International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 12 | Dec - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Crack Detection on the Outer Body of the Aircraft

Usha Srujana R[1], Vandana A S[2], Vedhasri K.P[3], Yashpal Singh[4], Zeba Muzammil[5], Dr.Nirmala S[6]
[1,2,3,4,5]Student, Dept. of Computer Science and Engineering, AMC Engineering College, Bengaluru, India
[6]Professor, Dept. of Computer Science and Engineering, AMC Engineering College, Bengaluru, India

ABSTRACT-The detection of crack in an aircraft wing is very important for the safety and maintenance of the aircraft. Traditional methods like visual inspection and manual analysis leads to human error and is very time consuming. We are using Python and its strong libraries like OpenCV, OS, and NumPy to explore this study. Image processing helps us to identify the cracks present on the aircraft wing surface by analysing the images captured using cameras. This process includes Image acquisition that gives high-resolution images and then stored and managed using OS library to manage the files of the images. Crack detection, this is extracted using OpenCv it highlights the region containing cracks. The detected cracks are compared to the original uncracked surface and is intimated later on. Here, we use CNN algorithm to improve the performance of crack detection, compared to the traditional ways. CNN is used to detect complex and simple cracks as well present in the aircraft wings. This enables for a faster inspection, signifying the safety and assurance of the aircraft.

Keywords: OpenCV, OS library, Crack detection, Convolutional Neural Network algorithm, high resolution images

1. INTRODUCTION

Aircraft wing crack detection is an essential part of aviation safety, vouching for structural soundness of the aircraft throughout its operational life. Wings are exposed to high stress and vibrations that can cause crack formation or other type of damage. Such cracks, if unobserved, may degrade the aircraft performance and threaten aircraft's safety greatly.

Advanced crack detection approaches such as visual inspection and so on, and more recently, machine learning and artificial intelligence-dependent methods have been implemented to detect and evaluate the presence and locations of cracks in the wings of an aircraft. These methods are developed to increase the accuracy, robustness, and efficiency of inspections, in the end prolonging service life of
aircraft and maintaining the safety of passengers and crew. Accurate crack detection is not only useful in preventing catastrophic failures but also cost saving solutions because of early detection and repairs in focal areas.

This area is still in development through novel sensor technologies, data analysis, and automation, enabling high precision, non-invasive, and punctual crack detection.

Wing of aircraft is an important structural element which can bear the lift and holding the stable condition and safety of aircraft in flight. Due to the high workloads and stresses of the wings, the relatively frequent exposure of the wings to environmental factors such as temperature changes, humidity and corrosion, and the high workload on the wings, monitoring the wing condition is crucial for airworthiness.

Especially wing structure, cracks are of serious safety concern. Minor concealed cracks that develop can extend in time with fatigue, vibration and other service stresses, which can eventually cause catastrophic structural collapse. For that reason, early and precise detection of cracks is of great importance for the integrity of the aircraft and for the safety of passengers and crew.

The wing of an aircraft is a critical structural component that can support lift and maintain, as well as safe condition during flight. Because of the very high workloads and stresses of the wings, the relatively frequent exposures of the wings to environmental stresses (e.g., temperature variations, humidity, corrosion) and the relevant high workloads of the wings, assessment of wing condition is important for airworthiness.

Especially wing structure, cracks are of serious safety concern. Cracks that develop may be minor and are

developed over time in response to fatigue, vibration and other service stresses that may lead to catastrophic structural failure.

1.1. OBJECTIVES

- Enhance Safety: Detecting the cracks before it become essential ensures the protection of the passengers, group and Cargo.
- Extending Aircraft Life: Timely maintenance primarily based on cracks detection helps to lessen wear and tear and can lengthen the operational lifestyles of the aircraft
- Cost Efficiency: Early detection can reduce the cost of repair.
- Time Efficiency: it takes 10000 hours of practice for a person to train him for the inspections of the aircraft which can be minimized.

1.2. PROBLEM STATEMENT

- Detection of crack in the outer structure of aircraft.
- A crack or a defect can lead to massive problems or a disaster.
- Due to stubborn crack found in aircraft wings this would lead to harmful situations like

1.3. PROPOSED SOLUTION

- Detection of crack in the outer structure of aircraft : we use AI where we will be using algorithms like CNN and Image Net.
- A crack or a defect can lead to massive problems or a disaster: The algorithm CNN is used for feature extraction, which will detect the crack or a dent.
- ImageNet is a tool, which is used for image processing.

