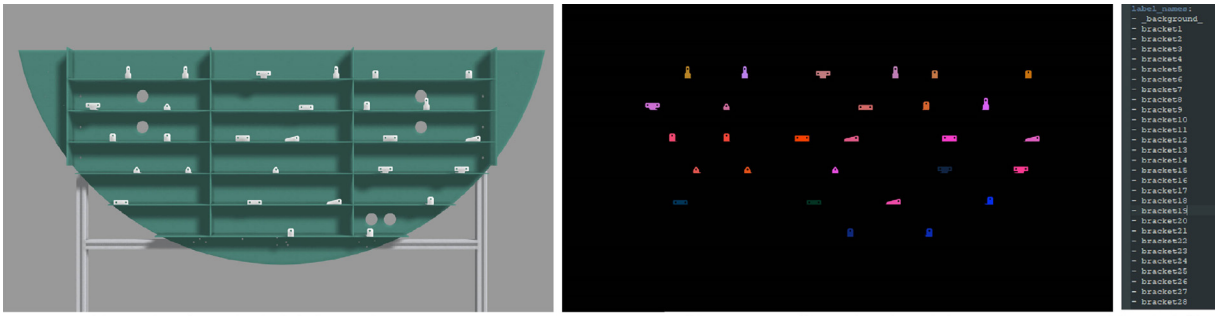


Fig. 1 Schematic diagram of viewpoint sampling.



(a) Rendered realistic RGB image (b) Corresponding colored mask image (c) Label file

Fig. 2 Example of synthetic dataset.



(a) Rendered standard global RGB image (b) Corresponding colored mask image (c) Label file

Fig. 3 Example of standard global image and corresponding annotation result.

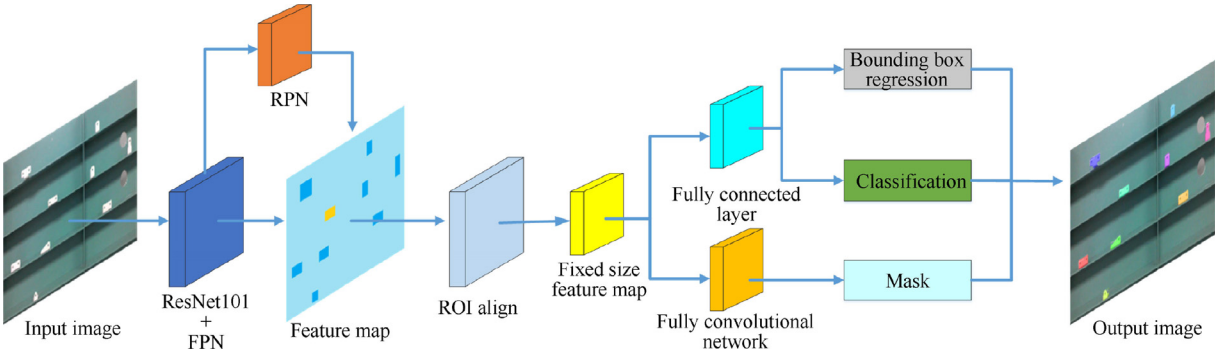


Fig. 4 Framework of brackets recognizer based on Mask R-CNN.

scale and position of target image.[30,31] The schematic diagram of multi-scale template matching is illustrated in Fig. 6.

Secondly, according to the multi-scale template matching result, the projection position of the target image's four corner points in the standard global image can be obtained. Then, four pairs of corner points are used to calculate homography matrix $\boldsymbol{H}_1$ between target image and standard global image as follows[32]:

$$\begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} = \boldsymbol{H}_1 \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{1}$$

where (x_i, y_i) is the coordinate of target image's corner point in the standard global image coordinate system, (x_i', y_i') is the coordinate of target image's corner point in the target image coordinate system. $\boldsymbol{H}_1$ represents the homography transformation between standard global image and target image, which is a 3×3 matrix and $h_{33} = 1$.

Thirdly, the above homography transformation $\boldsymbol{H}_1$ is applied to the standard global image and its masks to get the intermediate standard partial image and intermediate standard partial masks corresponding to the target image.

In general, the results of the above registration are insufficient. With the help of mask of every bracket, the centroid of every bracket is calculated. Then, Iterative Closest Point (ICP) algorithm is used for fine registration to get the perspective transformation $\boldsymbol{H}_2$ between target image and intermediate standard partial image.[33]

The schematic diagram of the initial value determining process of the ICP algorithm is shown in Fig. 7. After getting the

