

Aircraft Image Classification on a Small-Scale Dataset using MobileNetV2 with Grad-CAM as Explainable AI

Susi Lestari ^{1*}, Mohamad Alif Dzulfiqar ^{2**}, Ahmadi Irmansyah Lubis ^{3***}, Muhammad Andi Nova ^{4**},
Zaimah ^{5****}, Mulyadi ^{6*****}

* Akuntansi Manajerial, Politeknik Negeri Batam

** Teknik Perawatan Pesawat Udara, Politeknik Negeri Batam

*** Teknik Informatika, Politeknik Negeri Batam

**** Administrasi Bisnis Terapan, Politeknik Negeri Batam

***** Peternakan, Universitas Papua

susi@polibatam.ac.id ¹, mohamadalf@polibatam.ac.id ², ahmadi@polibatam.ac.id ³, andinova@polibatam.ac.id ⁴,
zaimah@polibatam.ac.id ⁵, mulyadi.papua63@gmail.com ⁶

Article Info

Article history:

Received 2025-07-29

Revised 2025-09-03

Accepted 2025-09-20

Keyword:

*Aircraft Image Classification,
Explainable AI (XAI),
Grad-CAM,
MobileNetV2,
pre-trained Convolutional
Neural Networks (CNN).*

ABSTRACT

This study explores aircraft image classification using MobileNetV2 combined with Gradient-weighted Class Activation Mapping (Grad-CAM) for model interpretability. A dataset of 1,500 balanced images—helicopters, propeller aircraft, and jets—was split into training, validation, and testing sets with data augmentation to reduce overfitting. Transfer learning with pre-trained MobileNetV2 achieved an accuracy of 87.56%, with macro-average precision and recall of 85.76% and 87.69%. Grad-CAM visualizations confirmed that correct predictions relied on distinctive features such as rotor blades, propellers, and engines, while misclassifications often stemmed from background distractions or less discriminative areas. These findings demonstrate the potential of lightweight architectures for small-scale datasets and highlight the value of Explainable AI in validating deep learning models. The study provides a practical reference for educational contexts and offers directions for future work with larger datasets.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. INTRODUCTION

Image classification has recently become a significant research focus, particularly in the context of aircraft imagery, due to its contributions to airport security, military strategy, and air traffic management. The application of CNN-based deep learning models to remote sensing imagery has been shown to provide high aircraft detection capabilities in a variety of challenging environmental conditions [1]. Meanwhile, other studies have demonstrated that utilizing aircraft imagery datasets, such as MTARSI, and modern deep learning models can achieve aircraft type classification with an accuracy of over 90% [2]. Along with the development of artificial intelligence technology, its application in the learning process in higher education also has the potential to replace conventional methods, which tend to be theoretical, with a visualization-based approach and digital practice that is more applicable and engaging for students [3].

The application of deep learning models, particularly Convolutional Neural Network (CNN) architectures, has proven effective in understanding complex visual patterns. One leading model is MobileNetV2, which is designed to deliver optimal performance with high computational efficiency, making it ideally suited for mobile and real-time applications [4]. In the context of aircraft image classification, the use of MobileNetV2 offers significant advantages in terms of speed, accuracy, and scalability.

However, a major challenge in applying deep learning models to critical sectors is the lack of model interpretability. Complex CNN-based systems often produce decisions that are difficult for non-technical users to understand, leading to mistrust in the system's reliability [5]. To address this, Explainable Artificial Intelligence (XAI) approaches are increasingly being applied. One well-known technique is Grad-CAM (Gradient-weighted Class Activation Mapping),

which allows the visualization of important areas in an image that influence model predictions, thereby increasing transparency and user trust [6].

Previous research has noted that the integration of Explainable AI (XAI) techniques into deep learning-based classification systems not only enhances the transparency of prediction results but also supports model performance evaluation and safety compliance — especially in the highly critical aviation domain [5], [7]. Therefore, the use of an efficient CNN architecture such as MobileNetV2, when combined with a visualization method such as Grad-CAM, can produce an aircraft image classification system that is not only fast and accurate but also easy to understand.

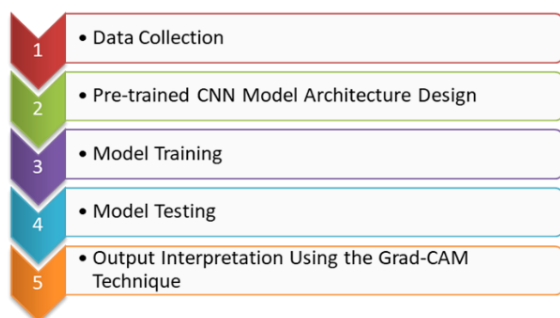
This research aims to develop an accurate and explainable aircraft image classification model using MobileNetV2 and Grad-CAM. MobileNetV2 was chosen because its architectural design prioritizes efficiency, speed, and lower memory usage, while maintaining high accuracy across a wide range of tasks [4].

