# Fine-Grained Aircraft Classification

Alaiba Nawaz
FAST School of Computing
National University of Computer and Emerging Sciences
Lahore, Pakistan
l215650@lhr.nu.edu.pk

Afnan Hussain
FAST School of Computing
National University of Computer and Emerging Sciences
Lahore, Pakistan
l215693@lhr.nu.edu.pk

Hammad Nazir
FAST School of Computing
National University of Computer and Emerging Sciences
Lahore, Pakistan
l218909@lhr.nu.edu.pk

## I. Introduction

The increasing availability of image data and the rapid advancements in artificial intelligence have spurred a myriad of applications in computer vision, with significant implications across diverse domains. In the aviation industry, the accurate classification of aircraft variants from images holds paramount importance for tasks ranging from maintenance scheduling to security surveillance. Motivated by the need for robust and efficient aircraft variant classification systems, this report presents a comprehensive exploration of various methodologies and techniques employed in the Fine Grained Aircraft Classification project.

## II. Motivation

The motivation behind this project stems from the critical role that accurate aircraft variant classification plays in enhancing safety, efficiency, and security within the aviation industry. Traditional methods of aircraft identification often rely on manual inspection or rudimentary automated systems, which are time-consuming and prone to errors. By harnessing the power of computer vision and machine learning techniques, we aim to develop a sophisticated classification system capable of accurately identifying aircraft variants from images with high precision and efficiency. Such a system not only streamlines maintenance operations and improves resource allocation but also enhances security measures by enabling rapid and reliable identification of aircraft in surveillance applications.

## III. Dataset Description

The dataset utilized in this project comprises 10,000 images of aircraft, meticulously labeled according to a hierarchical classification scheme. The primary focus is on the Variant level classification, with the dataset encompassing 100 different variants of aircraft. The images are organized into three sets: training, validation, and testing, with approximately 3334 images in the training set and 3333 images in each of the validation and testing sets. Each image is accompanied by its corresponding variant class name and categorical class label, ranging from 0 to 99, representing the different aircraft variants. The dataset provides a diverse representation of aircraft across manufacturers, families, and models, facilitating comprehensive model training and evaluation.

## IV. Preliminary Results and Dataset Explored

The dataset contains 10,000 images of aircraft, divided into 3334 training images, 3333 validation images, and 3333 testing images. The classification scheme is hierarchical, comprising four levels: Model, Variant, Family, and Manufacturer. However, for evaluation purposes, only the Variant level is considered due to the visual indistinguishability of certain models.

1) Model: e.g. Boeing 737-76J. Since certain models are nearly visually indistinguishable, this level is not used in the evaluation.
2) Variant: e.g. Boeing 737-700. A variant collapses all the models that are visually
3) indistinguishable into one class. The dataset comprises 100 different variants.
4) Family: e.g. Boeing 737. The dataset comprises 70 different families.
5) Manufacturer: e.g. Boeing. The dataset comprises 41 different manufacturers.

Our preliminary statistical analysis revealed that the classes are perfectly balanced with each class having 100 frequencies.

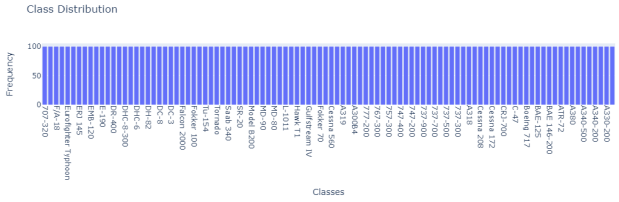


Fig. 1. Class Distribution

## V. Preprocessing Techniques

Preprocessing is crucial in computer vision tasks, ensuring data quality and reliability. In the Fine Grained Aircraft

Classification project, several preprocessing techniques were employed to enhance data quality and streamline computational processes for effective modeling. Here, we showcase these techniques applied to a random image from the training dataset to demonstrate their efficacy.

### A. Image Size Selection and Resizing

As the median size (1024, 699) from data may be too large. It might result in more complexity and longer training time. Usually, a smaller, square image size is used, such as (224, 224) or (299, 299) for many popular pre-trained models. Images were resized to (299, 299) to align with pre-trained model requirements, optimizing computational efficiency while maintaining essential details.

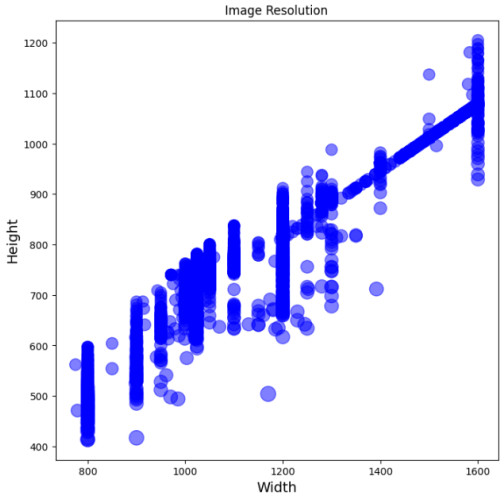


Fig. 2. Image Resolution

### B. Conversion to Gray scale

Converting images to gray scale simplified processing and highlighted key features, enhancing model discriminative power and accelerating convergence.