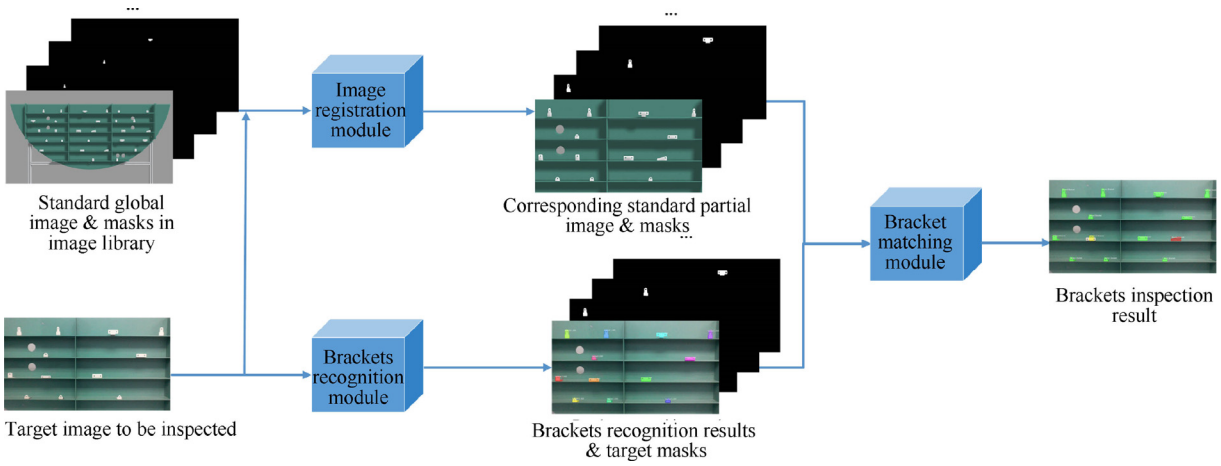


Fig. 5 Schematic diagram of brackets inspection pipeline.

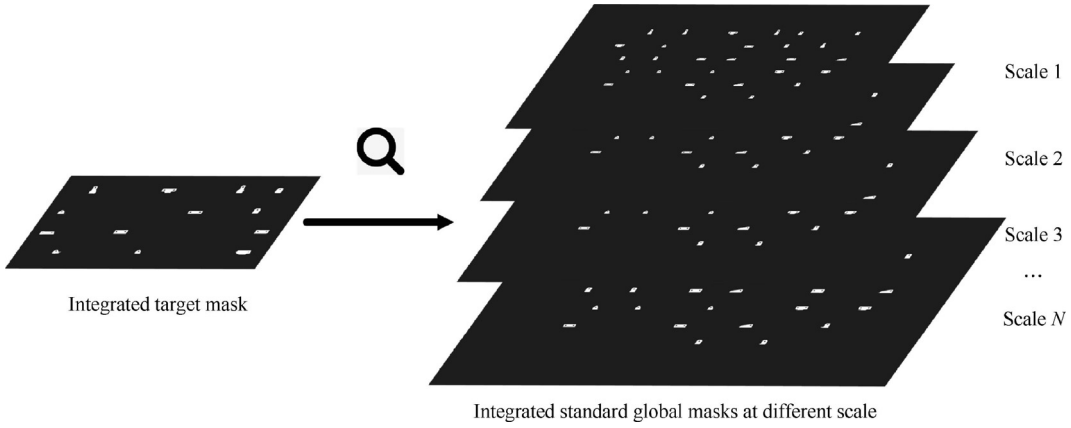


Fig. 6 Schematic diagram of multi-scale template matching.

coarse registration result by multi-scale template matching, the homography transformation $\boldsymbol{H}_1$ is applied to the standard global image and its masks to get the intermediate standard partial image and intermediate standard partial masks corresponding to the target image.

With the help of mask of every bracket, centroid of every bracket in both the target image and the intermediate standard partial image is calculated to generate two point sets: one is the centroid sets of brackets in the target image, and another one is the centroid sets of brackets in the intermediate standard partial image.

For every centroid of bracket in the target image, find the corresponding nearest centroid of bracket in the intermediate standard partial image. These point pairs are the initial value of ICP algorithm.

The final perspective transformation $\boldsymbol{H}$ is calculated as $\mathbf{H} = \mathbf{H}_1 \times \mathbf{H}_2$.

Finally, the standard partial image corresponding to the target image can be obtained by applying the perspective transformation $\boldsymbol{H}$ to the standard global image.

3.3.3. Bracket matching

Bracket matching between brackets in the target image and brackets in the corresponding standard partial image is used for brackets inspection. For every bracket in the target image, if there is no bracket at the same position in the corresponding standard partial image, it means this bracket is redundant, otherwise intersection over union (IoU) between mask of bracket in the target image and that in the corresponding standard partial image will be calculated to represent the similarity,[34] which is calculated by:

$$\mathrm{IoU}\,(\mathrm{mask_t}, \mathrm{mask_s}) = \frac{I(\mathrm{mask_t}, \mathrm{mask_s})}{U(\mathrm{mask_t}, \mathrm{mask_s})} \tag{2}$$

where $\mathrm{mask_t}$ is the binary mask of bracket in the target image, $\mathrm{mask_s}$ is the binary mask of bracket in the corresponding standard partial image. $I(\mathrm{mask_t}, \mathrm{mask_s})$ represents the intersection of $\mathrm{mask_t}$ and $\mathrm{mask_s}$. $U(\mathrm{mask_t}, \mathrm{mask_s})$ represents the union of $\mathrm{mask_t}$ and $\mathrm{mask_s}$. The bracket is considered to be installed correctly only if IoU is greater than a given threshold. In addition,