Most aircraft classification studies use large datasets such as FGVC-Aircraft[8], and most also adopt heavy-duty architectures such as ResNet or EfficientNet. This study chooses the lighter MobileNetV2[4] plus Grad-CAM[6], to demonstrate an interpretive proof-of-concept in an educational scenario with a limited dataset.

The dataset used consists of 1,500 aircraft images that are proportionally balanced. The relatively small size of the dataset is indeed a limitation, but it also represents educational scenarios often encountered in machine learning in the classroom and in early research. By using the efficient MobileNetV2 and Grad-CAM as an explainable AI approach, this study contributes as an initial benchmark for image classification research in the aeronautics domain. In addition, the results of this study can be used as practical learning materials in computer vision courses, as well as serve as a basic reference for model development with larger datasets in the future.

II. METHOD

This research method consists of five main stages, namely: (1) data collection, (2) design of CNN model architecture based on pre-trained MobileNetV2, (3) model training, (4) model testing, and (5) output interpretation using the Grad-CAM technique. The flow diagram of the research stages is shown in Figure 1.



Picture 1 Research stages

A. Data Collection

Data collection was carried out on 1,500 aircraft images from various sources with predetermined aircraft type categories, namely 500 helicopter images, 500 propeller aircraft images, and 500 jet aircraft images with .jpg and .jpeg image formats. The dataset was divided into training, validation, and testing datasets in a 70:15:15 ratio. Because the data was collected from multiple sources and varying shooting angles, even though it comes from the same dataset, the test dataset can be considered a more realistic representation of unseen conditions. Table I below presents the dataset for each category, along with the distribution of the dataset used for training, validation, and testing procedures.

TABLE I
DISTRIBUTION OF TRAINING DATA, VALIDATION DATA, AND TEST DATA

No	Class	Amount of Data Trained (70%)	Amount of Validated Data (15%)	Amount of Data Tested (15%)
1	Helicopter	350	75	75
2	Propeller Aircraft	350	75	75
3	Jet Aircraft	350	75	75
Total		1,050	225	225

B. Image Processing

The following are several stages in the image processing process carried out in this study:

1) *Data Augmentation*: Overfitting occurs when a model adapts too much to the training data, thus losing its ability to generalize to new data [9]. This condition is generally caused by excessive model complexity and limited training data. Signs of overfitting include high accuracy on training data but low accuracy on validation or test data, indicating that the model is unable to capture patterns beyond its familiar data fully. Data augmentation significantly improves the generalization ability of CNNs by mimicking a broader visual distribution [10]. It has been proven that the application of data augmentation, regularization, and dropout is effective in reducing the risk of overfitting in CNN models [11]. Using each technique individually has proven quite successful in reducing overfitting. However, combining all three can lead to underfitting, a condition where the model fails to learn a sufficiently complex representation of the data [11]. To mitigate the risks of overfitting and underfitting in the pre-trained CNN model MobileNetV2, this study employs a data augmentation approach on the training data. The data augmentation used in this study utilizes the ImageDataGenerator module from Keras. The augmentation configuration used includes.

TABLE II
DATA AUGMENTATION CONFIGURATION

Configuration	Size	Functions
Rescale	1./255	Normalizes image pixel values from a range [0, 255] to [0, 1], thus speeding up the convergence process during training [12]
Rotation Range	20°	Allows random rotation of the image from -20° to $+20^\circ$ during training, to improve the model's robustness to object orientation [13]
Width and height shift range	0.2 (20%)	Shifts the image horizontally or vertically up to 20% of its original dimensions, simulating changes in the object's position within the frame. Its main purpose is to help the model become more tolerant of object shifts.
Shear Range	0.2	A tilt transformation that mimics the effects of perspective or camera distortion, to enrich the variety of viewpoints of the model.
Zoom Range	0.2	Zoom in or out on an image within a range $\pm 20\%$, mimic changes in camera distance and object scale, thereby improving the model's ability to handle objects of different sizes [14]
Horizontal Flip	TRUE	Horizontal image flipping so that the model recognizes objects from both directions, for symmetrical objects or those recognized from different perspectives [14]
Fill Mode	nearest	Fills empty pixels with the values of their nearest neighbors, keeping artifacts from corrupting the image [14]

Although data augmentation techniques can help expand image variation, they cannot completely replace the need for large amounts of original data. Goodfellow et al. [15] emphasized that augmentation only adds artificial variation to the available data, not fundamentally new information. Therefore, with a relatively small dataset size (1,500 images), the risk of overfitting can still occur even after augmentation. To mitigate this, this study applies the lightweight MobileNetV2 architecture. Several previous studies—such as those using MobileNetV2-based models in the context of remote sensing imagery—show that this architecture is capable of providing stable performance even with limited training data [16]. Therefore, the results of this study are more appropriately positioned as a proof-of-concept rather than a final model ready for widespread implementation.