Fig. 3. Gray scaled Image

### C. Noise Detection and Removal

Noise was detected and removed to ensure image integrity, improving model accuracy by eliminating irrelevant artifacts.

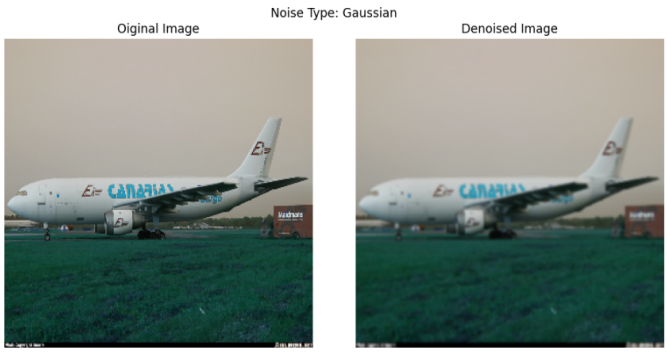


Fig. 4. Denoised Image

### D. Histogram Equalization

Histogram equalization enhanced image contrast and visibility, aiding in feature detection and improving model performance.

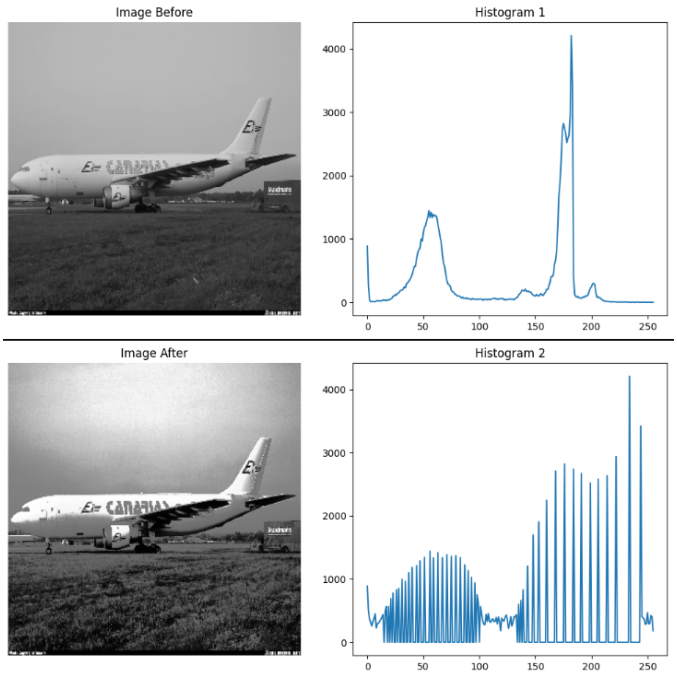


Fig. 5. Histogram Equalization

### E. Pixel Normalization

Pixel values were normalized to a common scale for stable training, facilitating smoother optimization and consistent model performance across varied inputs.

## VI. Applying Models

In this section, we delve into the application of various deep learning and machine learning models to classify aircraft variants. Each model brings unique strengths, and through rigorous experimentation, we aim to determine the most effective approach for accurate classification.

### A. Deep Learning

#### 1) CNN Trained with Preprocessed Data:

- ***Architecture:*** A CNN architecture was used with three Convolutional Layers for feature extraction (32, 64, and 128 filters), Max-Pooling Layers for downsampling, and Fully Connected Layers with 512 neurons, ReLU activation, dropout for overfitting, and a softmax output layer for multi-class classification (100 aircraft variants).
- ***Results:*** The CNN model exhibited promising results during training, achieving a high accuracy of approximately 97.5%. However, upon evaluation on the validation and test sets, the model's performance significantly declined, with test accuracy reaching only around 11.7%. This discrepancy between training and test accuracies indicated the presence of overfitting, where the model struggled to generalize to unseen data can be seen in **Fig. 5**. Further evaluation metrics, including precision, recall, and F1 score, echoed the suboptimal performance across all categories.

