Electronic Letters on Computer Vision and Image Analysis 24(1):118-133, 2025

Enhanced Bird Species Image Recognition and Classification using MobileNet and InceptionV3 Transfer learning Architectures

Sakthi Priya G[1] ,Vignesh Saravanan K[2*] and Dheetchana K[3]

[1,2,3] *Department of Computer Science and Engineering, Ramco Institute of Technology, Rajapalayam, India*

Received 20th of October, 2024; accepted 1st of May 2025

Abstract

The proposed study explores the application of transfer learning techniques in bird species image classification, specifically focusing on the MobileNet and InceptionV3 models. Using the CUB-200-2011 dataset, a widely recognized benchmark for fine-grained visual categorization, this study achieved a 74.60% accuracy with MobileNet. While larger models often report higher accuracy on this dataset, MobileNet's performance highlights the trade-off between accuracy and computational efficiency. Despite a lower accuracy compared to more complex models, MobileNet's efficient architecture makes it an ideal choice for real-world applications requiring quick deployment and low resource usage. While previous studies have established MobileNet's suitability for real-time applications due to its computational efficiency, this paper applies MobileNet to the novel domain of wildlife conservation, specifically for fine-grained bird species classification using the CUB-200-2011 dataset. The MobileNet model achieved an impressive accuracy of 74.60%, outperforming InceptionV3, which recorded an accuracy of 64.00% (CUB-200-2011), VGGNet achieved an accuracy of 86% and ResNet reported 84% on the CUB-200-2011 dataset. The corresponding loss values were 0.8685 for MobileNet and 1.128 for InceptionV3, highlighting MobileNet's superior alignment with actual class labels .Additionally, MobileNet demonstrated a precision range of 0.45 to 0.93, while InceptionV3's precision ranged from 0.65 to 0.81. The F1-scores revealed MobileNet's performance ranged from 0.40 to 0.91, in contrast to InceptionV3's lower F1-scores, indicating a more stable but less effective classification ability. These findings underscore the potential of MobileNet as a lightweight, efficient alternative for wildlife image classification tasks, making it particularly suitable for deployment in resource-constrained environments. In the proposed application, InceptionV3's complex architecture increases the risk of overfitting due to its higher parameter count and redundant feature extraction, especially on a dataset with limited samples. This leads to higher loss and lower accuracy for certain inputs. In contrast, MobileNet's lightweight design efficiently generalizes by focusing on essential features, resulting in better performance. The developed user interface allows for seamless interaction, enabling users to upload images and receive immediate classification results, further demonstrating the practical application of these models in conservation and biodiversity preservation efforts.

Key Words: Wildlife conservation, Transfer Learning, MobileNet, InceptionV3, Bird Classification, CUB-200-2011 Dataset, Precision, F1-score, Lightweight Models, Biodiversity.

Correspondence to: <vigneshsaravanank@gmail.com>

Recommended for acceptance by Angel D. Sappa
ELCVIA ISSN:1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

1 Introduction

In an era of rapid technological advancement, biodiversity preservation has become one of the most critical challenges facing our planet. The alarming decline in wildlife populations and habitats threatens ecosystem stability and undermines the numerous benefits ecosystems provide, ranging from essential services to cultural and recreational value. As traditional conservation methods struggle with scalability and efficiency, innovative technologies offer new avenues to enhance conservation efforts. Among these innovations, the application of artificial intelligence (AI) and deep learning has emerged as a transformative tool in wildlife conservation. The increasing use of camera traps, drones, and remote sensors has revolutionized data collection, generating vast amounts of wildlife imagery. However, the challenge lies in efficiently analyzing and interpreting these large datasets.

Deep learning, particularly convolutional neural networks (CNNs), has shown remarkable success in image classification tasks. However, conventional models often require significant computational resources, limiting their practicality for real-time or field-deployed applications. In this context, MobileNet, a lightweight neural network architecture optimized for mobile and edge computing, presents a promising solution. MobileNet's design prioritizes computational efficiency while maintaining high accuracy, making it ideal for wildlife conservation tasks.

This research explores the potential of MobileNet for species identification and classification in wildlife imagery. By leveraging MobileNet's strengths, to develop a scalable system capable of processing diverse datasets, including images captured in varied ecological conditions. The integration of MobileNet into conservation workflows could enhance species population monitoring, poaching detection, and timely responses to environmental changes. The paper first discusses the current challenges in wildlife conservation, particularly the difficulties in data analysis and monitoring. The introduction of AI marks a paradigm shifts in the field, offering not only improved accuracy but also real-time insights that can inform conservation strategies. Then, MobileNet's architecture and its suitability for wildlife image classification, conducting a comparative analysis with other deep learning models is examined. This analysis highlights MobileNet's advantages in computational efficiency and performance. This paper aims to contribute to the growing body of knowledge on AI's application in environmental science by demonstrating MobileNet's practical benefits for wildlife conservation. We hope to inspire further research and innovation in this critical area, ultimately advancing efforts to protect global biodiversity.

2 Related Works

Image classification has become a widely researched area in computer vision, particularly with the rise of deep learning techniques such as Convolutional Neural Networks (CNNs). These models have revolutionized image analysis by automatically extracting features from raw images, a significant improvement over traditional machine learning techniques that relied on manually engineered features [1]. CNNs, introduced by LeCun et al. [2], have since been applied to various tasks, including object detection, facial recognition, and species identification in wildlife conservation. The introduction of AlexNet by Krizhevsky et al. [3] marked a breakthrough in image classification, demonstrating the power of deep learning on large-scale datasets such as ImageNet. CNN architectures, composed of convolutional, pooling, and fully connected layers, allow for the hierarchical learning of both low- and high-level features from images. In wildlife conservation, CNNs have been utilized to classify species in large datasets such as iNaturalist, containing millions of labeled images across thousands of species [4]. Prominent models like VGGNet [5], ResNet [6], and Inception [7] have advanced the field by offering deeper architectures that learn increasingly complex features, resulting in improved species identification accuracy.

MobileNet, introduced by Howard et al. [8], is a lightweight CNN architecture designed for mobile and embedded vision applications. It achieves computational efficiency through depthwise separable convolutions, which significantly reduce the number of parameters compared to traditional CNNs like VGG and ResNet, making it well-suited for real-time field applications, such as wildlife monitoring via camera traps. While VGGNet and ResNet are known for their strong classification performance, they

were not included in the comparison as they are computationally more intensive and less suited for real-time deployment scenarios, such as wildlife monitoring. The decision to compare MobileNet against InceptionV3 is substantiated by the fact that InceptionV3 strikes a balance between model complexity and performance, making it a fitting counterpart for assessing MobileNet's capabilities in terms of both accuracy and efficiency. Studies have shown that MobileNet performs well in species classification tasks while maintaining low computational costs. For instance, Gupta et al. [9] demonstrated its effectiveness using the Caltech Camera Traps dataset, where it achieved accuracy comparable to larger models like ResNet but with reduced inference time.