2. LITERATURE SURVEY

2.1. Concrete bridge crack detection by image processing technique by using the improved OTSU method. SCIENCE DIRECT 2023.

Crack detection is an essential for observing the structural health and guaranteeing its safety despite its time-consuming factor. The examiners find it difficult to identify the depth and the exact magnitude of the crack available in concrete bridge by traditional methods. This method identifies and analyzes the crack in a smart way. OTSU method is the grey intensity adjustment model. Its objective is to decrease the failure in identifying cracks. It uses two main methodologies which are otsu method and sobel's filter. Otsu method defines the scope of thresholds, and the sobel's filter is used for detecting cracks on the edges . Its strategy is for various design investigation calculations, yielding the best results in digital images. Combination of otsu method and gray level discrimination methods is successful as it detects any objects in the pictures . They are using AK-TCN model to achieve this.

2.2. Crack detection using image processing techniques for radiography inspection of aircraft wing spar. Research Gate 2011.

Crack growth, if left undetected, can lead to catastrophic failure. The inspection of aging aircraft for fastener hole cracks has been a time-consuming but essential task for the military services and commercial airlines alike.

It is important to discover the cracks before it causes further damage, aircraft operators have updated their inspection techniques. While visual inspection is an essential for the detection process, many cracks in their initiation are too small to see visually, may be hidden beneath the head of the rivet, or may initiate in the inner layer of the lap joint and cannot be seen outside.

To detect these small and hidden cracks, non-destructive inspection (NDI) methods are used.

Deep Learning models, especially CNN's, are effective for identifying cracks in images by learning to recognize various damage patterns.

2.3. Using CNN to automate aircraft maintenance visual inspection-Aerospace 2020.

To detect multiple objectives in an image MASK-RCNN was chosen. It aims at detecting dents, cracks and many more defects. This project would reduce the burden of mankind as it takes 10000 hrs of practice to become an inspector of the aircraft defects. The main process of this is to convert image data into an information data.

MASK-RCNN is an algorithm, which can identify multiple object classes in a single image. Image augmentation is a technique which aims in generating new image from already existing one by resizing, flipping, cropping and improve it.

MASK-RCNN model may predict some dents even if the image has no dents. This would lead to false predictions. To avoid mispredictions on image without dents they use another classifier.

2.4. Yolov3-Lite: A lightweight crack detection network for aircraft structure based on depth wise separable convolutions.

Many aircraft accidents occur due to cracks in various aircraft structures and crack detection helps in identifying these cracks beforehand. To solve these problems YOLOv3-Lite method is used. This involves use of depth wise separable convolutions to build the core network. YOLOv3-Lite is a faster method to identify the crack, which adopts pyramid network of high and low resolution for bounding box regression.

2.5. Crack detection on aircraft composite structures using faster R-CNN.

In aviation, to inspect the internal and the external parts of the aircraft regularly in-order to guarantee the safety of the passengers during the flight. Faster R-CNN is a deep learning-based framework recently employed for object detection task widely. Faster R-CNN is a technique for crack detection because Faster R-CNN is characterized by excellent object detection accuracies, overcoming even that of highly complex structures, e.g., cracks in complex structures.

It is effective on big sets of data that are important as cracks can happen in different shapes and positions in aircraft structures. The use of Faster R-CNN for finding cracks in aircraft composite structures, which is a highly potent strategy to improve safety and cost-effectiveness.

3. SYSTEM DESIGN AND ARCHITECTURE

3.1. Flow Chart

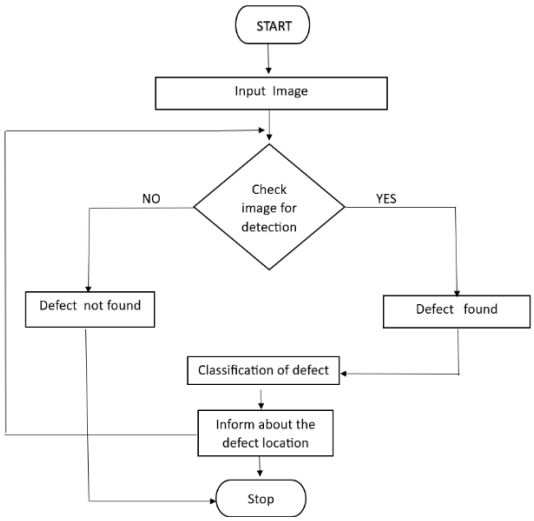

Fig.3.1 Flow Chart of crack detection in Aircraft

1. Start: The process begins with initializing the defect detection system.

2. Input Image: An image is provided to the system for analysis. It could be a photograph or a copy of a wing of airplane.
3. Check Image for Detection: Thus, the system can identify whether or not a defect exists in the input image.
4. Defect Not Found (If NO): However, assuming that the defect cannot be identified on the image, the further analysis will be terminated, and the output will be set as "Defect Not Found. The loop repeats to accommodate a new input image.