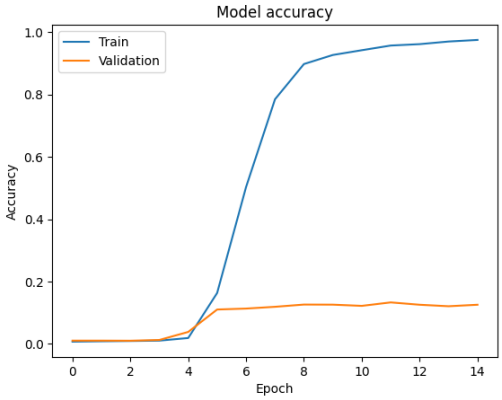


Fig. 6. Accuracy of CNN trained with Preprocessed Data

TABLE I
EVALUATION PERFORMANCE METRICS

| **Model** CNN Trained with Preprocessed Data | |
|---|---|
| **Test Accuracy** | 0.117012 |
| **Precision** | 0.130159 |
| **Recall** | 0.117032 |
| **F1 Score** | 0.116273 |

- ***Explainable AI:*** Explainable AI techniques, like Grad-CAM, were utilized to understand the CNN model's decision process. Grad-CAM generates heat maps highlighting important regions in images for classification, aiding in interpreting the model's behavior.

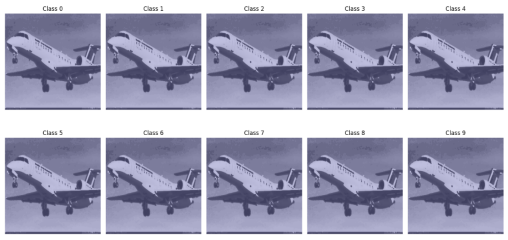


Fig. 7. CNN Preprocessed Data Explainable AI

#### 2) CNN Trained with Data Augmentation:

- ***Architecture:*** CNN was trained with data augmentation techniques to enhance model generalization and robustness. Data augmentation involves applying random transformations to training images, such as rotation, shifting, and flipping, to artificially increase the diversity of the training dataset. This helps prevent overfitting and improves the model's ability to generalize to unseen data.The Architecture remained similar to the previous model.
- ***Result:*** Despite employing data augmentation and early stopping to mitigate overfitting, the CNN model's performance showed only marginal improvement compared to the previous model. While achieving a slightly higher test accuracy of around 28.08%, the model still struggled to generalize well to unseen data. Both precision, recall, and F1 score remained low, indicating challenges in prediction consistency across different classes. The efficacy of data augmentation and early stopping in enhancing model performance was limited, highlighting the need for further optimization strategies.

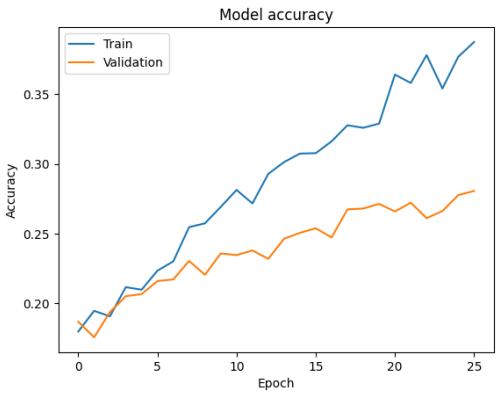


Fig. 8. Accuracy of CNN trained with Data Augmentation

TABLE II
EVALUATION PERFORMANCE METRICS

| **Model** CNN Trained with Data Augmentation | |
|---|---|
| **Test Accuracy** | 0.280828 |
| **Precision** | 0.007968 |
| **Recall** | 0.007531 |
| **F1 Score** | 0.007498 |

- ***Explainable AI:*** Utilized Grad-CAM for model interpretation revealed challenges in identifying discriminative features, suggesting the need for alternative techniques or model adjustments to improve interpretability and performance.

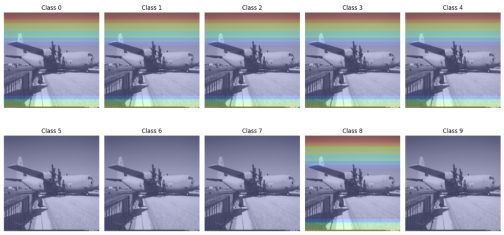


Fig. 9. CNN Data Augmentation Explainable AI

*3) Transfer Learning: ResNet-50:*

- ***Architecture:*** Leveraging transfer learning with ResNet-50, pre-trained on ImageNet, the model excluded the top fully connected layers and added a custom classification head. The architecture incorporated global average pooling, followed by a dense layer with 512 units and ReLU activation, and finally, a softmax output layer for multi-class classification across 100 aircraft variants.Data augmentation techniques during training, including rotation, width shift, and horizontal flip, enhanced model robustness and generalization.
- ***Result:*** The ResNet-50 model exhibited improved performance compared to the both previous CNN models, achieving a test accuracy of approximately 50.9%. Precision, recall, and F1 score remained low, indicating room for further optimization. Overfitting was mitigated through the use of data augmentation techniques, enhancing the model's ability to generalize to unseen data.