It should also be noted, because model training uses processes that contain stochastic elements, such as initial weight initialization, randomization of training data, and random data augmentation, the model's prediction results can vary slightly each time the training process is run [15]. However, this variation does not significantly impact the final

performance of the model, which still exhibits high accuracy and consistency.

2) *Model Architecture and Compilation*: The model was built using MobileNetV2 as a feature extractor, leveraging its efficient architecture pre-trained on the ImageNet dataset [4]. All layers of MobileNetV2 were frozen during the initial training phase (transfer learning) to maintain stable pre-trained weights and conserve computational resources [17]. This approach represents transfer learning based on a frozen feature extractor without fine-tuning the pre-trained convolutional layers. The model was then compiled using the Adam Optimizer with the default learning rate of Keras, which is 0.001, a categorical cross-entropy loss function, and accuracy metrics according to Keras guidelines, which are commonly used in multi-class classification. The model was trained for 10 epochs.

3) *Classification Evaluation*: In the evaluation phase, a confusion matrix was constructed to analyze the model's performance in aircraft image classification. The confusion matrix allows for the measurement of accuracy, precision, and recall, and provides deeper insight into classification errors [18].

The following is the confusion matrix format for the three aircraft classes used in this study.

TABLE III
GENERAL FORM OF CONFUSION MATRIX SIZE 3X3 [19]

		Prediction			
		A	B	C	
Actual	A	TP _A	E _{AB}	E _{AC}	N _A
	B	E _{BA}	TP _B	E _{BC}	N _B
	C	E _{CA}	E _{CB}	TP _C	N _C

TABLE IV.
CONFUSION MATRIX 2X2 FOR EVERY CLASSES

TABLE IVA. CLASS A

		Prediction	
		A	Not A
Actual	A	TP _A	FN _A = E _{AB} + E _{AC}
	Not A	FP _A =E _{BA} +E _{CA}	TN _A =TP _B +E _{BC} +E _{CB} +TP _C

TABLE IVB. CLASS B

			Prediction
		B	Not B
Actual	B	TP _B	FN _B = E _{BC} + E _{BA}
	Not B	FP _C =E _{AC} +E _{BC}	TN _B =TP _A +E _{AC} +E _{CA} +TP _C

TABLE IVC. CLASS C

			Prediction
		C	Not C
Actual	C	TP _C	FN _C = E _{CA} + E _{CB}
	Not C	FP _B =E _{AB} +E _{CB}	TN _C =TP _A +E _{AB} +E _{BA} +TP _B

The following equation can also be written in this manner.

Total data:

$$N_A + N_B + N_C = N$$

Total actual data for each class:

$$N_A = TP_A + E_{AB} + E_{AC} = TP_A + FN_A$$

$$N_B = TP_B + E_{BA} + E_{BC} = TP_B + FN_B$$

$$N_C = TP_C + E_{CA} + E_{CB} = TP_C + FN_C$$

where,

N_A: The amount of data in class A

N_B: The amount of data in class B

N_C: The amount of data in class C

TP_A: The number of objects A that are successfully recognized as A (*True Positive*)

TP_B: The number of objects B that are successfully recognized as B (*True Positive*)

TP_C: The number of objects C that are successfully recognized as C (*True Positive*)

E_{AB}: The number of objects A that are recognized as B

E_{AC}: The number of objects A that are recognized as C

E_{BA}: The number of objects B that are recognized as A

E_{BC}: The number of objects B that are recognized as C

E_{CA}: The number of objects C that are recognized as A

E_{CB}: The number of objects C that are recognized as B

N_A: The actual number of object A in the observation

N_B: The actual number of object B in the observation

N_C: The actual number of object C in the observation

N: The total amount of data

FP_A: The number of objects A that are recognized as not A (B or C) (*False Positive*)

FP_B: The number of objects B that are recognized as not B (A or C) (*False Positive*)

FP_C: The number of objects C that are recognized as not C (A or B) (*False Positive*)

FN_A: The number of objects not A that are recognized as A (*False Negative*)

FN_B: The number of objects not B that are recognized as B (*False Negative*)

FN_C: The number of objects not C that are recognized as C (*False Negative*)

TN_A: The number of objects that are not A and are recognized as other than A (*True Negative*)