5. Defect Found (If YES): Since the defects are within an image, the processing is based to the following stage.
6. Classification of Defect: The system classifies the type of defect detected. For example, it is possible to distinguish cracks, denting and scratching.
7. Inform about the Defect Location: In the entire paradigm, the user's position is determined to the exact position of the defect, and information that is relevant has been provided to the users.
8. Stop: The workflow is complete either after defect classification and reporting is finalized or if no defect is found. The system can be "stopped" or "started" from the cycle, as well as to extend the cycle with the new input in the system.

4. WORKING

4.1. Module 1: Data Collection and Pre-processing.

Image data must be pre-processed and calibrated for CNN model

Dataset Collection: Gathering of all the labelled image of aircraft, which include all the example of crack and non-crack regions on the surface.

Image Processing: OpenCV is used for the operations like resizing, normalization and augmentation such as rotation, flipping and adjustment. Converts images to Grayscale or RGB, which serves as an input for CNN.

File Management: Use OS for a systematic organization of the dataset into training, and validation, and testing sets.

Data Preparation: Convert image files into arrays using NumPy for effective processing and submit to CNN.

4.2. Module 2: Model Development and Training.

The architecture and training of CNN for crack detection.

CNN Architecture Design: Develop a deep neural setup using libraries such as TensorFlow or PyTorch.

Training the Model: Feed pre-processed image features to the CNN for supervised learning. Use techniques like data augmentation to improve model generalization.

4.3. Module 3: Deployment and Crack Detections.

The trained CNN is implemented for the real world crack detection and evaluate its performance.

Real-Time Detection: To use the trained CNN to analyze new aircraft surface images. Leverage OpenCV for real-time image capture and pre-processing.

Integration: Providing a user-friendly interface for the operator to upload images and to check the detection results.

Evaluation and Validation: In order to ensure the system is robust, it is tested on multiple datasets. Verify performance in various scenarios and measure detection accuracy

Scalability and Automation: To manage large data, the system has been designed to allow the operating system and data files to be read/written by the system with NumPy, respectively.

5. IMPLEMENTATIONS

5.1. Language used for implementation

The language in the project is Python.
The library used are:

5.1.1. Open CV:

A highly efficient library (computer vision) for image modification and image processing.
Open CV is used for:

- Image enhancement: Thresholding.
- Pillow (PIL): Image resizing etc.

5.1.2. Deep learning Libraries:

Convolutional Neural Network (CNNs) have a wide application in wide application in image classification and object detection.

5.2. Procedures to run the project:

- Importing necessary Libraries.
- Working directory to the path
- Define a function to load images and labels from a folder and append the image.
- Rotate the image by 90-degree, 180 degree and 270 degrees.
- Append all four images to the list.
- Define a function to load images and labels from a folder and append images with 180-degree rotation.
- Looking into the folder and all of its subfolders for images.
- Rotate the images by 180 degrees.
- Append images to the list.
- Set the path to the folder containing the images around beaumount.
- Defining image size
- Load the images and labels from the folders
- Combine the images and labels and shuffle them
- Split the data into training and testing sets by shuffling the order.
- Convert the data to NumPy.
- Display some of the images from both classes.
- Initialize the model.
- First layer is a convolutional layer
- Fitting the model
- Checking the model with sample dataset and evaluate.
- Plotting the ROC Curve.
- Computing the false rate, true positive rate and threshold values and compute the area under the ROC curve,
- Plot the ROC curve.
- Predicted labels on training set.
- Compute the classification matrix for training data and test set.
- Print confusion matrices and define it.
- Plot the confusion matrix for training set.
- Set the path to the folder containing the test images, load the test image, and resize them to same size used for training.
- Convert the test images to NumPy and Set the path to folder containing the test images.

- Show the test images and predicted class probabilities.