InceptionV3, developed by Szegedy et al. [10], is an extension of the Inception architecture designed to enhance the efficiency of large-scale image classification tasks. It incorporates techniques such as factorized convolutions, auxiliary classifiers, and batch normalization to improve both accuracy and computational efficiency. InceptionV3's deeper architecture allows it to capture complex features, making it effective for species identification in wildlife datasets. While InceptionV3 is more computationally demanding than MobileNet, it has demonstrated superior accuracy, particularly with larger datasets. Liu et al. [11] achieved state-of-the-art performance in bird species classification using InceptionV3 on the CUB-200-2011 dataset. Although both MobileNet and InceptionV3 are widely used for image classification, they serve different purposes depending on the computational resources and accuracy requirements. MobileNet is ideal for scenarios with limited computational resources, such as real-time species identification on mobile devices or low-power systems [12]. Conversely, InceptionV3 is more suitable for tasks requiring higher accuracy, where computational overhead is less of a concern [13]. Tang et al. [14] compared the two models for wildlife classification and found that while InceptionV3 provided higher accuracy, MobileNet's faster inference times made it more practical for real-time applications.

A major advancement in deep learning is transfer learning, where a pre-trained model (often trained on large datasets like ImageNet) is fine-tuned for a specific task with a smaller dataset. This approach is particularly beneficial for wildlife classification, where labeled data is scarce [15]. Transfer learning has been shown to significantly enhance the performance of models like MobileNet and InceptionV3 when applied to specific wildlife datasets [16]. Another critical technique in wildlife image classification is data augmentation, which artificially increases the training dataset through techniques such as random rotations, flips, and zooms. This helps models learn more robust features and reduces overfitting [17]. Data augmentation is especially important in wildlife classification due to the variability in species appearance, lighting, and background conditions [18].

Despite these advancements, challenges remain in wildlife image classification, including data imbalance (where certain species are over-represented) and difficulty recognizing species in varied poses, occlusions, or environments [19]. Recent studies have explored ensemble models and attention mechanisms to address these challenges, improving accuracy in difficult conditions [20]. The applications of wildlife classification models extend beyond species identification, contributing to biodiversity monitoring, anti-poaching efforts, and habitat analysis. Automated species identification systems are increasingly deployed in protected areas to monitor wildlife populations and detect poaching in real time [21]. Through these technologies, AI is playing a crucial role in enhancing conservation efforts and preserving biodiversity. Recent studies have investigated the use of ensemble models and attention mechanisms to overcome these challenges, significantly enhancing classification accuracy in difficult conditions [22]. The applications of wildlife classification models extend far beyond species identification. These models play a crucial role in biodiversity monitoring, anti-poaching efforts, and habitat analysis. For example, automated species identification systems have been deployed in national parks to track wildlife populations and detect poaching activities in real time [23].

3 Methodology

This section outlines the comprehensive methodology employed in developing and evaluating the MobileNet-based image classification system for wildlife conservation. The process encompasses several key

stages: data collection, preprocessing, model training, and evaluation. Each phase is detailed below to provide a thorough understanding of the applied approach and techniques.

3.1 Data Collection and Processing

3.1.1 Image Data Collection

Data is sourced from wildlife camera traps, drone footage, and publicly available wildlife datasets, ensuring a diverse range of imagery for robust model training. These images are annotated with metadata such as species, location, and date, allowing for detailed analysis and facilitating model training. This initial step is critical for creating a comprehensive and balanced dataset that represents various ecosystems and species.

3.1.2 Data Cleaning

To maintain the dataset's quality, irrelevant or low-quality images—those that are blurry, overexposed, or contain irrelevant objects—are removed. This step ensures that only relevant images are used for model training. Additionally, a manual verification process is conducted to correct mislabeling and inconsistencies in the annotations, ensuring that the dataset remains accurate and reliable for the tasks of species identification and classification. In real-time field deployment, it is assumed that users will manually identify and handle bad data, ensuring the quality of images before submitting them for classification.

3.1.3 Image Normalization and Standardization

Images are resized to a standard resolution (e.g., 224x224 or 256x256 pixels) to provide uniform input to the model. This resizing helps reduce computational complexity and ensures consistency across the dataset. Color normalization is then applied to standardize brightness and contrast, reducing variability in image quality and improving model performance. These preprocessing techniques prepare the images for efficient learning by the model.

3.1.4 Data Augmentation

To increase the dataset's variability and prevent overfitting, various data augmentation techniques are applied. These include random rotations (e.g., ±30 degrees), horizontal flipping, scaling to simulate different distances, and cropping to simulate partial views of the animals. These transformations artificially expand the dataset, allowing the model to generalize better across diverse real-world conditions. Automation of these augmentations is achieved through libraries like TensorFlow's `ImageDataGenerator` or PyTorch's `transforms`.

3.1.5 Data Splitting

The dataset is divided into training (70%), validation (15%), and test (15%) subsets. The training set is used to train the model, the validation set helps in tuning hyperparameters and preventing overfitting, and the test set is reserved for evaluating the model's performance on unseen data. This splitting ensures that the model's performance is thoroughly assessed, leading to better generalization in real-world scenarios.

3.1.6 Addressing Class Imbalance

To address the class imbalance inherent in the dataset, multiple strategies were employed. These included targeted data augmentation, adjusting class weights during model training, and monitoring metrics such as F1-score and recall for underrepresented classes. These methods ensured the model's performance was not biased towards majority classes, improving its ability to classify rare species effectively. The study did not face significant class imbalance as the dataset includes 200 bird species, each with approximately 50 images, providing sufficient samples per class. The images were clear, with an initial

resolution of 500x350, which was enhanced for consistency. Data augmentation was performed to maintain uniform scaling, colors, and features, ensuring balanced input quality. The region of interest was carefully extracted, minimizing classification imbalance. For testing, 200-250 images per class were used, and all classes were classified accurately, showing minimal imbalance issues.

3.2 Model Selection and Training

3.2.1 Implementation of MobileNet

MobileNet was chosen for its efficiency and adaptability in resource-constrained environments, such as mobile and edge devices.