TN_B: The number of objects that are not B and are recognized as other than B (*True Negative*)

TN_C: The number of objects that are not C and are recognized as other than C (*True Negative*)

The prediction performance calculations are applicable to each class are described as follows

1) *Accuracy*: Accuracy in the context of classification refers to the proportion of correctly classified samples (either as true positives or true negatives) to the total number of samples tested [20]. This metric provides a general overview of the model's ability to accurately identify classes, regardless of the type of error that occurs.

$$Accuracy = \frac{TP + TN}{N}$$

where,

TP: The number of data points in a class that are successfully predicted as that class (*True Positive*)

TN: The number of data points outside a class that are successfully predicted as not being that class (*True Negative*)

N: The total number of data points in an observation.

2) *Recall (Sensitivity)*: Recall, also known as sensitivity, is an evaluation metric that measures the proportion of positive samples that a model correctly

classifies compared to the total number of positive samples [20]. In other words, recall emphasizes the model's ability to capture as many positive cases as possible in the dataset.

$$Recall = \frac{TP}{N_+} = \frac{TP}{TP + FN}$$

where,

TP: The number of data points in a class that were successfully predicted as that class (True Positive)

N+: The total number of data points in a class that were observed

FN: The number of data points in a class that were not predicted as that class (False Negative).

3) *Precision*: Precision measures the proportion of samples predicted as positive that are positive. This value is calculated as the ratio of the number of correctly predicted positive samples to the total number of samples predicted as positive by the model [20]. Precision is essential in situations where the consequences of false positives must be minimized.

$$Precision = \frac{TP}{TP + FP}$$

where,

TP: The number of data items in a class that are successfully predicted as belonging to that class (True Positive)

FP: The number of data items that are not members of a class but are predicted as belonging to that class (False Positive).

The metrics above are used to calculate model performance for each class. For overall performance, the following equation can be used.

The accuracy value for the entire dataset within a class is:

$$Accuracy = \frac{TP_A + TP_B + TP_C}{N}$$

The calculation of model performance in the form of recall and precision for all classes as a whole can be performed using three average metrics: macro-averages, micro-averages, and weighted-averages [19]. In this study, the evaluation metric used to assess the performance of the classification model is macro-averages. The selection of this metric is based on the characteristics of the dataset used, specifically its balanced number of samples for each class. In conditions like this, macro-averages are the right choice because they provide a fair average of model performance for all classes, without prioritizing certain classes based on the amount of data.

$$Recall_{macro} = \frac{Recall_A + Recall_B + Recall_C}{3}$$

$$Precision_{macro} = \frac{Precision_A + Precision_B + Precision_C}{3}$$

C. Visualizing Decision Models with Grad-CAM

This study uses the Gradient-weighted Class Activation Mapping (Grad-CAM) method to visualize and interpret the decisions of deep learning models, specifically Convolutional Neural Network (CNN) models. Grad-CAM was chosen because it can provide interpretive insights into image regions that influence model decisions, without the need for architectural modifications or retraining.

Grad-CAM calculates the gradient of the predicted class score relative to the output of the last convolutional layer. These gradients are pooled globally to obtain the weighted contribution of each feature channel. These weights are then used to combine feature activations, producing a heatmap that shows the important parts of the image in the classification process. The basic formula for Grad-CAM is as follows:

$$L_{Grad-CAM}^c = ReLU \left(\underbrace{\sum_k \alpha_k^c A^k}_{\text{linear combination}} \right)$$

where,

α_k^c : Importance weight of the feature map k to the class target
 A^k : Activation of feature maps k in the last convolutional layer

The ReLU function is used to retain only positive values from the results of linear combinations.

In this study, Grad-CAM was implemented using the Python programming language with the help of TensorFlow, NumPy, and Matplotlib libraries. This implementation consists of two main parts: generating the heatmap and superimposing it onto the original image.

Grad-CAM was evaluated qualitatively. By comparing the areas highlighted by the heatmap to visual features in the image (such as the shape of the fuselage, wings, and engines), we gained insight into how relevant the model's focus was to the predicted labels. This also helped identify potential biases or weaknesses in the model concerning specific aircraft types.

III. RESULTS AND DISCUSSION

A. Model Training

In this study, the aircraft image testing process was conducted using the transfer learning method with a pre-trained MobileNetV2-based Convolutional Neural Network (CNN) model, aided by Google Colab software and the Python programming language. As mentioned before, the dataset of 1,500 aircraft images was divided into 3 parts, those

are training (70% or 1.050 images), validation (15% or 225 images), dan testing (15% or 225 images). The testing was carried out using 225 predetermined test data points, which were distributed into three categories: 75 images for the helicopter class, 75 images for the jet class, and 75 images for the propeller class. The classification results in the image recognition process were obtained through model training on the dataset and testing using the test data. The level of classification accuracy is greatly influenced by the quality of feature extraction from each analyzed image, which is then processed using the MobileNetV2 architecture. The predictive label is determined based on the highest probability value generated by the model. If the image is successfully recognized by the system based on the test data, it is classified as True; conversely, if the image is not recognized, it is categorized as False. The results of all tests on the test data are presented in Table V below.