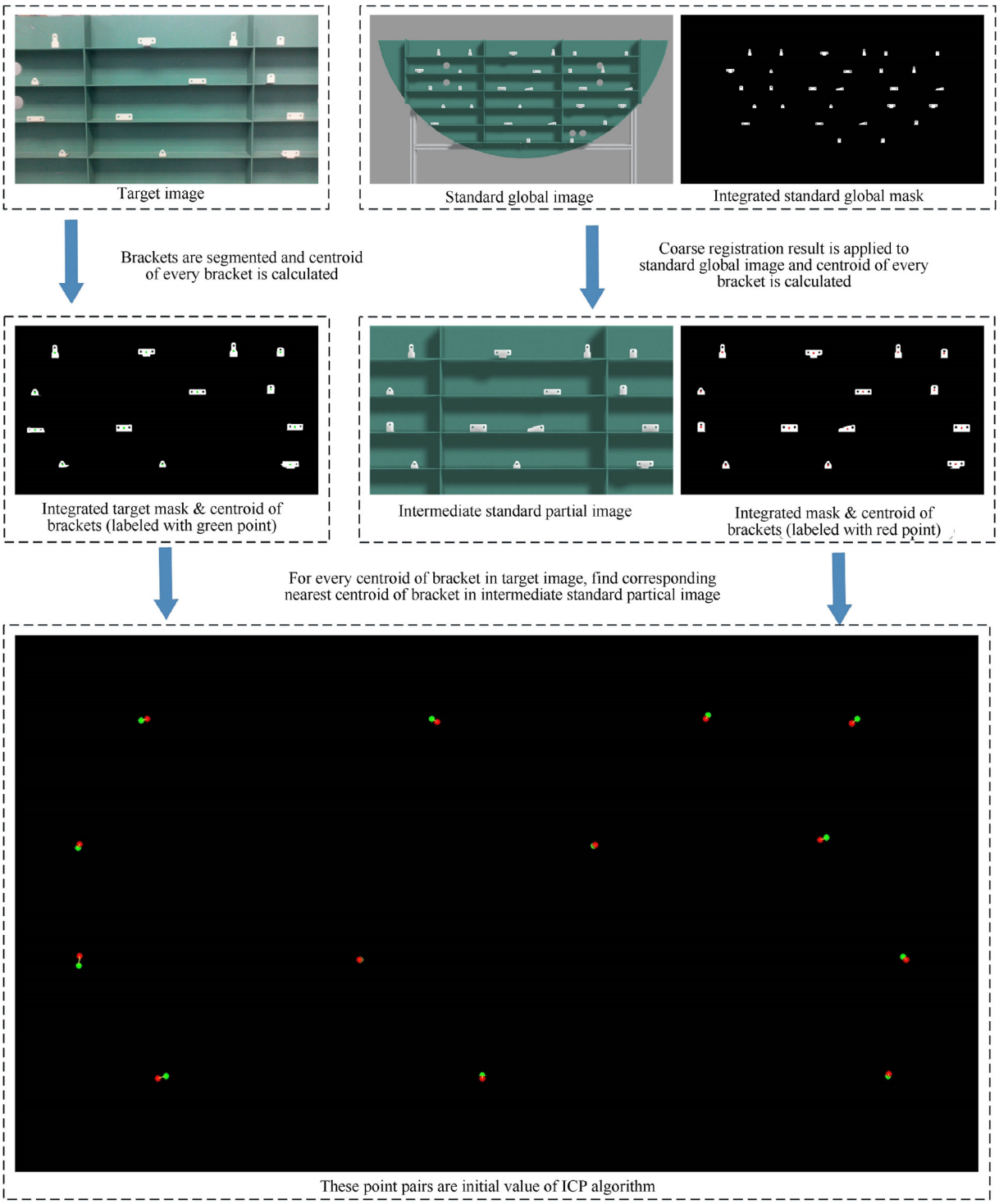


Fig. 7 Schematic diagram of initial value determining process of ICP algorithm.

for every bracket in the corresponding standard partial image, if there is no bracket at the same position in the target image, it means this bracket is missed. The algorithm of bracket matching is as follows:

Algorithm: Bracket matching for brackets inspection

Inputs:
$mask_{ti}$ {The ith bracket mask in the target image}
N_t {The number of bracket masks in the target image}
$\boldsymbol{mask_{sj}}$ {The jth bracket mask in the standard partial image corresponding to target image}
N_s {The number of bracket masks in the standard partial image corresponding to target image}
Output:
$Bracket_{missed}$ {missed bracket}
$Bracket_{redundant}$ {redundant bracket}
$Bracket_{correct}$ {correct bracket}
$Bracket_{incorrect}$ {incorrect bracket}
Algorithm:
for $mask_{ti} \leftarrow mask_{t1}$ ***to*** $mask_tN_t$ ***do***
if there is no mask at the same position in the corresponding standard partial image; $bracket_{ti}$ is marked as $Bracket_{redundant}$.
if there is a $mask_s$ at the same position in the corresponding standard partial image and $IoU(mask_s, mask_{ti}) > Threshold_{iou}$; $mask_{ti}$ is marked as $Bracket_{correct}$.
if there is a $mask_s$ at the same position in the corresponding standard partial image and $IoU(mask_s, mask_{ti}) \leq Threshold_{iou}$; $mask_{ti}$ is marked as $Bracket_{incorrect}$.
for $mask_{sj} \leftarrow mask_{s1}$ **to** $mask_s$ N_s **do**
if there is no a mask at the same position in the target image; $mask_{sj}$ is marked as $Bracket_{missed}$.

Examples of bracket matching results are shown in Fig. 8, in which the bracket recognition results (represented with green color mask) and the standard bracket masks obtained by image registration (represented with red color mask) are superimposed on the target image, and bracket matching results are labeled above the brackets.

3.3.4. Visualizing the inspection result

After getting the inspection result, visualization module is adopted for the convenience of inspector. For every bracket, it will be labeled with various colors, such as green color for correct bracket, red color for missed bracket, and yellow color for incorrect bracket.

However, two-dimensional target image annotated with inspection result is insufficient. In order to view the inspection results from all aspects, a visualization method based on AR is presented.