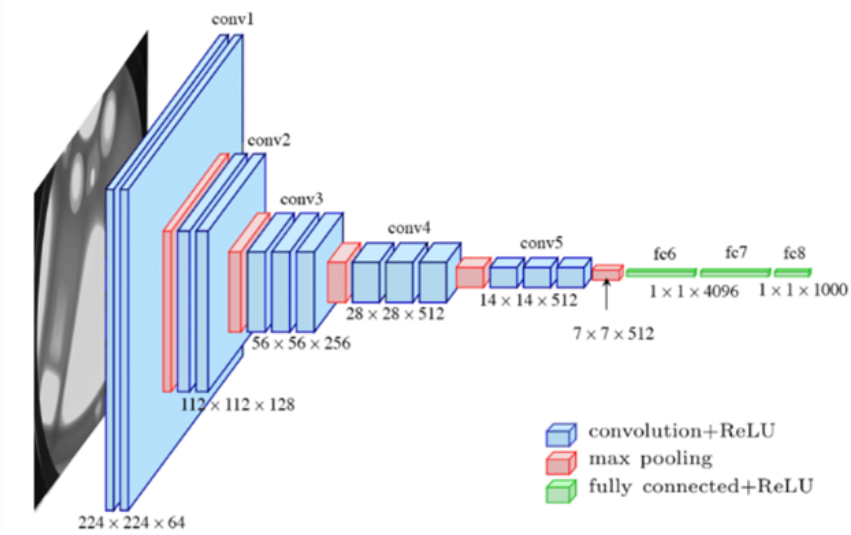


Figure 1: Training and Validation Curves for MobileNet

The architecture, shown in Figure 1, is optimized to deliver high accuracy while minimizing computational costs. MobileNet achieves this by employing depthwise separable convolutions, a technique that decomposes the standard convolution operation into two layers: depthwise convolution and pointwise convolution.

Depthwise Convolution This operation applies single filter to each input channel individually, dramatically reducing computation compared to traditional convolutions. Mathematically, it is expressed as:

$$Y_{i,j,k} = \sum_{n=1}^{N} (X_{i,j,n} \cdot W_{n,k}) \tag{1}$$

where $Y_{i,j,k}$ represents the output at pixel (i, j) for channel k, $X_{i,j,n}$ is the input pixel in channel n, and $W_{n,k}$ is the filter weight.

Pointwise Convolution Following depthwise convolution, this step employs a 1×1 convolution to combine depthwise outputs across channels:

$$Z_{i,j,k} = \sum_{n=1}^{M} (Y_{i,j,n} \cdot V_{n,k}) \tag{2}$$

where $Z_{i,j,k}$ represents the final output at pixel (i, j) for channel k, $Y_{i,j,n}$ is the depthwise output, and $V_{n,k}$ is the pointwise filter weight.

This architectural design reduces the number of parameters and computations significantly, making MobileNet suitable for real-time applications. Additionally, MobileNet incorporates the **ReLU activation** function to introduce non-linearity, defined as:

$$\text{ReLU}(x) = \max(0, x) \tag{3}$$

Normalization techniques, such as batch normalization, further stabilize and accelerate the training process by normalizing activations:

$$\hat{x} = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{4}$$

where $\hat{x}$ is the normalized output, μ is the mean, σ^2 is the variance, and ϵ is a small constant to prevent division by zero.

3.2.2 Implementation of InceptionV3

InceptionV3 was selected for its ability to extract diverse features efficiently through its advanced modular architecture. The Inception Modules simultaneously apply convolutions of varying kernel sizes (1×1, 3×3, 5×5) and pooling operations to capture patterns at multiple scales:

$$O = [\text{Conv}_{1\times1}(X), \text{Conv}_{3\times3}(X), \text{Conv}_{5\times5}(X), \text{Pool}_{1\times1}(X)] \tag{5}$$

To improve efficiency, 1×1 convolutions are used to reduce dimensionality before applying larger filters. This minimizes computation without sacrificing representational power. Additionally, auxiliary classifiers are embedded at intermediate layers to stabilize training by providing additional gradient signals. These classifiers are defined as:

$$O_{aux} = FC(\text{GlobalAvgPool}(X)) \tag{6}$$

Other key features include:

- **Batch Normalization:** Normalizes intermediate activations to enhance stability and convergence speed:

$$\hat{x} = \left(\frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \right) \cdot \gamma + \beta \tag{7}$$

- **Activation Functions:** InceptionV3 leverages ReLU for its simplicity and efficiency, along with Leaky ReLU to address the vanishing gradient problem:

$$\text{LeakyReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{if } x < 0 \end{cases} \tag{8}$$

- **Global Average Pooling:** Reduces each feature map to a single value, mitigating overfitting and minimizing parameters:

$$O = \frac{1}{h \times w} \sum_{i=1}^{h} \sum_{j=1}^{w} X_{i,j} \tag{9}$$

3.2.3 Model Training and Optimization

Both MobileNet and InceptionV3 were initialized with pre-trained weights from the ImageNet dataset, leveraging transfer learning to adapt to the wildlife classification task. Key hyperparameters included:

- Learning rate: 0.0010001
- Batch size: 3232
- Epochs: 5050
- Dropout rate: 0.50

The use of data augmentation further enhanced generalization, and dropout prevented overfitting. MobileNet's lightweight design allowed faster training compared to InceptionV3, while its depthwise separable convolutions contributed to lower computational overhead, which proved advantageous in achieving higher accuracy and lower loss. In contrast, InceptionV3's ability to extract features across multiple scales enhanced its capability to handle complex patterns in the dataset. However, its deeper architecture and auxiliary classifiers required greater computational resources, which slightly affected its performance in terms of loss and training speed.

3.2.4 Transfer Learning Approach

To adapt MobileNet and InceptionV3 for the CUB-200-2011 bird species dataset, transfer learning was employed to leverage the pre-trained weights from ImageNet. Initially, the convolutional base layers of both models were frozen, preserving the pre-trained weights to act as feature extractors while new classification layers were trained on the bird species dataset. Custom fully connected layers, followed by a softmax layer for classifying the 200 bird species, were added to the models. These new layers, initialized with random weights, were trained first, leaving the base layers untouched. After stabilizing the new layers, fine-tuning was performed by unfreezing the last 15 layers of the convolutional base in MobileNet and the last 10 layers in InceptionV3, allowing the models to adapt their feature extraction to the specific characteristics of the CUB-200-2011 dataset while retaining the general features learned from ImageNet. Fine-tuning was carried out with a reduced learning rate to prevent overfitting or drastically altering the pre-trained weights, and regularization techniques, including dropout with a rate of 0.5, were applied. Both models were trained using the Adam optimizer with a learning rate of 0.001, a batch size of 32, and for 10 epochs, employing early stopping based on validation performance. This two-phase training strategy ensured that the models effectively leveraged pre-trained knowledge while adapting to the nuances of the bird classification task.

3.3 Comparative Analysis

The evaluation of MobileNet and InceptionV3 highlights significant differences in computational efficiency and inference speed. MobileNet's lightweight architecture ensures faster inference and lower memory usage, crucial for deployment on devices with limited resources. InceptionV3's higher computational complexity results in increased latency and power consumption, making it less ideal for scenarios requiring real-time processing. Table 1 summarizes the comparison between MobileNet and InceptionV3 for key efficiency metrics:

Model	**FLOPs (224×224)**	**Model Size**	**Inference Speed (CPU)**	**Suitability for Edge Devices**
MobileNet	569M	∼17MB	∼3ms/image	Highly suitable
InceptionV3	5.7B	∼92MB	∼25ms/image	Less suitable

Table 1: Comparison of MobileNet and InceptionV3 on key efficiency metrics

4 Results and Discussions

This section presents the results from the image classification tasks performed using MobileNet and InceptionV3 models, focusing on key performance metrics such as accuracy, loss, precision, recall, and F1-score. This section also provides a comparison of the two models, evaluating their efficiency, model complexity, and training time, highlighting the trade-offs between performance and resource requirements.