TABLE V
DATA TESTING RESULTS

Testing Data	True Label	Predicted Label	TRUE/FALSE
1	Propeller	Propeller	TRUE
2	Helicopter	Helicopter	TRUE
3	Helicopter	Helicopter	TRUE
4	Jet	Jet	TRUE
5	Propeller	Propeller	TRUE
6	Helicopter	Helicopter	TRUE
7	Helicopter	Helicopter	TRUE
8	Helicopter	Helicopter	TRUE
9	Jet	Jet	TRUE
10	Jet	Jet	TRUE
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
101	Propeller	Propeller	TRUE
102	Helicopter	Helicopter	TRUE
103	Helicopter	Helicopter	TRUE
104	Helicopter	Helicopter	TRUE
105	Jet	Jet	TRUE
106	Propeller	Propeller	TRUE
107	Propeller	Jet	FALSE
108	Helicopter	Helicopter	TRUE
109	Jet	Propeller	FALSE
110	Jet	Jet	TRUE
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
216	Helicopter	Helicopter	TRUE
217	Jet	Jet	TRUE
218	Propeller	Propeller	TRUE
219	Jet	Jet	TRUE
220	Helicopter	Helicopter	TRUE

221	Jet	Jet	TRUE
222	Helicopter	Helicopter	TRUE
223	Propeller	Propeller	TRUE
224	Propeller	Propeller	TRUE
225	Helicopter	Helicopter	TRUE

Furthermore, referring to the test results presented in Table V, the classification value obtained from the dataset test through evaluation using the Confusion Matrix is presented in Table VI below.

TABLE VI
THE 3X3 CONFUSION MATRIX

		Prediction			
		Helicopter	Propeller	Jet	
Actual	Helicopter	69	6	0	75
	Propeller	3	59	13	75
	Jet	0	6	69	75

Based on the confusion matrix, most of the model's predictions align with the actual labels; however, several misclassifications are still observed. The Propeller class shows the highest misclassification rate, with 13 samples incorrectly predicted as Jet and 3 samples predicted as Helicopter. Additionally, 6 Helicopter samples were misclassified as Propeller, while 6 Jet samples were also predicted as Propeller.

These errors suggest that the model has difficulty distinguishing the visual characteristics of Propeller aircraft, particularly when compared to Jet. One possible cause is the visual similarity between small jets and certain propeller aircraft from specific angles, which can confuse the model. Another contributing factor could be the relatively small dataset size, which may limit the model's ability to generalize across diverse aircraft appearances. Furthermore, class imbalance or insufficient intra-class variation might also play a role in these misclassifications.

The following is a 2x2 confusion matrix table for each class.

TABLE VII
CONFUSION MATRIX FOR EACH CLASS

TABLE VIIA. HELICOPTER CLASS

		Prediction	
		Helicopter	Not Helicopter
Actual	Helicopter	69	6
	Not Helicopter	3	147

TABLE VII.B. PROPELLER AIRCRAFT CLASS

		Prediction	
		Propeller	Not Propeller
Actual	Propeller	59	16
	Not Propeller	12	138

TABLE VII.C. JET AIRCRAFT CLASS

		Prediction	
		Jet	Not Jet
Actual	Jet	69	6
	Not Jet	13	137

The following is the calculation of the accuracy value for the model based on the results in the confusion matrix table above.

$$\begin{aligned}
 accuracy &= \frac{TP_{Helicopter} + TP_{Propeller} + TP_{Jet}}{N} \\
 &= \frac{69 + 59 + 69}{225} \\
 &= 87.56\%
 \end{aligned}$$

Based on the accuracy score above, the model generally performed exceptionally well, with an overall accuracy value of 87.56%.

Next, the recall and precision calculations for each class and overall can be seen as follows.