AR is a visualization technology used for augmenting real world with virtual information, such as 3D model, text and so on.[35] It has developed for decades and widely used in the industry field.[36–38] Fortunately, there are many mature commercial AR tools on the market, such as Wikitude[39] and Vuforia.[40] With the help of Model Target of Vuforia, a three-dimensional visualization module is developed. The correct brackets are superimposed on the nearby of target brackets to be inspected. Demonstration of three-dimensional visualization are shown in Fig. 9.

4. Results and discussion

In order to verify the effectiveness of synthetic dataset, experiment is carried out in a laboratory environment. The brackets recognizer based on Mask R-CNN[4] is trained on real dataset and synthetic dataset separately. The real dataset is annotated with traditional tool named LabelMe[41] During the experiments, 240 real images (2513 brackets) and 17,955 synthetic images (247300 brackets) is selected separately for training (80% images are used as training set and 20% images are used as validation set). In addition, 100 real images (970 brackets) is taken as testing set.

Fig. 8 Examples of bracket matching result.

Fig. 9 Demonstration of three-dimensional visualization based on AR.

4.1. Evaluation of brackets detection result

The precision and recall are used for evaluating brackets detection results, which are calculated as follows[42]:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4}$$

where TP represents the number of cases which are positive and detected as positive, FP represents the number of cases which are negative and detected as positive, and FN represents the number of cases which are positive and detected as negative. The precision indicates whether detection result is accurate or not. The recall indicates whether detection result is complete or not. In Tables 1–4, detection result trained on real dataset and trained on synthetic dataset are compared at different IoU_{bb} threshold. IoU_{bb} is calculated as follows:

$$\text{IoU}_{\text{bb}} = \frac{I(\text{BoundingBox}_{\text{g}}, \text{BoundingBox}_{\text{p}})}{U(\text{BoundingBox}_{\text{g}}, \text{BoundingBox}_{\text{p}})} \tag{5}$$

where $\text{BoundingBox}_{\text{g}}$ represents the ground truth bounding box, $\text{BoundingBox}_{\text{p}}$ represents the predicted bounding box. I ($\text{BoundingBox}_{\text{g}}$, $\text{BoundingBox}_{\text{p}}$) represents the intersection area of ground truth bounding box and predicted bounding box. U($\text{BoundingBox}_{\text{g}}$, $\text{BoundingBox}_{\text{p}}$) represents the union area of ground truth bounding box and predicted bounding box.

According detection result trained on real dataset @ IoU_{bb} = 0.50, the recall is 99.18%, the precision is 99.48%. According detection result trained on real dataset @ IoU_{bb} = 0.75, the recall is 90.00%, the precision is 90.28%. According detection result trained on synthetic dataset @ IoU_{bb} = 0.50, the recall is 99.79%, the precision is 99.49%. According detection result trained on synthetic dataset @ IoU_{bb} = 0.75, the recall is 96.49%, the precision is 96.20%. Thus, it can be seen that the precision and recall trained on our synthetic dataset is higher. That is to say that our synthetic dataset is valid and more accurate.

Table 1 Detection result trained on real dataset @ IoU_{bb} = 0.50.

Predicted class	Ground truth	
	Bracket	Background
Bracket	962	5
Background	8	

Table 2 Detection result trained on real dataset @ IoU_{bb} = 0.75.

Predicted class	Ground truth	
	Bracket	Background
Bracket	873	94
Background	97	

Table 3 Detection result trained on synthetic dataset @ IoU_{bb} = 0.50.

Predicted class	Ground truth	
	Bracket	Background
Bracket	968	5
Background	2	

Table 4 Detection result trained on synthetic dataset @ IoU_{bb} = 0.75.

Predicted class	Ground truth	
	Bracket	Background
Bracket	936	37
Background	34	

4.2. Evaluation of brackets segmentation result

The segmentation result is evaluated by pixel-level mean IoU, which is calculated as follows:

$$\bar{\text{IoU}} = \frac{1}{N_{\text{t}}} \sum_{i=1}^{N_{\text{t}}} \text{IoU}_i(\text{mask}_{\text{g}i}, \text{mask}_{\text{p}i}) \tag{6}$$

$$\text{IoU}_i(\text{mask}_{\text{g}i}, \text{mask}_{\text{p}i}) = \frac{I_i(\text{mask}_{\text{g}i}, \text{mask}_{\text{p}i})}{U_i(\text{mask}_{\text{g}i}, \text{mask}_{\text{p}i})} \tag{7}$$

where $\text{mask}_{\text{g}i}$ is the ground truth binary mask of ith bracket, $\text{mask}_{\text{p}i}$ is the predicted binary mask of ith bracket and N_{t} is the number of brackets in the target image. $I_i(\text{mask}_{\text{g}i}, \text{mask}_{\text{p}i})$ represents the intersection of $\text{mask}_{\text{g}i}$ and $\text{mask}_{\text{p}i}$. $U_i(\text{mask}_{\text{g}i}, \text{mask}_{\text{p}i})$ represents the union of $\text{mask}_{\text{g}i}$ and $\text{mask}_{\text{p}i}$. IoU_i(-

$mask_{gi}$, $mask_{pi}$) represents the intersection over union of $mask_{gi}$ and $mask_{pi}$.

According to segmentation result trained on real dataset, the mean IoU is 89.08%. According to segmentation result trained on synthetic dataset, the mean IoU is 84.51%, which is slightly lower. The reason is that the testing set is manually annotated and not accurate enough.