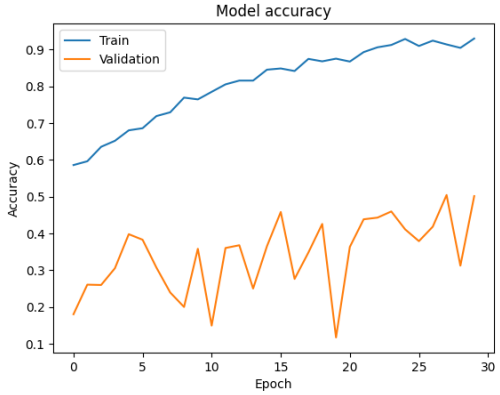


Fig. 10. Accuracy of ResNet-50

TABLE III
EVALUATION PERFORMANCE METRICS

| **Model** Transfer Learning: ResNet-50 | |
|---|---|
| **Test Accuracy** | 0.509151 |
| **Precision** | 0.009527 |
| **Recall** | 0.010205 |
| **F1 Score** | 0.009400 |

- ***Explainable AI:*** Utilizing LIME (Local Interpretable Model-Agnostic Explanations), the model's predictions were explained through visualization, providing insights into the important features influencing the classification decision.

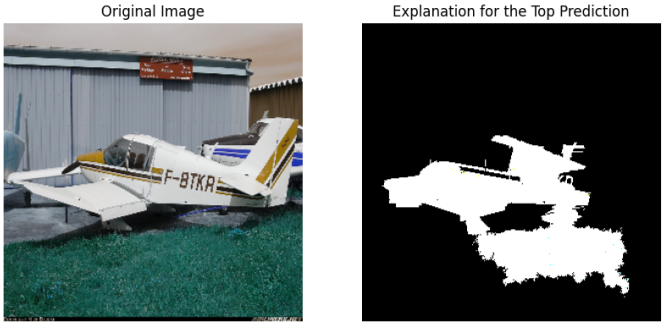


Fig. 11. ResNet-50 Explainable AI

*4) Transfer Learning: VGG19:*

- ***Architecture:*** VGG19, pretrained on ImageNet, was employed for transfer learning. Custom classification layers were added on top, including global average pooling, a dense layer with 256 units and ReLU activation, and a softmax output layer for multi-class classification across 100 aircraft variants. Data augmentation, including rotation, width shift, and horizontal flip, was applied during training to improve model robustness and generalization.
- ***Result:*** The VGG19 model exhibited poor performance, with a test accuracy of approximately 0.99%. Precision, recall, and F1 score were extremely low, indicating ineffective classification. Despite the application of data augmentation techniques during training, including rotation, width shift, and horizontal flip, the model failed to generalize well to unseen data.Comparison with previous models reveals that the VGG19 model's performance is significantly worse, highlighting its limitations in this context.

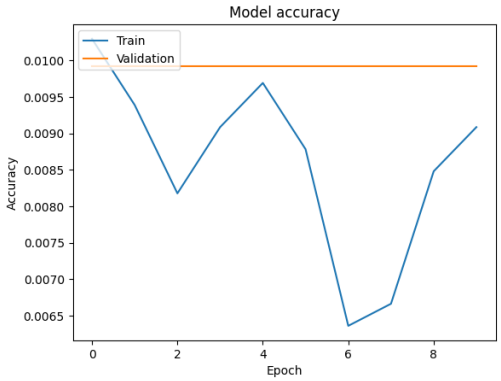


Fig. 12. Accuracy of VGG19

TABLE IV
EVALUATION PERFORMANCE METRICS

| **Model** Transfer Learning: VGG19 | |
|---|---|
| **Test Accuracy** | 0.009901 |
| **Precision** | 0.000099 |
| **Recall** | 0.010000 |
| **F1 Score** | 0.000196 |

- ***Explainable AI:*** LIME (Local Interpretable Model-Agnostic Explanations) was employed to provide insights into the model's predictions. Through visualization, LIME highlighted important features influencing the classification decision, aiding in the interpretation of the model's behavior.

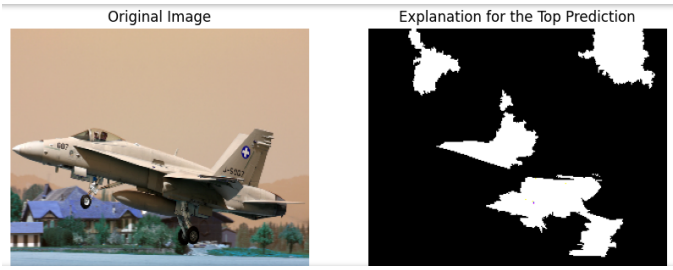


Fig. 13. VGG19 Explainable AI

*5) Multi Layer Perceptron (MLP):*

- ***Architecture:*** The MLP model consists of multiple dense layers with ReLU activation, followed by batch normalization and dropout for regularization. The input shape is (299, 299, 1), and the output layer utilizes softmax activation for multi-class classification across 100 aircraft variants. Data augmentation techniques such as rotation, width shift, and horizontal flip were applied during training to enhance model generalization.
- ***Result:*** The MLP model demonstrated limited performance, with a test accuracy of approximately 2.4%. Precision, recall, and F1 score remained low, indicating suboptimal classification performance.While data augmentation was applied during training, the model's performance did not significantly improve.. Comparatively, the MLP model's performance was notably inferior to previous models, suggesting that its architecture might not be well-suited for this task.