5.3. Pseudo code

Step 1: Start
Setup Environment
Install libraries using the command: Copy
pip install NumPy pandas matplotlib scikit-learn TensorFlow keras OpenCV-python

Step 2: Prepare Dataset
Define paths:
crack_dir = 'path/to/crack/images'
non_crack_dir = 'path/to/non-crack/images'
Load images from a folder:
Function load_images_from_folder(folder):
Initialize images = []
For each filename in folder:
Read the image img = cv2.imread(folder/filename)
If img is not None: - If img is not None:
Resize to 128x128.
Append img to images.
Return images.
Load and preprocess images:
Call load_images_from_folder(crack_dir) → crack_images
Call load_images_from_folder(non_crack_dir) → non_crack_images
Label images:
crack_labels = [1] * len(crack_images)
non_crack_labels = [0] * len(non_crack_images)
Combine and normalize data:
Combine images: X = np.array(crack_images + non_crack_images)
Combine labels: y = np.array(crack_labels + non_crack_labels)
Normalize X: X = X / 255.0
Split dataset:
train_test_split is applied to the pair X and y in order to divide each into training and testing sets [X_train, X_test, y_train, y_test].
Convert labels to one-hot encoding:
y_train = to_categorical(y_train, 2)

Step 3: Build the CNN Model
Initialize model:
model = Sequential()
Add layers:
Introduce a Conv2D layer having 64 filters, 3x3 kernel size,and ReLU activation.
Add another MaxPooling2D function using the pool size 2x2.
Add a reshaping layer to reshape the features into 1D format.
Add a fully connected layer with 128 nodes, ReLU activation and 50% dropout.
Introduce an Dense output layer which has 2 units and softmax activation.
Compile model:
Optimizer: Adam.
Loss: Categorical crossentropy.
Metrics: Accuracy.
Print model summary:
model.summary()

Step 4: Train the Model
Train the model using:
X_train, y_train
Hyperparameters:
Epochs = 20
Batch size = 32
Save the training history.

Step 5: Evaluate the Model
Plot performance metrics:
Plot training and validation accuracy over epochs.
Evaluate test data:
Compute test loss and accuracy: test_loss, test_acc = model.evaluate(X_test, y_test)
Print test accuracy.

Step 6: Make Predictions
Define a prediction function:
Function predict_image(image_path, model):

Load the image from image_path.
Resize to 128x128.
Normalize and expand dimensions for model input.
Use model.predict() to get prediction.
Return the predicted class (0 or 1).
Test with a new image:
Define image_path = 'path/to/new/image.jpg'

Call predict_image(image_path, model) → result
Print Prediction: 'Crack' if result == 1 else 'No Crack'

Step 7: End

6. TESTING

Table 1- Test case table

Threshold	True Positive Rate (TPR)	False Positive Rate(FPR)	True Positives(TP)	False Positives(FP)	True Negatives (TN)	False Negatives(FN)
1.00	1.00	0.00	100	0	100	0
0.80	0.95	0.05	95	5	95	5
0.60	0.90	0.10	90	10	90	10
0.40	0.85	0.15	85	15	85	15
0.20	0.75	0.25	75	25	75	25
0.00	0.00	1.00	0	100	0	100

X-Axis (False Positive Rate - FPR):
It describes the percentage of falsely positive instances of negatives.
Calculated as
FPR= FP / FP+TN
At FPR 0.00, false negatives are missed, whereas at FPR 1.00, false negatives are all flagged.
Y-Axis (True Positive Rate - TPR):
Measures the fraction of positive cases correctly classified.
Calculated as
TPR=TP / TP + FN
At TPR 1.00, all the positive samples are accurately labeled, but, at TPR 0.00, none of the confirmed cases are rightly classified.

Threshold:
Indicates the decision boundary for classification. A higher threshold yields us fewer positives thus simultaneously improving precision (rate of true positives over total positives) and reducing (rescaling down) recall (rate of identified positives over total negatives).

Model Performance:
At threshold 1.00, the model is correct to classify all positive and negative samples (TPR 1.00, FPR 0.00).By reducing the threshold, plotting TPR deteriorates only slightly while FPR increases, describing the inherent trade-off between sensitivity and specificity. At threshold 0.00, the model always assigns a positive label to each item, i.e., FPR 1.00, TPR 0.00.
AUC Interpretation:
AUC 1.0 stands for a true positive model, TPR 1.0, FPR 0.0. This is equal to the measured ROC curve, with the orange dashed line at the top left corner.