The examples of recognition result trained on real dataset and recognition result trained on synthetic dataset are compared in Fig. 10.

4.3. Evaluation of brackets inspection result

There are 89 images which are registered correctly among 100 images to be inspected. The reason for the low success rate of image registration is that the number of incorrect brackets in those pictures is nearly half. The inspection result statistics is shown in Table 5.

The accuracy of inspection result is evaluated as follows:

$$\text{Accuracy} = \frac{N_{\text{right}}}{N_{\text{total}}} \tag{8}$$

where N_{right} represents the number of brackets whose inspection result is right, N_{total} represents the total number of brackets. According to inspection result of 89 images, the inspection accuracy is 90.67%. Although the inspection accuracy is not very high, the 621 brackets that are judged to be correct are all correctly installed brackets. The inspector only needs to manually inspect those brackets that are judged to be incorrect to ensure 100% accuracy. This means that the system can significantly improve the inspection efficiency.

The examples of image registration result and inspection result based on recognition result trained on synthetic dataset are shown in Fig. 11.

4.4. Evaluation of brackets detection, segmentation and inspection efficiency

In this experiment, NVIDIA Quadro P4000 is used for GPU acceleration. Since detection and segmentation are two parallel computing branches of the Mask R-CNN, the time consumption of detection and segmentation is counted together. Through the statistical analysis of the detection and segmentation time of 89 images, the average detection and segmentation time is 1.259 s per image. In addition, thorough the statistical analysis of the inspection time of 89 images, the average inspection time is 8.803 s per image. In summary, the average total time, including detection, segmentation and inspection, is 10.062 s per image. And there are about 6–16 brackets per

(a) Original image (b) Trained on real dataset (c) Trained on synthetic dataset

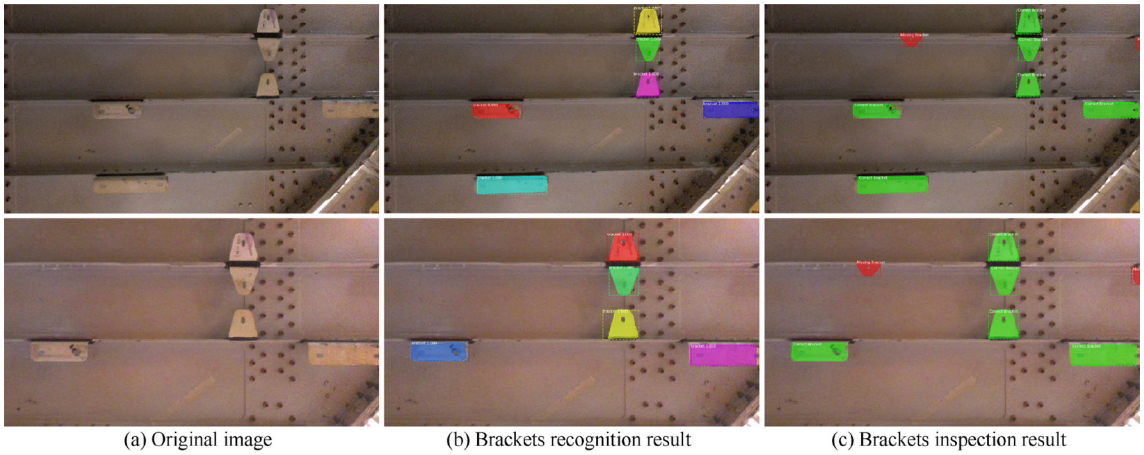
Fig. 10 Comparison of recognition result trained on real dataset and recognition result trained on synthetic dataset.

Table 5 Inspection result.

Ground truth	Inspection result			
	Correct	Incorrect	Missing	Redundant
Correct	621	22	16	0
Incorrect	0	121	53	0
Missing	0	0	87	0
redundant	0	0	0	55

(a) Original image (b) Image registration result (c) Inspection result

Fig. 11 Image registration result and inspection result based on recognition result trained on synthetic dataset.



(a) Original image (b) Brackets recognition result (c) Brackets inspection result

Fig. 12 Examples of experimental results in real aircraft.

image. By calculating the total time divided by the total number of brackets, we can draw a conclusion that it takes about 1.015 s to inspect a bracket. However, the inspection time of traditional method is about 10 s per bracket.

In addition, how to further improve efficiency will be considered in future work, such as optimization of instance segmentation model and development of faster template matching algorithm.

4.5. Validation in the real aircraft

With the help of Commercial Aircraft Corporation of China Ltd (COMAC), the intelligent inspection system was validated in the C919 aircraft. Examples of experimental results are shown in Fig. 12.

5. Prototype system

A cloud-based prototype system is developed to solve the problem of insufficient computing performance of mobile terminals,[43] which is depicted in Fig. 13.

5.1. Client application

To adapt to different mobile terminals, such as Pad, AR glasses and mobile phone, inspection application is developed by Unity, which is a cross-platform game engine.[44] The functions of this client application include personnel login, taking pictures, sending pictures to the server, receiving the brackets recognition result, sending inspection instructions to the server

Please cite this article in press as: ZHAO G et al. A mask R-CNN based method for inspecting cable brackets in aircraft, *Chin J Aeronaut* (2020), https://doi.org/10.1016/j.cja.2020.09.024

Fig. 13 Experimental deployment of cloud-based system in laboratory.