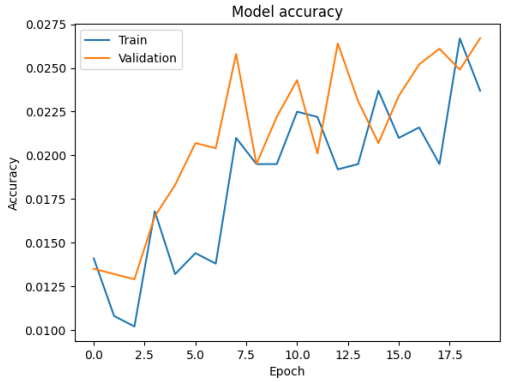


Fig. 14. Accuracy of MLP

TABLE V
EVALUATION PERFORMANCE METRICS

| **Model** MLP | |
|---|---|
| **Test Accuracy** | 0.024002 |
| **Precision** | 0.005681 |
| **Recall** | 0.011649 |
| **F1 Score** | 0.005794 |

- ***Explainable AI:*** Feature importance analysis was conducted to assess the contribution of each input feature to the model's predictions. Additionally, layer visualization techniques were employed to understand the activations at different layers of the model. However, despite these efforts, the model's performance and behavior remain subpar, indicating the need for further optimization or potentially a different model architecture.

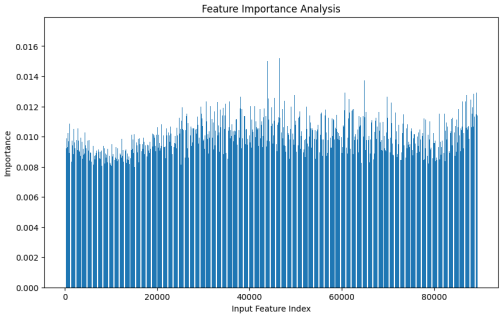


Fig. 15. MLP Explainable AI

*6) RNN:*

- ***Architecture:*** The architecture utilized two simple RNN layers with 128 and 64 units, capturing temporal dependencies. A Flatten layer reshaped the output for fully connected layers. Two Dense layers with ReLU activation (256 units and output classes) followed, concluding

with softmax activation for multi-class classification (100 aircraft variants).

- ***Result:*** The RNN model achieved a test accuracy of approximately 3.15%, significantly lower than previous models, attributed to its struggle in capturing long-term dependencies and complex patterns in sequential data. This resulted in limited generalization to unseen data. Additionally, low precision, recall, and F1 score around 0.03 indicated imprecise predictions, showcasing the model's difficulty in identifying true positives and negatives. Overall, the RNN's performance fell short due to challenges in learning and generalizing sequential patterns effectively.

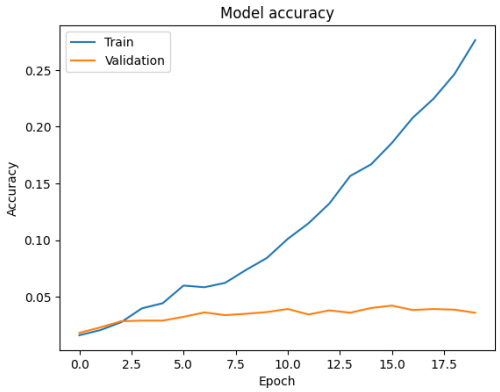


Fig. 16. Accuracy of RNN

TABLE VI
EVALUATION PERFORMANCE METRICS

| **Model** RNN | |
|---|---|
| **Test Accuracy** | 0.031503 |
| **Precision** | 0.032143 |
| **Recall** | 0.031497 |
| **F1 Score** | 0.029347 |

- ***Explainable AI:*** Feature importance analysis and layer visualization were conducted to provide insights into the model's behavior and internal representations. However, due to the nature of RNNs, interpretability may be more challenging compared to feedforward neural networks. Additional techniques such as attention mechanisms or layer-wise relevance propagation could be explored for better interpretability in RNNs.