1) Helicopter:

$$\begin{aligned}
 recall_{Helicopter} &= \frac{TP_{Helicopter}}{TP_{Helicopter} + FN_{Helicopter}} \\
 &= \frac{69}{69 + 6} \\
 &= 92\%
 \end{aligned}$$

This result indicates that the model successfully identified 92% of all helicopter images in the dataset. Out of 75 actual helicopters, 69 were correctly classified, while 6 were missed. A high recall value demonstrates that the model is effective at minimizing false negatives for this class.

$$\begin{aligned}
 precision_{Helicopter} &= \frac{TP_{Helicopter}}{TP_{Helicopter} + FP_{Helicopter}} \\
 &= \frac{69}{69 + 3} \\
 &= 95.83\%
 \end{aligned}$$

This means that when the model predicts an image as a helicopter, it is correct 95.83% of the time. Out of 72 predictions of helicopters, 69 were accurate and 3 were

misclassified. A high precision value shows that the model makes few false positive errors for this class.

2) Propeller:

$$\begin{aligned}
 recall_{Propeller} &= \frac{TP_{Propeller}}{TP_{Propeller} + FN_{Propeller}} \\
 &= \frac{59}{59 + 16} \\
 &= 78.67\%
 \end{aligned}$$

This indicates that the model was able to correctly identify 78.67% of actual propeller aircraft. However, 16 instances were misclassified as other classes (false negatives), showing that the model sometimes struggles to detect propeller aircraft.

$$\begin{aligned}
 precision_{Propeller} &= \frac{TP_{Propeller}}{TP_{Propeller} + FP_{Propeller}} \\
 &= \frac{59}{59 + 12} \\
 &= 83.10\%
 \end{aligned}$$

This means that when the model predicts an aircraft as a propeller type, it is correct 83.10% of the time. Out of 71 predictions of propeller aircraft, 59 were accurate while 12 were misclassified (false positives). The precision is relatively good, but lower than the helicopter class, shows that some other aircraft types were mistakenly predicted as propellers.

3) Jet:

$$\begin{aligned}
 recall_{Jet} &= \frac{TP_{Jet}}{TP_{Jet} + FN_{Jet}} \\
 &= \frac{69}{69 + 6} \\
 &= 92\%
 \end{aligned}$$

This shows that the model successfully identified 92% of actual jet aircraft. Only 6 jets were missed and misclassified into other categories (false negatives), which demonstrates that the model is effective in detecting jets.

$$\begin{aligned}
 precision_{Jet} &= \frac{TP_{Jet}}{TP_{Jet} + FP_{Jet}} \\
 &= \frac{69}{69 + 13} \\
 &= 84.15\%
 \end{aligned}$$

This means that when the model predicted an aircraft as a jet, it was correct 84.15% of the time. Out of 82 predictions, 69 were true positives while 13 were misclassified as jets (false positives).

The overall macro-average recall and precision are calculated as follows:

$$recall_{macro} = \frac{recall_{Helicopter} + recall_{Propeller} + recall_{Jet}}{3}$$

$$\begin{aligned}
 &= \frac{92\% + 78.67\% + 92\%}{3} \\
 &= 87.56\% \\
 \text{precision}_{\text{macro}} &= \frac{\text{precision}_{\text{Helicopter}} + \text{precision}_{\text{Propeller}} + \text{precision}_{\text{Jet}}}{3} \\
 &= \frac{95.83\% + 83.10\% + 84.15\%}{3} \\
 &= 87.69\%
 \end{aligned}$$

The macro-average values for precision and recall were 85.76% and 87.69%, respectively, showing the model performed exceptionally well in recognizing and classifying all three classes equally.

Overall, these results demonstrate that the image classification approach using the MobileNetV2 model with transfer learning can produce optimal performance in recognizing aircraft types from available imagery.

However, it should be noted that model performance may experience slight variations with each retraining due to the stochastic nature of the training process, including random initialization and data augmentation. Nevertheless, the model generally demonstrated stable and consistent performance throughout the evaluation period.

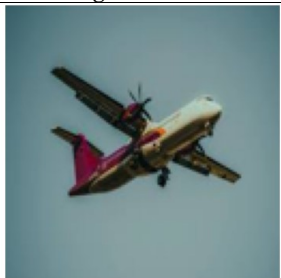




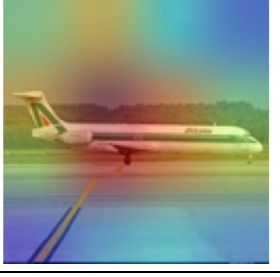
B. Implementing Grad-CAM






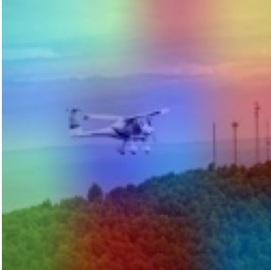
After obtaining classification results that demonstrated satisfactory performance based on accuracy, precision, and recall metrics, the next step was to interpret the model's decision-making process using the Gradient-weighted Class Activation Mapping (Grad-CAM) method. This approach aims to understand the image regions that the model deems most informative in generating predictions for each aircraft class.