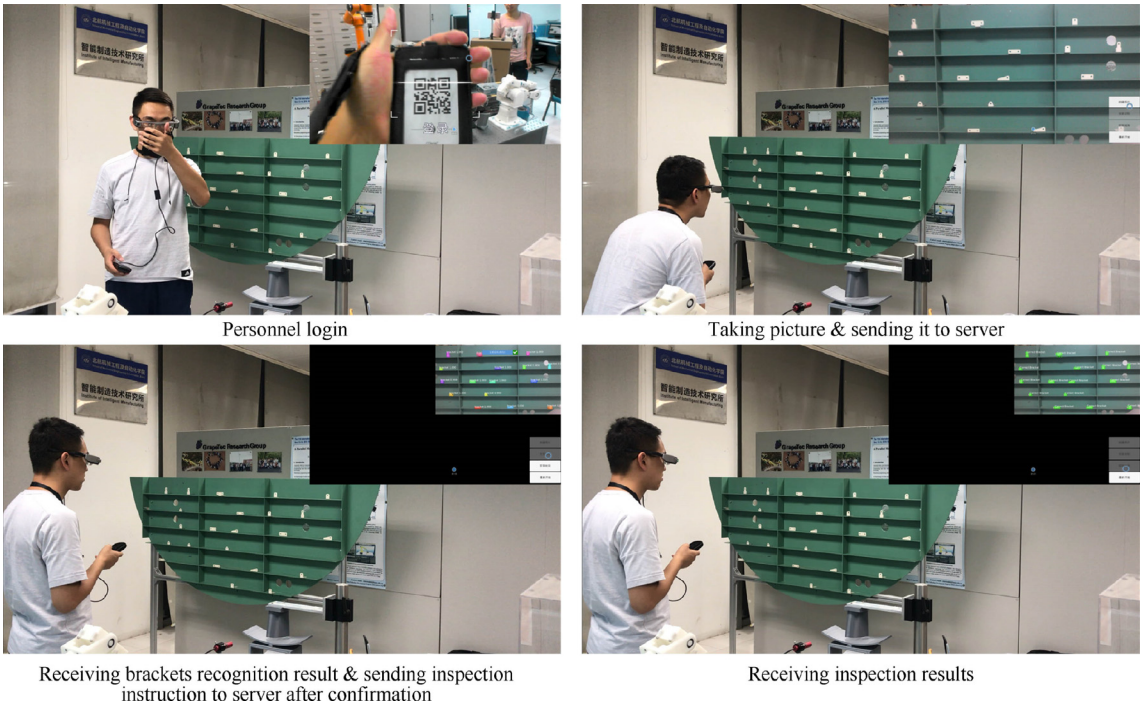


Fig. 14 Demonstration of client application.

after confirmation, and receiving inspection results. Demonstration of client application is shown in Fig. 14.

5.2. Cloud service

The framework of the cloud-based prototype system is shown in Fig. 15. To ensure the functional scalability of the cloud service, business logic and business are separated. The cloud consists of three major components: AI Server is dedicated to running deep learning-related algorithms, which requires higher processing performance. Manager Server is used for personnel management and task deployment. Grape Server, as the center of the entire cloud architecture, is responsible for processing business logic. The client application makes a

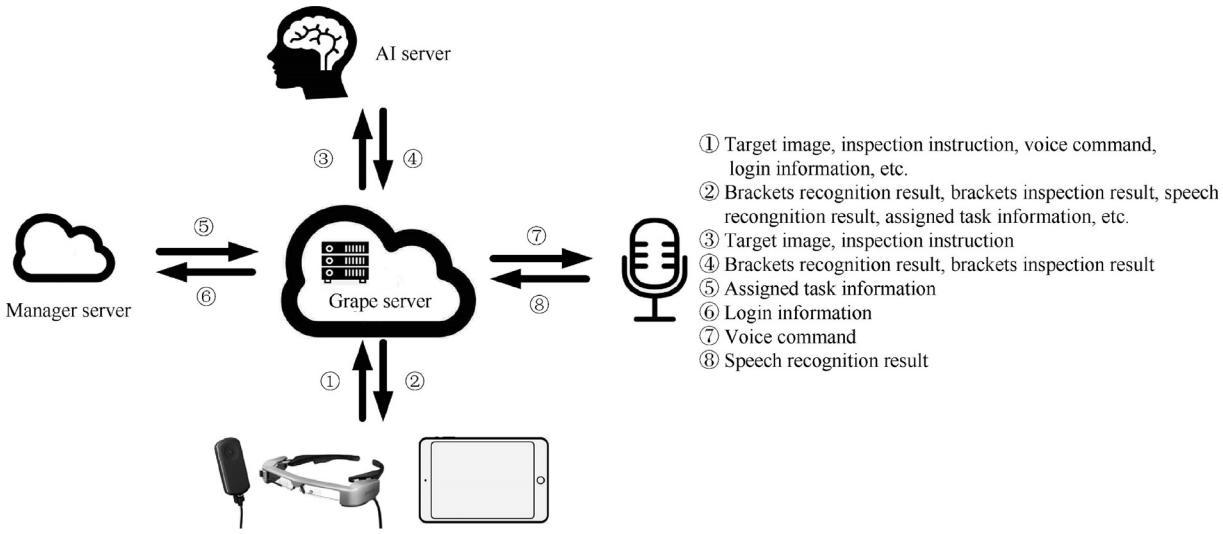


Fig. 15 Framework of a cloud-based prototype system.

business request to Grape Server. Then, Grape Server decides whether to forward to Manager Server or AI Server according to the business type, and returns the processing result of Manager Server or AI Server to the client. With this framework, the cloud will easily implement functional extensions. For example, speech recognition is available by connecting a commercial speech recognition service to Grape Server, which can be used for voice control of client application and liberate inspectors' hands.