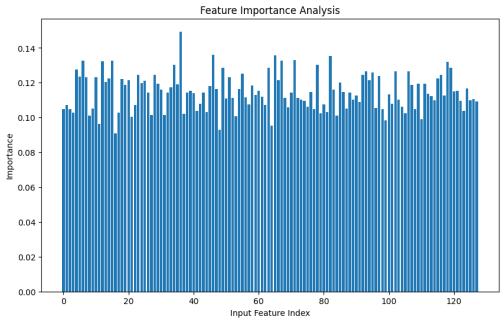


Fig. 17. RNN Explainable AI

*7) LSTM:*

- ***Architecture:*** The LSTM model comprises a single LSTM layer with 128 units, configured to return sequences, followed by a time-distributed dense layer with 64 units and ReLU activation. This is then flattened before a dense softmax output layer is applied for multi-class classification across 100 aircraft variants.
- ***Result:*** The LSTM model yielded a test accuracy of approximately 4.38%. Precision, recall, and F1 score hovered around 0.04, indicating imprecise predictions. These results underscored the challenge of capturing long-term dependencies within sequential data, leading to sub optimal classification outcomes.

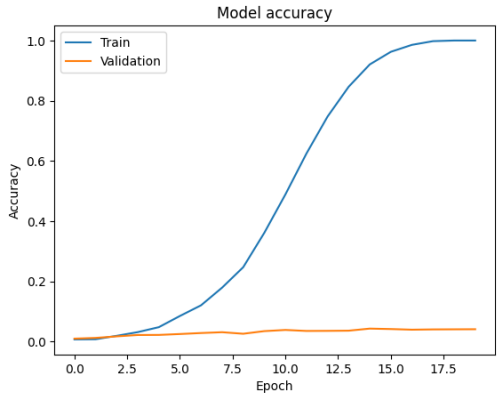


Fig. 18. Accuracy of LSTM

TABLE VII
EVALUATION PERFORMANCE METRICS

| **Model** LSTM | |
|---|---|
| **Test Accuracy** | 0.043804 |
| **Precision** | 0.044051 |
| **Recall** | 0.043824 |
| **F1 Score** | 0.043141 |

- ***Explainable AI:*** By extracting and analyzing the weights of the first dense layer in the LSTM model, we can assess the importance of different input features. The resulting feature importance analysis provides insights into which features are influential in the model's decision-making process, aiding in understanding its behavior.

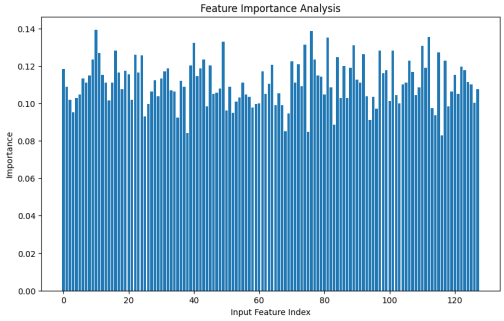


Fig. 19. LSTM Explainable AI

*8) Transformer:*

- ***Architecture:*** The Transformer model integrates a ResNet feature extractor with a TransformerEncoderLayer for sequence modeling. Leveraging transfer learning, the pre-trained ResNet-50 model, excluding its top fully connected layers, was employed. The custom classification head comprised global average pooling, followed by a dense layer with 512 units and ReLU activation, and finally, a softmax output layer for multi-class classification across 100 aircraft variants. Data augmentation techniques during training, such as rotation, width shift, and horizontal flip, were utilized to bolster model robustness and generalization.
- ***Result:*** The training process of the Transformer model over 10 epochs resulted in marginal improvement in validation loss and accuracy, though both remained very low. The model's training accuracy hovered around 0.69%, while the validation accuracy stayed at approximately 1.02%. These results indicate poor generalization and suggest that the model failed to learn meaningful patterns from the data. Furthermore, the test accuracy remained extremely low at 0.99%, affirming the model's inadequate performance in accurately classifying aircraft variants. Further analysis and optimization are necessary to enhance the model's effectiveness.

TABLE VIII
EVALUATION PERFORMANCE METRICS

| **Model** Transformer | |
|---|---|
| **Test Accuracy** | 0.009901 |
| **Precision** | NaN |
| **Recall** | NaN |
| **F1 Score** | NaN |

*9) GAN:*

- ***Architecture:*** The GAN (Generative Adversarial Network) model comprises a generator and discriminator. The generator aims to produce realistic images from random noise, while the discriminator seeks to distinguish between real and generated images. Through a series of transposed convolutional layers, the generator transforms noise into images, while the discriminator employs convolutional layers to classify images as either real or fake.

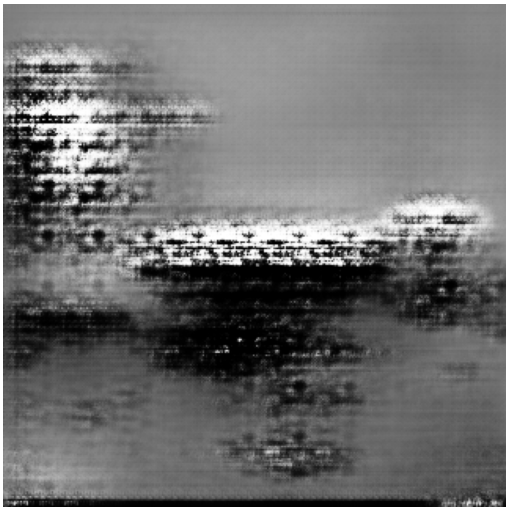

Fig. 20. Train image at epoch 1

- ***Training Process:*** During training, the model optimizes both the generator and discriminator simultaneously using gradient descent. The generator minimizes the discrepancy between generated and real images, while the discriminator maximizes this discrepancy. Training extends over 6000 epochs with a batch size of 10.Loss functions guide the training process. The generator loss is computed based on the discriminator's output when provided with generated images, while the discriminator loss evaluates its ability to discern between real and generated images.