The Grad-CAM visualizations generated in this study allowed researchers to evaluate the interpretability and visual validation of the model's predictions. Suppose the Grad-CAM highlighted area falls within the main object (e.g., the Propeller for the propeller class or the rotor for the helicopter). In that case, it can be concluded that the model has learned the appropriate features. Conversely, if the highlighted area falls within the background or an irrelevant element, this could indicate potential bias or noise in the model's learning.

The following figures show a comparison between the original image and the Grad-CAM activation map for three aircraft classes: helicopter, Propeller, and jet.

TABLE IX
COMPARISON OF CLASSIFIED IMAGES WITH GRAD-CAM

Prediction		Original Picture	Grad-CAM	Explanation
True	Original Label: Propeller Prediction: Propeller (0.99)			The model successfully classified the image correctly (probability 0.99). The Grad-CAM visualization indicates that the model primarily focuses on the propeller and wings, which are characteristic features of propeller-driven aircraft. High activation in these areas indicates that the model uses visually relevant features in its decision-making.
	Original label: Helicopter Prediction: Helicopter (1.00)			The model correctly classifies helicopters (probability 1.00). Grad-CAM highlights the rotors and main fuselage, indicating that the model focuses on the distinctive features that distinguish helicopters from other aircraft.
	Original label: Jet Prediction: Jet (1.00)			The model correctly classifies jet aircraft (probability 1.00). The model appears to focus on the main fuselage, particularly the wings and engines, which are key characteristics in distinguishing jet aircraft.

False	Original label: Jet Prediction: Propeller (0.75)			The model incorrectly predicted a jet aircraft as a propeller aircraft with a confidence level of 0.75. Grad-CAM results showed that the model focused more on the cockpit glass and front wings, rather than the rear engine, which is characteristic of jets, indicating the model's difficulty in recognizing distinguishing features when the images are visually similar.
	Original label: Propeller Prediction: Jet (0.98)			Grad-CAM revealed that the model primarily focused on the fuselage and wing center section, while neglecting the propeller area. This led to misclassification, as essential features of a propeller aircraft were not correctly recognized, indicating the model's limitations in capturing crucial visual elements.
	Original label: Propeller Prediction: Helicopter			The model incorrectly predicted a propeller aircraft as a helicopter with a confidence level of 0.85. Grad-CAM revealed that the model focused more on the background (forest and tower) than on the aircraft, resulting in misclassification.

The results of the Grad-CAM visualization provide further insights into the model's decision-making process. In correctly classified cases, the highlighted regions correspond to the most distinctive features of each aircraft category. For example, helicopters were primarily recognized through attention to rotor blades, while propeller aircraft predictions focused on the propeller and wing regions. Similarly, jets were correctly classified when the model emphasized the wing and engine areas. This indicates that the model relies on semantically meaningful features when producing accurate classifications.

On the other hand, misclassified cases reveal the model's limitations. In several instances, propeller aircraft were incorrectly predicted as jets due to the model overlooking the propeller region and instead focusing on the fuselage. A similar issue occurred when a propeller aircraft was misclassified as a helicopter, where attention shifted toward background objects rather than the aircraft itself. These findings align with the error patterns observed in the confusion matrix, suggesting that misclassifications often occur when attention is misdirected to non-discriminative or overlapping features.

Overall, the combination of confusion matrix analysis and Grad-CAM visualization enhances the interpretability of the model. Beyond accuracy, these results demonstrate that the model is generally capable of capturing relevant visual cues, while also highlighting areas where

classification errors are likely to emerge. This provides valuable guidance for refining the dataset and improving robustness against background interference.

IV. CONCLUSION

In conclusion, this research has effectively demonstrated the utility of a MobileNetV2-based Convolutional Neural Network (CNN) for aircraft image classification, achieving a commendable overall accuracy of 87.56%. The systematic approach employed, which included data collection, model architecture design, training, testing, and the application of Grad-CAM for interpretability, has highlighted the robustness of the model even when utilizing a relatively small dataset of 1,500 images. The findings underscore the importance of data augmentation techniques in enhancing the model's generalization capabilities, mitigating risks of overfitting, and ensuring reliable performance across different aircraft categories. Moreover, the use of Grad-CAM has provided valuable insights into the model's decision-making process, enhancing transparency and interpretability, which are critical in applications within the aviation sector. The research not only contributes to the existing body of knowledge in aircraft image recognition but also lays the groundwork for future studies that may explore larger datasets and more complex architectures. Overall, this study serves as a

significant step toward advancing explainable AI in the domain of aviation and offers practical implications for educational scenarios in machine learning.