6. Conclusion

This paper proposes a Mask R-CNN based method for inspecting cable brackets in the aircraft to improve inspection efficiency. The specific work is summarized as follows:

(1) A simulation platform is developed to automatically generate synthetic realistic images with pixel-level annotations based on 3D digital model, which can significantly save the time cost of data labeling.
(2) The brackets recognizer based on Mask R-CNN is trained, which can automatically detect and segment brackets. Brackets detection result of Mask R-CNN model trained on our synthetic dataset shows that the average precision is 99.79%, the recall is 99.49%, and the mean intersection over union rate for instance segmentation is 84.51%, which means that synthetic dataset is effective.
(3) A semi-automatic assembly states inspection method for aircraft brackets is proposed. The inspection result shows that the inspection accuracy is 90.67%. The reason for the low inspection accuracy is that there are more wrong brackets than correct brackets in some of the target images, which results in low success rate of image registration. Although the inspection accuracy is not very high, the 621 brackets that are judged to be correct are all correctly installed brackets, which means that the system can significantly improve the inspection efficiency.
(4) A cloud-based prototype system to solve the problem of insufficient computing performance of mobile terminals is developed.
(5) Future work will focus on optimization and comparison of instance segmentation model used for more complex scenarios and development of faster template matching algorithm.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is supported by the Civil Airplane Technology Development Program.

References

1. Mei Z, Maropoulos PG. Review of the application of flexible, measurement-assisted assembly technology in aircraft manufacturing. *Proc Inst Mech Eng, Part B: J Eng Manufacture* 2014;**228**(10):1185–97.
2. Khan AI, Al-Habsi S. Machine learning in computer vision. *Procedia Comput Sci* 2020;**167**:1444–51.
3. Silva RL, Rudek M, Szejka AL, et al. Machine vision systems for industrial quality control inspections. *IFIP Adv Inf Commun Technol* 2018;**540**:631–41.
4. He K, Gkioxari G, Dollar P, et al. Mask R-CNN. *IEEE Trans Pattern Anal Mach Intell* 2020;**42**(2):386–97.
5. Jovancevic I, Orteu JJ, Sentenac T, et al. Automated visual inspection of an airplane exterior. In: Meriaudeau F, Aubreton O, editors. *12th international conference on quality control by artificial vision*; 2015 Jun 3-5; Le Creusot, France. Bellingham: SPIE-INT SOC Optical Engineering; 2015. p. 95340Y.
6. Cha YJ, Choi W, Suh G, et al. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Comput Civ Infrastruct Eng* 2018;**33**(9):731–47.
7. Airbus S.A.S. [Internet]. French: Airbus S.A.S.; c2020 [updated 2019 Apr; cited 2020 Jun 10]. Available from: https://www.