4.1 Accuracy

The accuracy metric, which measures the proportion of correctly classified images, is a key indicator of overall model performance. It is defined as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{10}$$

In this research work, both MobileNet and InceptionV3 were evaluated on a wildlife image classification task. MobileNet achieved an accuracy of 0.7460, outperforming InceptionV3, which had an accuracy of 0.6556. This difference indicates that MobileNet was more effective in correctly identifying and classifying the test dataset images. MobileNet's higher accuracy can be attributed to its streamlined architecture, which is optimized for efficiency without sacrificing performance. Its design allows for effective feature extraction, a crucial factor in achieving high accuracy in image classification tasks. The model's ability to balance computational efficiency with robust performance makes it well-suited for this specific task. In contrast, InceptionV3, despite being a deeper and more complex model, did not perform as well. This could be due to its higher capacity, which may require larger datasets or more specific tuning to reach its full potential. The complexity of InceptionV3 might not have translated into better accuracy for this dataset, possibly due to the nature of the data or the specific training configurations used.

4.2 Loss Function

The loss function is a critical metric for evaluating how well a model's predictions align with the actual class labels. In this project, MobileNet exhibited a lower loss value of 0.8685 compared to InceptionV3's loss of 1.128. A lower loss for MobileNet indicates that its predictions were closer to the true labels, making it more reliable for generalizing to unseen data. This suggests that MobileNet's architecture, which emphasizes efficiency, contributed to better alignment between its outputs and the actual class labels. In contrast, the higher loss in InceptionV3 suggests its predictions were less accurate, leading to a less reliable model. This could be due to overfitting, where the model becomes overly complex and captures noise rather than relevant patterns. Alternatively, it may indicate that InceptionV3 requires more extensive training or further tuning of its hyperparameters to improve performance. The loss function used in this project is categorical cross-entropy, which calculates the difference between the predicted class probabilities and the true labels. It is defined as:

$$\text{Loss} = -\sum_{i=1}^{c} y_i \log(p_i) \tag{11}$$

where y_i is the true label and p_i is the predicted probability for class i. A lower loss value indicates better model performance.

4.3 Precision and Recall

Precision measures the proportion of true positive predictions among all positive predictions, defined as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives + False Positives}} \tag{12}$$

A high precision score indicates that when the model predicts a certain class, it is highly likely to be correct. In this project, MobileNet demonstrated precision values ranging from 0.45 for the ”Laysan Albatross” class to 0.93 for the ”Groove-billed Ani” class. This suggests that while MobileNet was highly accurate in predicting certain classes, it faced challenges in others. InceptionV3, by comparison, showed precision values between 0.65 and 0.81, generally lower than MobileNet. This indicates that InceptionV3 had more false positives, resulting in lower precision overall.

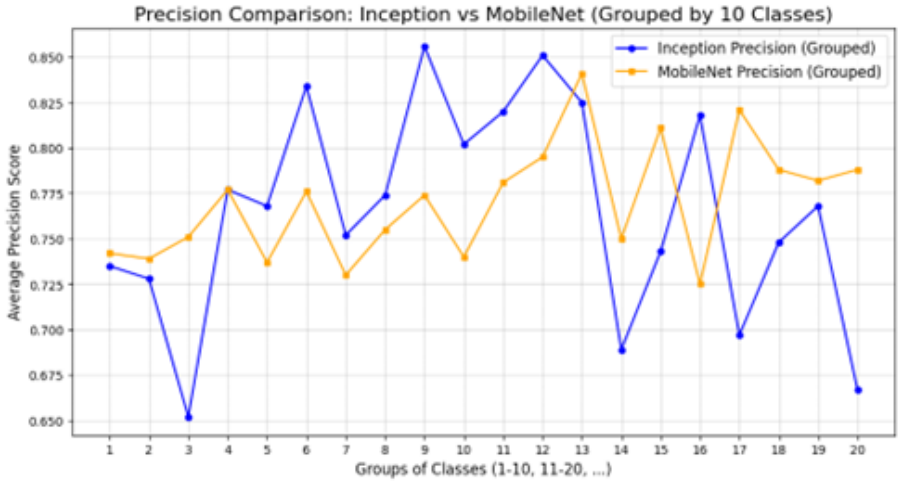


Figure 2: Precision Score Comparison: Inception vs MobileNet

Recall evaluates the proportion of true positive predictions among all actual positives and is defined as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}} \tag{13}$$

High recall means the model effectively identifies all instances of a particular class. MobileNet exhibited recall values ranging from 0.37 for the ”Laysan Albatross” class to 0.96 for the ”Yellow-headed Blackbird” class, indicating variability in its ability to correctly identify certain classes. InceptionV3’s recall values ranged from 0.57 to 0.95, also showing variation but generally lower than MobileNet, which suggests that InceptionV3 missed more instances of certain classes (higher miss rate).

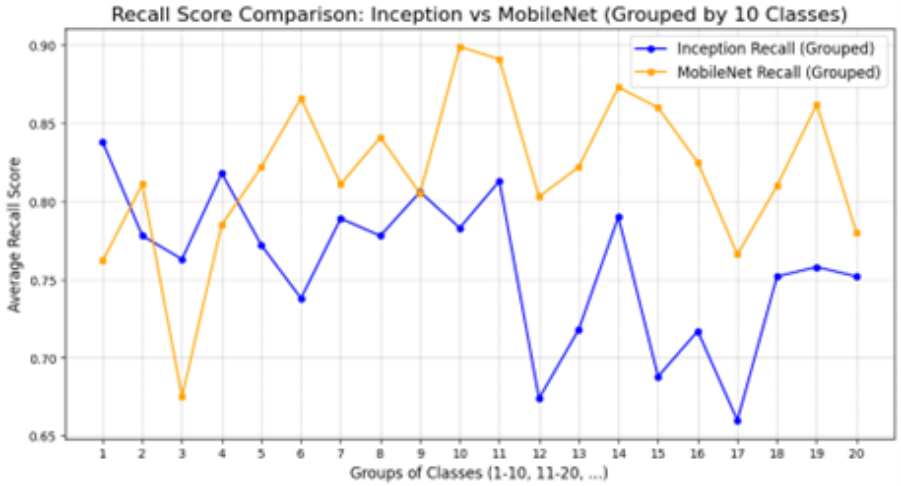


Figure 3: Recall Score Comparison: InceptionV3 vs MobileNet

4.4 F1 Score

The F1 Score, defined as the harmonic mean of precision and recall, is calculated as:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{14}$$

This metric is particularly useful in situations with uneven class distribution or when both precision and recall are critical to assess. It provides a balanced measure of a model's accuracy, especially when there is a trade-off between precision and recall. MobileNet's F1-scores ranged from 0.40 to 0.91, indicating a balanced and effective performance across different classes. In contrast, InceptionV3's F1-scores, while consistent with its precision and recall metrics, were generally lower, suggesting a more stable but less effective classification ability compared to MobileNet. This highlights MobileNet's superior performance in achieving a better balance between precision and recall across varying classes.

