ROBOTIC FUSELAGE INSPECTION FOR DENTS AND SCRATCHES

STUDENTS: Anael Aguayo-Chong, Ray Chen, Shreeya Gadgil, Tommy Le, Hao Lin, Sanjar Normuradov

Background

- Before a fuselage is painted, an inspector needs to manually inspect the surface for defects such as dents, scratches, and unflushed fasteners. This inspection process usually takes hours, as the inspector must walk around the fuselage with a flashlight, carefully examining and identifying any defects. The fuselage inspection project aims to increase the efficiency of the current fuselage inspection process.
- Our team wants to develop a vision system that utilizes the Fanuc CRX-20iA/L and an Intel RealSense camera to capture images of defects on a fuselage panel. The images taken with RealSense camera will then be used as a dataset for a computer vision model to be classified into different defects. The classified defects and their locations will then be visualized on an Augmented Reality (AR) app.

Fig. 1: Fanuc CRX 20iA/L

Requirements

Robotic Arm:

- Develop a scanning motion path using ROBOGUIDE for the robotic arm to traverse the fuselage panel surface
- Capture images of defects present on the fuselage panel using an Intel Realsense camera.
- Develop a computer vision model based on the Inception V3 architecture to classify the captured defect images.

AR App:

- Scan the object and then export it as a 3D model
- Anchor 3D models at a specific location in the real world.
- Mark spots outside and locate those inside.
- Display defect locations on the 3D model

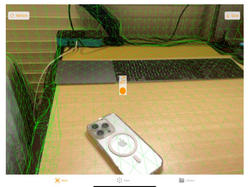

Fig. 2: Scan function of app

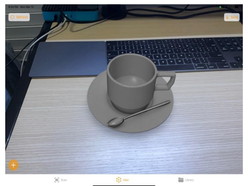

Fig. 3: View function of app

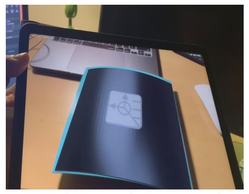

Fig. 4: Anchoring function of app

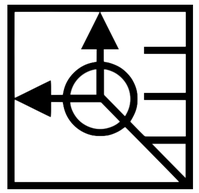

Fig. 5: Anchoring Align Sticker

Implementation & Results

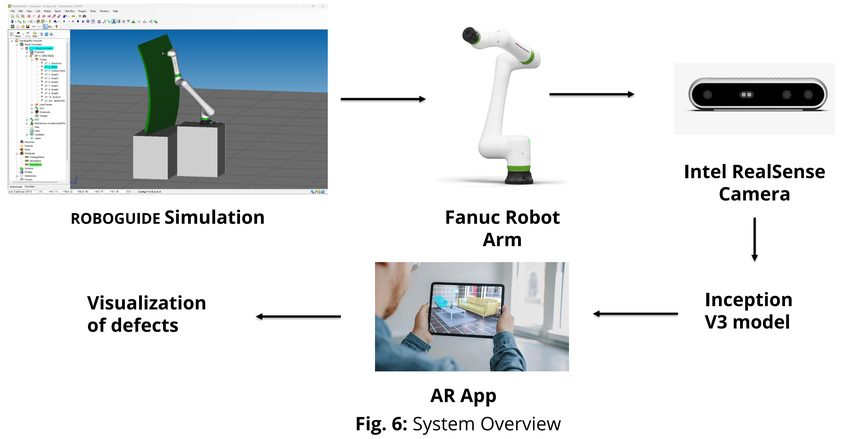

Fig. 6: System Overview

RoboGuide Simulation:
We set up a simulation environment in ROBOGUIDE. Defined the virtual representation of the fuselage panel and configure the robot arm to interact with it. This includes defining the robot arm's movements, positions, and actions required for scanning the panel.

Path Planning:
Using ROBOGUIDE, we planned the path the robot arm will follow while scanning the panel. Determined the start and end points, as well as any intermediate positions that the robot arm should visit to capture comprehensive scans.