Fig. 21. Train image at epoch 6000

- ***Result:*** Results are observed through progressively generated images captured at intervals during training. These samples provide insights into the model's learning progress and the quality of the generated images, assessing their resemblance to real counterparts and diversity.

Fig. 22. GAN generated random image

### B. Machine Learning

For training machine learning models, the dataset was meticulously prepared through stratified sampling, ensuring 100 instances per class for balanced representation. Images were uniformly resized and normalized, followed by flattening into feature vectors for streamlined processing. Labels were then encoded for supervised learning tasks. This meticulous approach aimed to optimize model performance by providing standardized input data for training and evaluation.

#### 1) SVM:

- ***Architecture:*** The SVM model, employing the radial basis function (RBF) kernel with regularization parameter C set to 1.0, was trained on the preprocessed data.
- ***Result:*** Upon validation, it achieved an accuracy of 0.036, while on the test set, it yielded a slightly lower accuracy of 0.034.

TABLE IX
EVALUATION PERFORMANCE METRICS

| **Model** SVM | |
|---|---|
| **Test Accuracy** | 0.032000 |
| **Precision** | 0.052198 |
| **Recall** | 0.032000 |
| **F1 Score** | 0.027067 |

#### 2) Logistic Regression:

- ***Architecture:*** The logistic regression model, utilizing the one-vs-rest (ovr) strategy and a maximum of 1000 iterations, was trained on the prepared dataset.
- ***Result:*** Its performance was assessed on both the validation and test sets, achieving accuracies of 0.058 and 0.061, respectively.

TABLE X
EVALUATION PERFORMANCE METRICS

| **Model** Logistic Regression | |
|---|---|
| **Test Accuracy** | 0.061000 |
| **Precision** | 0.065490 |
| **Recall** | 0.061000 |
| **F1 Score** | 0.060638 |

## VII. OVERALL RESULTS OF MODELS

### A. Accuracy

Among the models evaluated, the Transfer Learning approach with ResNet-50 emerged as the top performer, achieving the highest test accuracy of approximately 50.9%. This was followed by the CNN trained with data augmentation, which demonstrated notable improvement over the model trained with original data, attaining a test accuracy of 28.1%. In contrast, the VGG19 model lagged significantly behind with a test accuracy of only 0.99%, suggesting that its performance was inferior to the other models assessed.

Additionally, both the Multi-Layer Perceptron (MLP) and RNN models exhibited modest test accuracies of 2.4% and 3.15%, respectively, indicating limited effectiveness in classifying the data. The LSTM model performed slightly better with a test accuracy of 4.38%, while the Transformer model and SVM model showed similar low accuracies of 0.99% and 3.4%, respectively.

The Logistic Regression model performed relatively better than the SVM and Transformer models, achieving a test accuracy of 6.1%. Overall, the Transfer Learning approach with ResNet-50 emerged as the most effective method for this classification task, outperforming all other models evaluated.

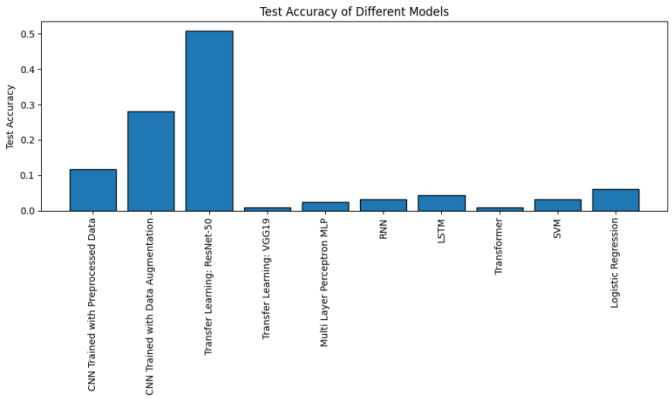


Fig. 23. Accuracy comparison of all models

### B. Precision

Precision measures the proportion of correctly identified positive cases among all cases predicted as positive by the model. In this evaluation, the models varied widely in precision. The Logistic Regression model achieved the highest precision of 0.065, indicating it correctly identified a higher proportion of positive cases. Conversely, the VGG19 model exhibited the lowest precision, barely surpassing 0.0001.