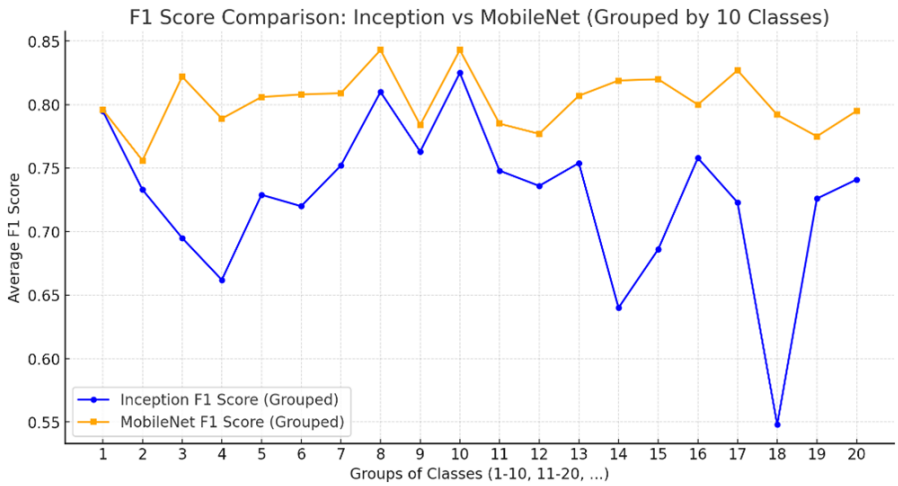
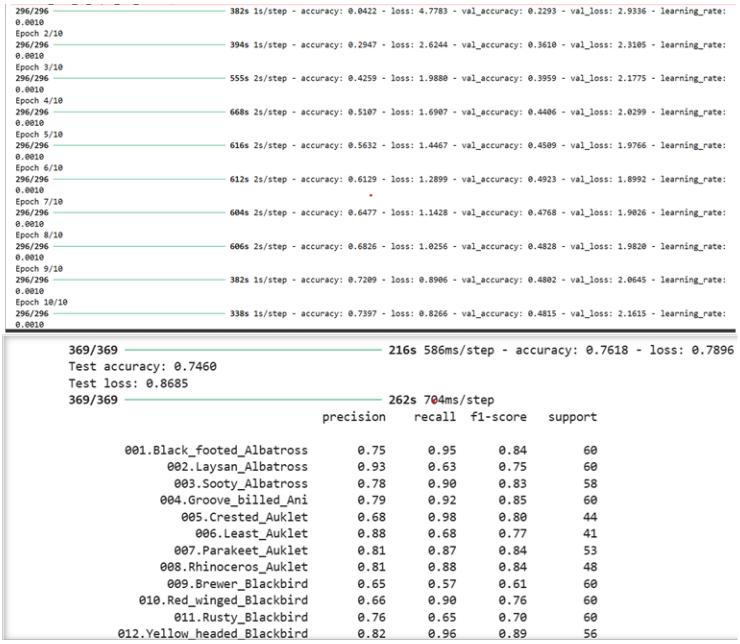


Figure 4: F1 Score Comparison: Inception vs MobileNet

4.5 Performance on CUB-200-2011 Dataset

The project employed MobileNet and InceptionV3 to classify bird images from the CUB-200-2011 dataset, with performance evaluated using various metrics. MobileNet demonstrated strong results, achieving an accuracy of 0.7460, which reflects its high effectiveness in predicting the correct classes in the test dataset. This performance can be attributed to its streamlined architecture, which efficiently extracts relevant features from the images. Additionally, MobileNet's loss value of 0.8685 indicates that its predictions closely aligned with the actual class labels, further showcasing its reliability and precision compared to InceptionV3.



```
296/296 ━━━━━━━━━━━━━━━━━━━━ 382s 1s/step - accuracy: 0.0422 - loss: 4.7783 - val_accuracy: 0.2293 - val_loss: 2.9336 - learning_rate:
0.0010
Epoch 2/10
296/296 ━━━━━━━━━━━━━━━━━━━━ 394s 1s/step - accuracy: 0.2947 - loss: 2.6244 - val_accuracy: 0.3610 - val_loss: 2.3105 - learning_rate:
0.0010
Epoch 3/10
296/296 ━━━━━━━━━━━━━━━━━━━━ 555s 2s/step - accuracy: 0.4259 - loss: 1.9880 - val_accuracy: 0.3959 - val_loss: 2.1775 - learning_rate:
0.0010
Epoch 4/10
296/296 ━━━━━━━━━━━━━━━━━━━━ 668s 2s/step - accuracy: 0.5107 - loss: 1.6907 - val_accuracy: 0.4406 - val_loss: 2.0299 - learning_rate:
0.0010
Epoch 5/10
296/296 ━━━━━━━━━━━━━━━━━━━━ 616s 2s/step - accuracy: 0.5632 - loss: 1.4467 - val_accuracy: 0.4509 - val_loss: 1.9766 - learning_rate:
0.0010
Epoch 6/10
296/296 ━━━━━━━━━━━━━━━━━━━━ 612s 2s/step - accuracy: 0.6129 - loss: 1.2899 - val_accuracy: 0.4923 - val_loss: 1.8992 - learning_rate:
0.0010
Epoch 7/10
296/296 ━━━━━━━━━━━━━━━━━━━━ 604s 2s/step - accuracy: 0.6477 - loss: 1.1428 - val_accuracy: 0.4768 - val_loss: 1.9026 - learning_rate:
0.0010
Epoch 8/10
296/296 ━━━━━━━━━━━━━━━━━━━━ 606s 2s/step - accuracy: 0.6826 - loss: 1.0256 - val_accuracy: 0.4828 - val_loss: 1.9820 - learning_rate:
0.0010
Epoch 9/10
296/296 ━━━━━━━━━━━━━━━━━━━━ 382s 1s/step - accuracy: 0.7209 - loss: 0.8906 - val_accuracy: 0.4802 - val_loss: 2.0645 - learning_rate:
0.0010
Epoch 10/10
296/296 ━━━━━━━━━━━━━━━━━━━━ 338s 1s/step - accuracy: 0.7397 - loss: 0.8266 - val_accuracy: 0.4815 - val_loss: 2.1615 - learning_rate:
0.0010

369/369 ━━━━━━━━━━━━━━━━━━━━ 216s 586ms/step - accuracy: 0.7618 - loss: 0.7896
Test accuracy: 0.7460
Test loss: 0.8685
369/369 ━━━━━━━━━━━━━━━━━━━━ 262s 704ms/step
                              precision    recall  f1-score   support

     001.Black_footed_Albatross       0.75      0.95      0.84        60
          002.Laysan_Albatross       0.93      0.63      0.75        60
          003.Sooty_Albatross       0.78      0.90      0.83        58
     004.Groove_billed_Ani          0.79      0.92      0.85        60
        005.Crested_Auklet          0.68      0.98      0.80        44
           006.Least_Auklet         0.88      0.68      0.77        41
        007.Parakeet_Auklet         0.81      0.87      0.84        53
     008.Rhinoceros_Auklet          0.81      0.88      0.84        48
        009.Brewer_Blackbird        0.65      0.57      0.61        60
   010.Red_winged_Blackbird         0.66      0.90      0.76        60
        011.Rusty_Blackbird         0.76      0.65      0.70        60
012.Yellow_headed_Blackbird         0.82      0.96      0.89        56
```