airbus.com/newsroom/press-releases/en/2016/04/Airbus-Group-Unit-Testia-to-Supply-To-Spirit-AeroSystems.html.
8. Biagio MS, Beltrán-González C, Giunta S, Bue AD, et al. Automatic inspection of aeronautic components. *Mach Vis Appl* 2017;**28**(5-6):591–605.
9. Zhao Z-Q, Zheng P, Xu S-T, et al. Object detection with deep learning: a review. *IEEE Trans Neural Netw Learning Syst* 2019;**30**(11):3212–32.
10. Garcia-Garcia A, Orts-Escolano S, Oprea S, Villena-Martinez V, Martinez-Gonzalez P, et al. A survey on deep learning techniques for image and video semantic segmentation. *Appl Soft Comput* 2018;**70**:41–65.
11. Bian X, Lim SN, Zhou N. Multiscale fully convolutional network with application to industrial inspection. *2016 IEEE winter conference on applications of computer vision (WACV 2016)*; 2016 Mar 7-10; Lake Placid, New York. New York: IEEE; 2016. p. 1-8.
12. LI X, LI J, QU Y, HE D. Semi-supervised gear fault diagnosis using raw vibration signal based on deep learning. *Chin J Aeronaut* 2020;**33**(2):418–26.
13. Park K-B, Kim M, Choi SH, et al. Deep learning-based smart task assistance in wearable augmented reality. *Rob Comput Integr Manuf* 2020;**63**:101887.
14. Li L, Ota K, et al. Deep learning for smart industry: efficient manufacture inspection system with fog computing. *IEEE Trans Ind Inf* 2018;**14**(10):4665–73.
15. Yu Y, Zhang K, Yang Li, et al. Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. *Comput Electron Agric* 2019;**163**:104846.
16. Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell* 2017;**39**(6):1137–49.
17. Shelhamer E, Long J, Darrell T. Fully convolutional networks for semantic segmentation. *IEEE Trans Pattern Anal Mach Intell* 2017;**39**(4):640–51.
18. Petrovai A, Nedevschi S. Efficient instance and semantic segmentation for automated driving. 30th IEEE intelligent vehicles symposium (IV); 2019 Jun9-12; Paris, France. New York: IEEE; 2019. p. 2575-81.
19. Tobin J, Fong R, Ray A, et al. Domain randomization for transferring deep neural networks from simulation to the real world. In: Bicchi A, Okamura A, editors. *IEEE/RSJ international conference on intelligent robots and systems (IROS) / workshop on machine learning methods for high-level cognitive capabilities in robotics*; 2017 Sep 24-28; Vancouver, Canada. New York: IEEE; 2017. p. 23-30.
20. Ros G, Sellart L, Materzynska J, et al. The SYNTHIA dataset: a large collection of synthetic images for semantic segmentation of urban scenes. *2016 IEEE conference on computer vision and pattern recognition (CVPR)*; 2016 Jun 27-30; Las Vegas, NV. New York: IEEE; 2016. p. 3234-43.
21. Gaidon A, Wang Q, Cabon Y, et al. Virtual worlds as proxy for multi-object tracking analysis. *2016 IEEE conference on computer vision and pattern recognition (CVPR)*; 2016 Jun 27-30; Las Vegas, NV. New York: IEEE; 2016. p. 4340-9.
22. Wang S, Yue J, Dong Y, He S, Wang H, et al. A synthetic dataset for Visual SLAM evaluation. *Rob Auton Syst* 2020;**124**:103336.
23. OpenSceneGraph.org [Internet]. [updated 2019 Jul 31; cited 2020 Jun 12]. Available from: http://www.openscenegraph.org/.
24. Lin TY, Maire M, Belongie S, et al. Microsoft COCO: common objects in context. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T, editors. *13th European conference on computer vision (ECCV)*; 2014 Sep 6-12; Zurich, Switzerland; Berlin: Springer; 2014. p. 740-55.
25. He KM, Zhang XY, Ren SQ, et al. Deep residual learning for image recognition. *2016 IEEE conference on computer vision and pattern recognition (CVPR)*; 2016 Jun 27-30; Las Vegas, NV. New York: IEEE; 2016. p. 770-8.
26. Lin TY, Dollar P, Girshick R, et al. Feature pyramid networks for object detection. *30th IEEE/CVF conference on computer vision and pattern recognition (CVPR)*; 2017 Jul 21-26; Honolulu, HI. New York: IEEE; 2017. p. 936-44.
27. Brown LG. A survey of image registration techniques. *ACM Comput Surv* 1992;**24**(4):325–76.
28. Lowe DG. Distinctive image features from scale-invariant keypoints. *Int J Comput Vis* 2004;**60**(2):91–110.
29. Rosenfeld A. Some uses of pyramids in image processing and segmentation. *Proc DARPA Imaging Underst Work* 1980;112–20.
30. Kim HY, de Araujo SA. Grayscale template-matching invariant to rotation, scale, translation, brightness and contrast. *Adv. Image Video Technol* 2007;**4872**:100–13.
31. OpenCV.org [Internet]. Santa clara: Intel Corporation; c2020 [updated 2020 Jan 28; cited 2020 Jun 12]. Available from https://opencv.org/.
32. Andrew AM. Multiple view geometry in computer vision. *Kybernetes* 2001;**30**(9/10):1333–41.
33. Rusinkiewicz S, Levoy M. Efficient variants of the ICP algorithm. *Proc Int Conf 3-D Digit Imaging Model 3DIM*; 2001 May 28-Jun 1; Quebec, Canada. New York: IEEE; 2001. p. 145-52.
34. Rezatofighi H, Tsoi N, Gwak J, et al. Generalized intersection over union: a metric and a loss for bounding box regression. *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*; 2019 Jun 16-20; Long Beach, CA. 2019. New York: IEEE; 2020. p. 658-66.
35. Azuma RT. A survey of augmented reality. *Presence Teleoperators Virtual Environ* 1997;355–85.
36. Frigo MA, da Silva ECC, Barbosa GF. Augmented reality in aerospace manufacturing: a review. *J Ind Intell Inf* 2016;**4**(2):125–30.
37. Safi M, Chung J, Pradhan P. Review of augmented reality in aerospace industry. *Aircr Eng Aerosp Technol* 2019;**91**:1187–94.
38. Cardoso LFD, Mariano FCMQ, Zorzal ER. A survey of industrial augmented reality. *Comput Ind Eng* 2020;**139** 106159.
39. Wikitude GmbH. [Internet]. Salzburg: Wikitude GmbH.; c2020 [updated 2020 May 19; cited 2020 Jun 12]. Available from: https://www.wikitude.com/.
40. PTC Inc. [Internet]. Boston: PTC Inc.; c2011-2020 [updated 2020 May 6; cited 2020 Jun 12]. Available from: https://library.vuforia.com/content/vuforia-library/en/features/objects/model-targets.html.
41. Russell BC, Torralba A, Murphy KP, et al. LabelMe: a database and web-based tool for image annotation. *Int J Comput Vis* 2008;**77**(1–3):157–73.
42. Powers D. Evaluation: from precision, recall and f-measure to ROC, informedness, markedness & correlation. *Mach Learn Technol* 2011;**2**(1):37–67.
43. Zhang WX, Lin S, Hassani BF, et al. CloudAR: a cloud-based framework for mobile augmented reality; *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*; Mountain View, California. 2017. New York: ACM. p. 194-200.
44. Unity Technologies [Internet]. San Francisco: Unity Technologies; c2020 [updated 2020 Mar 10; cited 2020 Jun 12]. Available from https://unity.com/.

Please cite this article in press as: ZHAO G et al. A mask R-CNN based method for inspecting cable brackets in aircraft, *Chin J Aeronaut* (2020), https://doi.org/10.1016/j.cja.2020.09.024