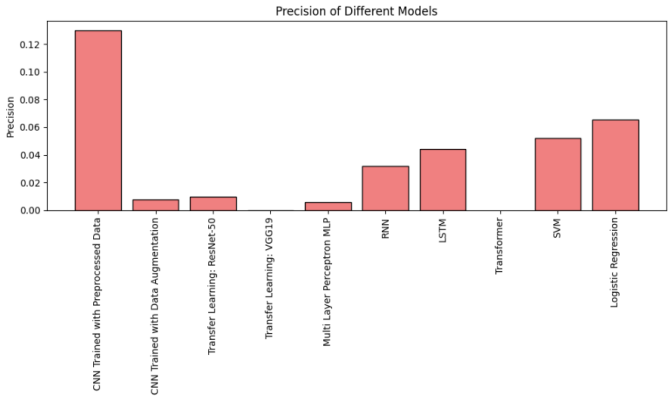


Fig. 24. Precision comparison of all models

### C. Recall

Recall, also known as sensitivity, represents the proportion of actual positive cases that were correctly identified by the model. Across the models, the LSTM model demonstrated the highest recall at 0.044, implying it effectively captured a greater portion of true positive cases. Conversely, the VGG19 and Transformer models showed the lowest recall, both barely reaching 0.010, indicating they missed a significant portion of positive cases.

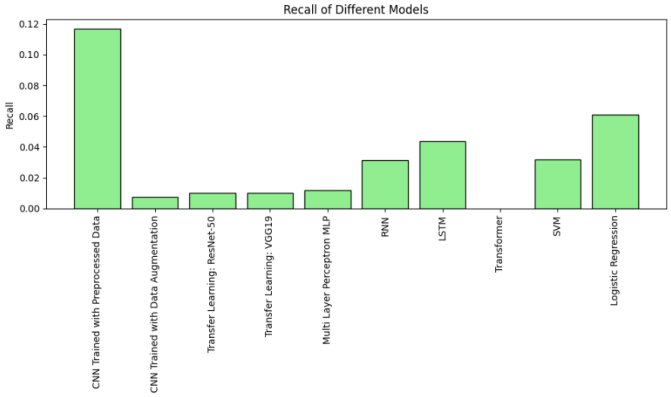


Fig. 25. Recall comparison of all models

### D. F1 score

The F1 score, a harmonic mean of precision and recall, provides a balanced assessment of a model's performance. Among the models, the Logistic Regression model achieved the highest F1 score at 0.061, suggesting a good balance between precision and recall. Conversely, the VGG19 model exhibited the lowest F1 score at 0.0002, reflecting poor performance in both precision and recall, resulting in an overall low F1 score.

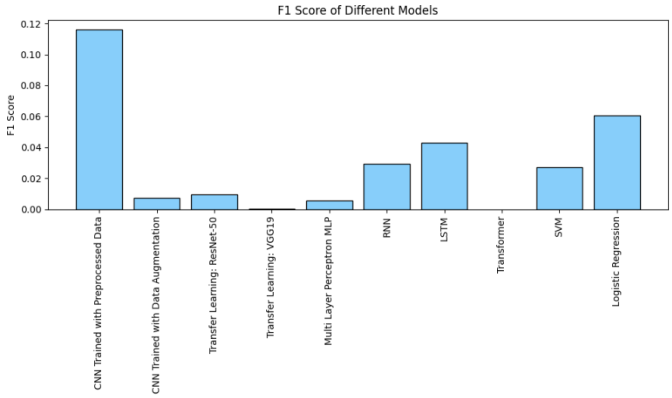


Fig. 26. F1 score comparison of all models

## VIII. Conclusion

In conclusion, while Transfer Learning with ResNet-50 emerged as the most effective model, achieving a test accuracy of approximately 50.9%, other models such as VGG19, MLP, RNN, LSTM, Transformer, SVM, and Logistic Regression demonstrated relatively lower accuracies ranging from 0.0099% to 6.1%. These models struggled to capture the intricate features and patterns within the dataset adequately. For instance, models like VGG19 and MLP may have suffered from architectural limitations in handling the complexity of the dataset, while RNNs, LSTMs, and Transformers might have faced challenges in effectively capturing temporal dependencies or long-range dependencies in the data. Moreover, SVM and Logistic Regression models, being linear classifiers, might have struggled to delineate nonlinear decision boundaries effectively, impacting their performance. Therefore, for this particular classification task, these models were not as suitable due to their inherent limitations in handling the complexity and nuances of the dataset.

## References

[1] "FGVC Aircraft", https://www.kaggle.com/datasets/seryouxblaster764/fgvc-aircraft, 2020.

[2] Aravind Ramalingam , "How to Pick the Optimal Image Size for Training Convolution Neural Network?", https://medium.com/analytics-vidhya/how-to-pick-the-optimal-image-size-for-training-convolution-neural-network-65702b880f05, 2021