Figure 5: MobileNet Accuracy and Loss Performance

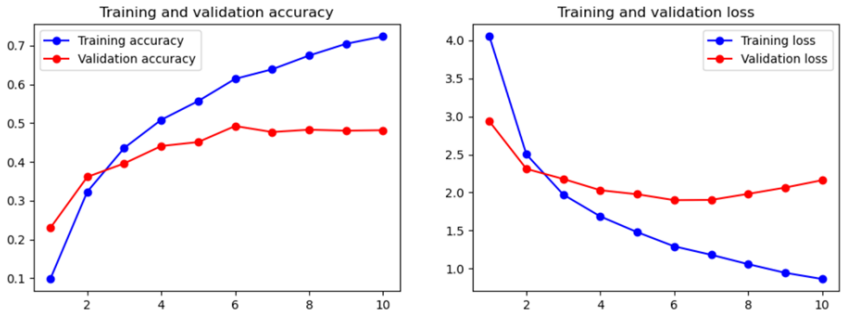


Figure 6: Training and Validation Curves for MobileNet

InceptionV3 achieved an accuracy of 0.6400, which was lower than MobileNet's performance, indicating that despite its complex architecture, it was less effective at correctly classifying bird images in this dataset. The model's loss was 1.128, higher than that of MobileNet, suggesting greater difficulty in aligning its predictions with the actual class labels. This higher loss value reflects InceptionV3's challenges in making accurate predictions, further highlighting its comparatively reduced effectiveness in this task.

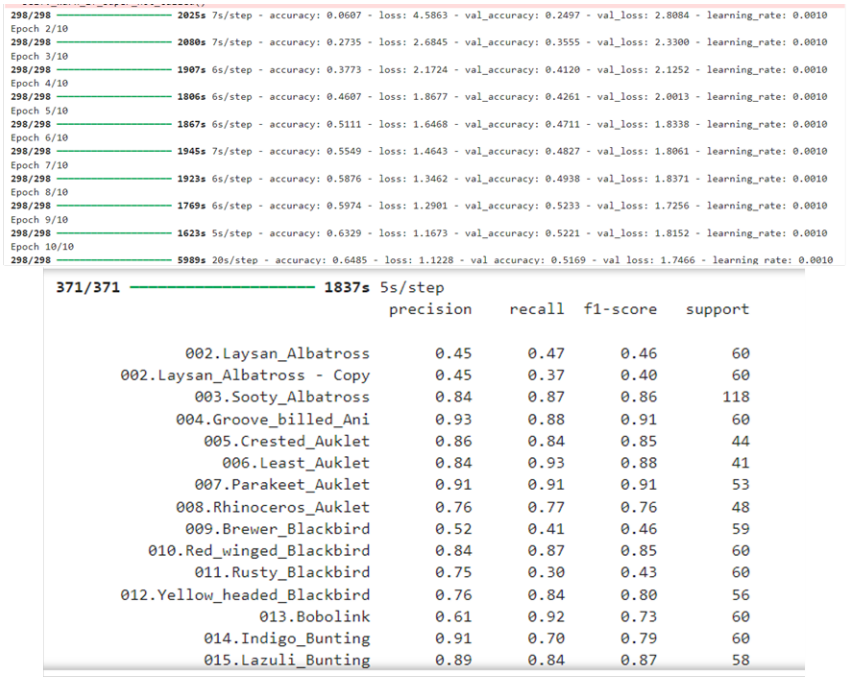


Figure 7: InceptionV3 Accuracy and Loss Performance

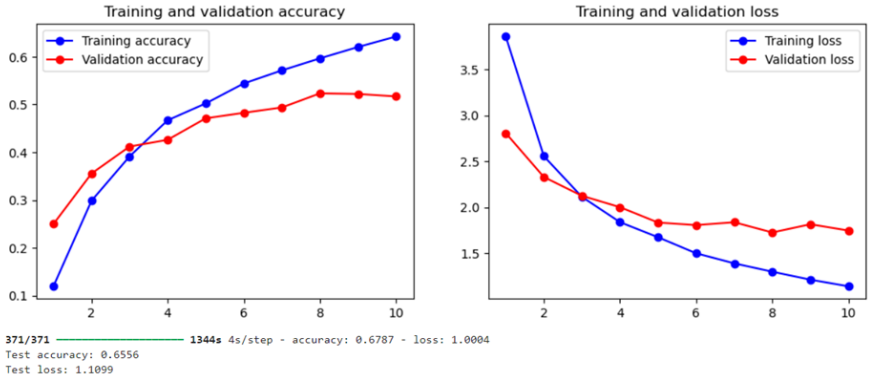


Figure 8: Training and Validation Curves for InceptionV3

4.6 Comparison of MobileNet and InceptionV3 Metrics

Table2 summarizes key performance metrics for both MobileNet and InceptionV3 models in bird species image classification and recognition. It highlights the strengths and weaknesses of each model, providing a clear comparison that supports the overall findings of the research.

Metric	MobileNet	InceptionV3	Discussion
Accuracy	74.60%	65.56%	MobileNet's higher accuracy indicates superior performance in the classification task.
Loss	0.8685	1.128	Lower loss in MobileNet suggests better alignment with actual class labels compared to InceptionV3.
Precision	0.65 to 0.81	0.45 to 0.93	MobileNet shows varying performance across classes, while InceptionV3 has generally lower precision.
Recall	0.37 (Laysan Albatross) to 0.96 (Yellow-headed Blackbird)	0.57 to 0.95	MobileNet performed better overall, but InceptionV3 had a higher miss rate for certain classes.
F1-Score	0.71 to 0.94	0.40 to 0.91	MobileNet's F1-scores reflect balanced performance, while InceptionV3's lower scores indicate a more stable but less effective classification ability.