REFERENCES

- [1] Y. Alraba'nah and M. Hiari, "Improved convolutional neural networks for aircraft type classification in remote sensing images," *IAES International Journal of Artificial Intelligence*, vol. 14, no. 2, pp. 1540–1547, Apr. 2025, doi: 10.11591/ijai.v14.i2.pp1540-1547.
- [2] H. Nur Aydin, B. Şener, Ç. Berke Erdaş, and H. Nur Aydin, "Leveraging Deep Learning for Accurate Aircraft Classification in Remote Sensing Images." [Online]. Available: <https://www.researchgate.net/publication/385015866>
- [3] M. Alif Dzulfiqar, A. Irmansyah Lubis, P. Studi Teknik Perawatan Pesawat Udara, P. Negeri Batam, and P. Studi Teknologi Rekayasa Perangkat Lunak, "Media Pembelajaran Pengenalan Citra Pesawat Udara Dengan Memanfaatkan Metode Jaringan Saraf Tiruan," 2024. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JATRA> <https://jurnal.polibatam.ac.id/index.php/JATRA>
- [4] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks."
- [5] N. M. Imam, A. Ibrahim, and M. Tiwari, "Explainable Artificial Intelligence (XAI) Techniques To Enhance Transparency In Deep Learning Models," *IOSR J Comput Eng*, vol. 26, no. 6, pp. 29–36, Dec. 2024, doi: 10.9790/0661-2606012936.
- [6] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626. doi: 10.1109/ICCV.2017.74.
- [7] S. Sutthithatip, S. Perinpanayagam, S. Aslam, and A. Wileman, "Explainable AI in Aerospace for Enhanced System Performance," in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, 2021, pp. 1–7. doi: 10.1109/DASC52595.2021.9594488.
- [8] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-Grained Visual Classification of Aircraft," Sep. 2013, doi: 10.48550/arXiv.1306.5151.
- [9] V. Diukarev and Y. Starukhin, "Proposed Methods for Preventing Overfitting in Machine Learning and Deep Learning," *Asian Journal of Research in Computer Science*, vol. 17, no. 10, pp. 85–94, 2024, doi: 10.9734/ajrcos/2024/v17i10511.
- [10] A. Hernández-García and P. König, "Further Advantages of Data Augmentation on Convolutional Neural Networks," in *Artificial Neural Networks and Machine Learning – ICANN 2018*, V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, Eds., Cham: Springer International Publishing, 2018, pp. 95–103.
- [11] Z. Xia, "Overfitting of CNN model in cifar-10: Problem and solutions," *Applied and Computational Engineering*, 2024, doi: 10.54254/2755-2721/37/20230511.
- [12] EITCA Academy, "Why is it necessary to normalize the pixel values before training the model?" Accessed: Jul. 15, 2025. [Online]. Available: <https://eitca.org/artificial-intelligence/eitca-ai-tff-tensorflow-fundamentals/tensorflow-js/using-tensorflow-to-classify-clothing-images/examination-review-using-tensorflow-to-classify-clothing-images/why-is-it-necessary-to-normalize-the-pixel-values-before-training-the-model/>
- [13] Analytics Vidhya, "Image Augmentation on the fly using Keras ImageDataGenerator." Accessed: Jul. 15, 2025. [Online]. Available: <https://www.analyticsvidhya.com/blog/2020/08/image-augmentation-on-the-fly-using-keras-imagedatagenerator/>
- [14] Keras Team, "Keras Documentation." Accessed: Jul. 22, 2025. [Online]. Available: https://keras.io/api/data_loading/image/
- [15] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning."
- [16] D. Yu, Q. Xu, H. Guo, C. Zhao, Y. Lin, and D. Li, "An efficient and lightweight convolutional neural network for remote sensing image scene classification," *Sensors (Switzerland)*, vol. 20, no. 7, Apr. 2020, doi: 10.3390/s20071999.
- [17] TensorFlow, "Transfer learning and fine-tuning." Accessed: Jul. 22, 2025. [Online]. Available: https://www.tensorflow.org/tutorials/images/transfer_learning?utm_source
- [18] H. Talebi, A. K. Bardsiri, and V. K. Bardsiri, "Developing a hybrid machine learning model for employee turnover prediction: Integrating LightGBM and genetic algorithms," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 11, no. 2, Jun. 2025, doi: 10.1016/j.joitmc.2025.100557.
- [19] M. Fahmy Amin, "Confusion Matrix in Three-class Classification Problems: A Step-by-Step Tutorial," *Journal of Engineering Research*, vol. 7, no. 1, pp. 0–0, Mar. 2023, doi: 10.21608/erjeng.2023.296718.
- [20] M. Fahmy Amin and F. Amin, "Confusion Matrix in Binary Classification Problems: A Step-by-Step Tutorial."