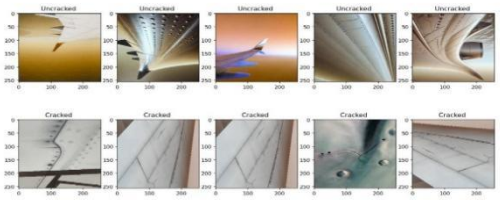

Fig.1. Classified Images

Top Row (Uncracked Images):
These images represent surfaces labeled as "Uncracked." Presented surfaces have no visible cracks or defects. These serve as the negative class (no defect).
Bottom Row (Cracked Images):
These images show surfaces labeled as "Cracked." Visible cracks or fractures are apparent in the surfaces. These serve as the positive class (defect present).

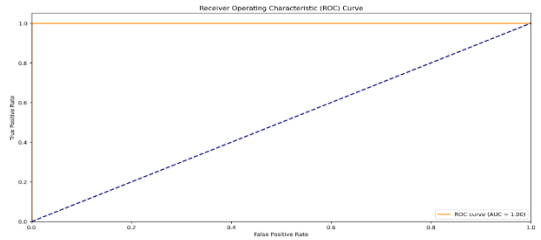

Fig.2. ROC Curve

AUC 1.00 indicates the classifier achieves 100% accuracy in predicting positive and negative classes, there is no overlap at all. The dashed diagonal line indicates the performance of a random classifier (AUC 0.5). Your model significantly outperforms this baseline.

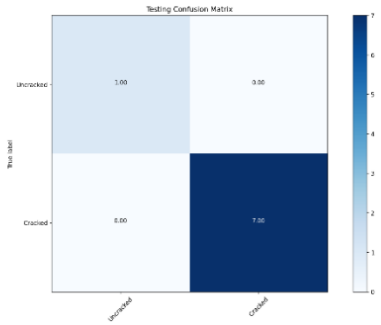

Fig.3. Confusion Matrix

Across all examples, predictions were correct without any errors. Both the "Uncracked" and "Cracked" categories have perfect scores, which indicates that error in the model is not present in this data set. In all cases, accuracy and F1-score tended to be approximately 1 (i.e., 100% accuracy as well).

7. CONCLUSION AND FUTURE WORK

7.1. Conclusion

In our project, it is possible to show effectively the uses within deep learning techniques, especially CNN for the detection of unattended cracks on the outer surface of an aircraft. By using image processing and feature extraction techniques, the system makes a significant step towards a more efficient and less manual inspection overcoming the bottleneck of access, illumination and human error. Simultaneously with this invention, it is now possible to achieve more accurate and reliable detection of structural imperfections that, therefore, imply more demanding safety requirements in the aviation field.

7.2. Future Work

Here are potential future works for the project:

1. Multi-Type Defect Detection: In order that the overgeneralization may be applied to any irregularities of the structural forms, i.e., not accounting for cracks, i.e., corrosion, dents, deformations, and to the multiclass classification framework.
2. Acknowledging the algorithm to be functional in dim light, high noisy environments, and when shown within its textured background.
3. Generalization Across Aircraft Models: It is trained and then validated by using a range of different, diverse datasets covering different scale of aircraft classes, different type of aircraft fuselage material, and different operational conditions and environmental conditions under which aircraft fly, so that it gets robustness and generalization.
4. Automated Repair Suggestions: In order that repair works be compared with the crack and the crack size that is visible, i.e. the system is to be implemented in a database.
5. Cross-Domain Applications: Indirect effects are generated, e.g., during the adaptation of the YOLOv3-Lite model to other domains (bridges, pipeline, building).
6. Enhanced Data Annotation Techniques: To reduce the need for annotation, the semi-or unsupervised learning methods are proposed, and it is shown that even model accuracies can be justified.

REFERENCES

1. Lijun Jing, Syed Hesham, Haoming Shi,Hamid Saeedipour , "Crack detection on aircraft composite structures using faster R-CNN,"
2. Brian Stephen Wong, Xin Wang, Chen Ming Koh, Chen guan tui , "Crack detection using image processing techniques for radiography inspection of aircraft wing," Research Gate, 2011.
3. Yadan Li, Zhenqi Han, Haoyu Xu, Lizhuang Liu, Xiaoqiang Li,Keke Zhang, "YOLOv3-Lite: A light weight crack detection network for aircraft structure based on depth wise separable convolutions," Applied Science, 2021.
4. Anil Dogru, Soufiane Bouarfa, Ridwan Arizar and Reyhan Aydogan, "Using Convolutional Neural Networks to Automate Aircraft

Maintenance Visual Inspection," Aerospace, 2020.

5. V.Vivekanathan,R.Vignesh ,S.Vasanthaseelan, E.joel, K. Sunil Kumar, "Concrete bridge crack detection by image processing technique by using the improved OTSU method," Science Direct ,2023.