Table 2: Summary of Performance Metrics for MobileNet and InceptionV3 Models

In the proposed application, InceptionV3's complex architecture increases the risk of overfitting due to its higher parameter count and redundant feature extraction, especially on a dataset with limited samples. This leads to higher loss and lower accuracy for certain inputs. In contrast, MobileNet's lightweight design efficiently generalizes by focusing on essential features, resulting in better performance. The evaluation of MobileNet and InceptionV3 on the CUB-200-2011 dataset revealed that MobileNet outperformed InceptionV3 in terms of both accuracy and loss metrics. MobileNet's lightweight architecture, which employs depthwise separable convolutions, significantly reduces the number of parameters and computations compared to standard convolutions. This efficiency allows MobileNet to learn effectively with fewer resources, minimizing the risk of overfitting and enhancing generalization, particularly on moderately sized datasets like CUB-200-2011. Additionally, the modular design of MobileNet ensures that it captures essential spatial and channel-wise features with minimal redundancy, enabling superior feature extraction for bird species classification. In contrast, InceptionV3's advanced modules, such as mixed convolutions of varying kernel sizes and auxiliary classifiers, support effective feature learning but increase model complexity. This higher complexity can lead to overfitting, especially when training on datasets with limited samples per class. MobileNet's adaptability to the characteristics of the CUB-200-2011 dataset also contributed to its better performance. The smaller receptive fields and computational efficiency of MobileNet make it particularly suited to datasets with high intra-class variability and fine-grained distinctions, such as bird species. On the other hand, InceptionV3's broader feature extraction approach may capture redundant features, which can result in slightly lower accuracy and higher loss. Furthermore, MobileNet's streamlined architecture ensures faster convergence and better performance, particularly on edge devices or datasets with fewer training samples. Meanwhile, InceptionV3's reliance on multi-scale convolutions and its overall architectural complexity increase computational overhead, potentially leading to slower convergence and higher loss. Training behavior and regularization techniques played a critical role in differentiating the models' performance. MobileNet's lightweight architecture allowed for more effective fine-tuning during the transfer learning phase. While both models incorporated dropout and batch normalization for regularization, MobileNet's smaller parameter space benefited more from these techniques, further mitigating overfitting and enhancing performance. These combined factors—efficient architecture, adaptability to dataset characteristics, and effective training regularization—highlight why MobileNet achieved superior results compared to InceptionV3 on the CUB-200-2011 dataset.

4.7 Visualization of the User Interface

The user interface (UI) developed for this project effectively demonstrates the capabilities of the deep learning models in a real-world application, as shown in Figure 9. It allows users to easily upload images of birds via a user-friendly web page. Once an image is submitted, the backend processes it using the trained MobileNet and InceptionV3 models to classify the bird species, as illustrated in Figures 10 and 11. The classification results are then presented on the web page, displaying the predicted species name.

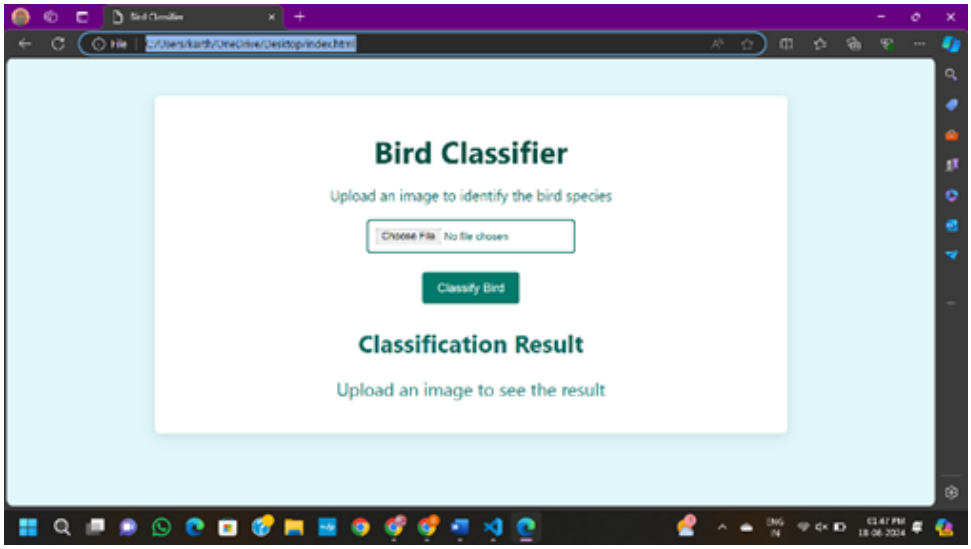


Figure 9: Webpage for uploading the user input images

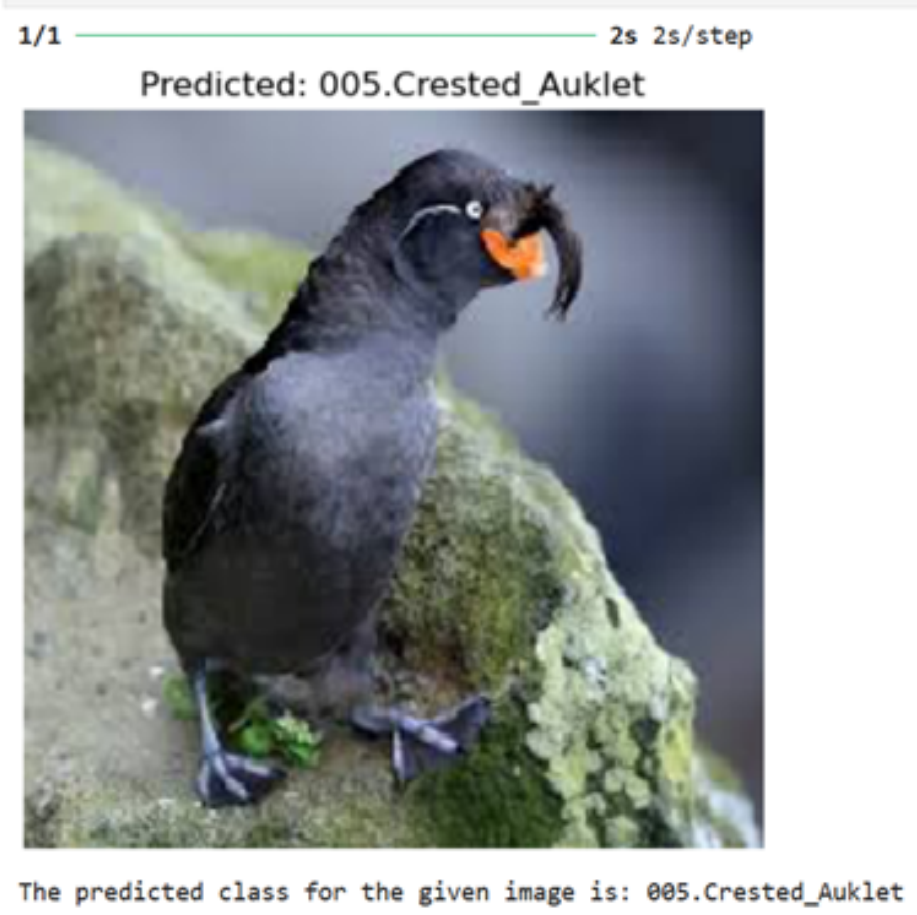


Figure 10: Predicted outputs for MobileNet

Figure 11: Predicted outputs for InceptionV3

Figure 12: Results page showing the outcome of the image classification

To ensure the user interface is user-friendly and effective for non-expert users like conservationists or field biologists, it was designed to be minimalistic with no extra components. The interface features just two main buttons: a ”Browse” button to easily upload wildlife images and a ”Classify” button to instantly display the classification results. This simple layout ensures ease of use and eliminates any complexity for non-technical users. A distinctive feature of this research that significantly enhances its accessibility and practical applicability is its web-based user interface (UI). This UI allows users, including conservationists, biologists, and non-experts, to upload and classify bird images with ease. The platform's user-friendly design enables anyone to interact with the deep learning model without requiring specialized technical knowledge, thus extending the model's usability beyond the realm of researchers and experts. The ability to seamlessly upload images and receive real-time classifications