Machine Learning:
Once the scanning and image capture process is complete, we preprocessed the images to have a size of 299 x 299 and rotated each image by 90, 180, and 270 degrees. We had a dataset of about 560 images and 1007 images after performing data augmentation on the training data. Then, we trained the Inception V3 model on 100 epochs with a learning rate of 0.001 and achieved an accuracy of 98.5 %.

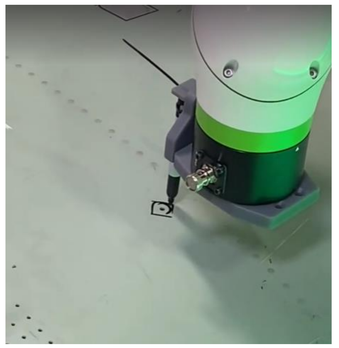

Fig. 7: Robot marking test defect

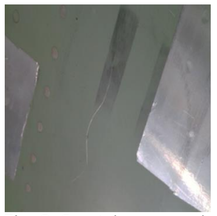

Fig. 8: Scratch on panel

Fig. 9: No scratch on panel

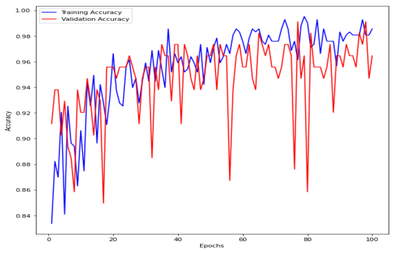

Fig. 10: Training vs Validation Accuracy

Object Scanning and 3D model conversion:
To make this function, we implemented Apple's ARKit and the mesh to be able to scan and generate a 3D model. ARKit provides tools and APIs for object scanning, allowing you to capture the geometry and texture of real-world objects using the device's camera. The scanned data would be processed to create a 3D model that accurately represents the physical object.

Image Detection and Anchoring Alignment Sticker:
To anchor the 3D model to the real-world object, the app would utilize image detection techniques. You would need to create an anchoring alignment sticker or marker that can be placed on the physical object. This sticker would serve as a visual cue for the app to detect and recognize the object's position and orientation accurately.

Drawing Perpendicular Lines to Indicate Defects:
Once the 3D model is correctly aligned with the physical object using the anchoring alignment sticker, the app can proceed to identify and mark defects. To draw perpendicular lines at specified spots, we determine the locations of the defects on the 3D model based on the locations given from the robot arm and camera. Based on the defect data or predefined coordinates, the app can calculate the appropriate positions and orientations for the perpendicular lines.

Conclusion

The objective of our project was to enhance the efficiency of the current fuselage inspection process.

- To achieve this, our team worked on the development of a vision system using the Fanuc CRX-20iA/L robot arm and an Intel RealSense camera. This system captures images of defects on a fuselage panel, which are then used as a dataset for a computer vision model.
- The computer vision model analyzes the captured images and classifies the defects into different categories. The classified defects and their locations are then visualized on an AR application. This approach allows inspectors to identify and locate defects without the need for manual inspection quickly and accurately.
- The combination of the robot arm, vision system, and AR app enables automated defect detection and visualization, providing a more efficient and effective approach to fuselage inspection.

Future Work, References, and Acknowledgments

Future Goals:

- Development of the motion path further to span the entire body of a fuselage
- Incorporating the Keyence LJX-8400 laser profiler into our vision system
- Build a larger dataset that consists of more surface defects
- Data transfer between robot arm and AR app
- Implementing found defects on the AR app

Acknowledgements:

- We would like to thank Gina Kim of Universal Robots and Dana Lopez of Fanuc America for their guidance and help on programming and using the robot arms
- We would also like to thank our mentors, Serah Stephens, Mike Woogerd, and John Raiti for their guidance and help throughout the project.

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

ADVISERS: Mike Woogerd, Serah Stephens, John Raiti

SPONSOR: Airbus Robotics