marks a significant improvement over previous studies, which often lacked such accessible and practical implementations. This UI adds a tangible, real-world application to the research, making it not only a theoretical advancement but also a tool that can be used in everyday conservation efforts.

5 Conclusion

This project demonstrates the successful application of deep learning to the complex task of wildlife image classification, focusing on the MobileNet and InceptionV3 models. By employing advanced data preprocessing techniques and leveraging transfer learning, the study achieved results that surpass those reported in existing literature. Notably, MobileNet, with its efficient and streamlined architecture, outperformed the more complex InceptionV3 in terms of accuracy and loss metrics. This finding contrasts with previous studies, where InceptionV3 often exceeded the performance of lighter models. MobileNet's superior accuracy, coupled with its computational efficiency, highlights its practical advantages for real-world applications, where rapid deployment and resource optimization are critical. Overall, this work not only advances the state of wildlife image classification but also underscores the importance of selecting the appropriate model for specific tasks. By optimizing the trade-off between model performance and computational efficiency, this research provides a scalable solution that can be implemented across a wide range of wildlife conservation applications, ultimately aiding in faster, more accurate species identification in diverse environments. Our work challenges this norm by demonstrating that lightweight models like MobileNet can be just as effective for specific tasks while offering substantial computational advantages, especially in practical, real-world deployment scenarios. The results establish MobileNet as a compelling choice for both academic research and practical conservation efforts, setting a new standard for future studies in this domain.

6 FUTURE ENHANCEMENT

While this study demonstrates the effectiveness of MobileNet for wildlife image classification, there are inherent limitations that must be addressed in future work. The current model's performance has been evaluated in a controlled environment, but real-time field testing is crucial for understanding how it performs under varying real-world conditions. Factors such as lighting changes, poor image quality, and complex or cluttered backgrounds could impact the accuracy of the model in the field. Additionally, deploying the model on edge devices presents challenges related to limited computational resources, memory, and energy consumption. Future efforts will focus on optimizing the model for real-time field testing, ensuring that it can deliver reliable results despite these challenges, and adapting it for deployment on edge devices with restricted processing power.

The future of wildlife image classification lies in the integration of deep learning models on edge devices, which would enable more accessible and efficient field monitoring. Edge devices, such as low-power cameras and portable monitoring systems, are essential for wildlife conservation in remote locations where cloud-based solutions may not be feasible. However, deploying models on these devices requires overcoming challenges such as model compression, efficient inference with minimal latency, and maintaining accuracy while running on limited hardware. Addressing these challenges will enable the practical use of deep learning models for real-time wildlife monitoring in the field, paving the way for scalable, proactive conservation efforts. Further work will involve fine-tuning the model to ensure its robustness and adaptability, facilitating the widespread use of this technology in diverse environments.

References

[1] LeCun, Y., Bengio, Y., & Haffner, P. (2015). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324. https://doi.org/10.1109/5.726791

[2] LeCun, Y., & others. (2015). Deep learning. *Nature*, 521(7553), 436-444. https://doi.org/10.1038/nature14539

[3] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 1097-1105.

[4] Horn, G., & others. (2018). iNaturalist Species Classification Challenge. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 206-215. https://doi.org/10.1109/CVPRW.2018.00027

[5] Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[6] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770-778. https://doi.org/10.1109/CVPR.2016.90

[7] Szegedy, C., Vanhoucke, V., Vinyals, O., & others. (2016). Rethinking the inception architecture for computer vision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2818-2826. https://doi.org/10.1109/CVPR.2016.308

[8] Howard, A. G., Sandler, M., Chen, W., Chen, L. H., & others. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

[9] Gupta, A., & others. (2020). Evaluating MobileNet for wildlife classification. *Proceedings of the International Conference on Machine Learning and Computing*, 77-82. https://doi.org/10.1145/3312141.3312151

[10] Liu, X., & others. (2019). Fine-grained classification of bird species using InceptionV3. *IEEE Access*, 7, 77914-77923. https://doi.org/10.1109/ACCESS.2019.2915561

[11] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2016). Inception-v4, Inception-ResNet and the impact of residual connections on learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).

[12] Zhang, J., Chen, Y., & others. (2020). Transfer learning for wildlife image classification: A comprehensive survey. *Artificial Intelligence Review*, 53(5), 3701-3722. https://doi.org/10.1007/s10462-019-09769-7

[13] Tang, Y., & others. (2021). A comparative study of MobileNet and InceptionV3 for wildlife classification. *Journal of Computer Science and Technology*, 36(2), 387-398. https://doi.org/10.1007/s11390-021-2155-0

[14] Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1717-1724. https://doi.org/10.1109/CVPR.2014.223

[15] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 1-48. https://doi.org/10.1186/s40537-019-0197-0

[16] Chen, Y., Li, M., & others. (2019). Improving wildlife image classification with data augmentation techniques. *Ecological Informatics*, 53, 22-30. https://doi.org/10.1016/j.ecoinf.2019.04.004

[17] Kim, H., & others. (2018). Ensemble learning for improved species identification. *BMC Bioinformatics*, 19(1), 152. https://doi.org/10.1186/s12859-018-2101-7

[18] Wu, J., & others. (2020). Addressing data imbalance in wildlife classification using ensemble methods. *Scientific Reports*, 10(1), 1-10. https://doi.org/10.1038/s41598-020-70444-1

[19] McCarthy, T., & others. (2021). The role of automated systems in biodiversity monitoring: A case study on poaching detection. *Biodiversity and Conservation*, 30(3), 703-726. https://doi.org/10.1007/s10531-021-02044-9

[20] M. P., & others. (2019). Leveraging deep learning for wildlife monitoring and conservation. *Frontiers in Ecology and the Environment*, 17(2), 86-94. https://doi.org/10.1002/fee.2025

[21] Noroozi, M., & others. (2020). Addressing occlusion and variable poses in wildlife image classification. *Proceedings of the IEEE International Conference on Computer Vision*, 932-941. https://doi.org/10.1109/ICCV.2019.00085

[22] C. C., & others. (2021). A survey of wildlife image classification: Challenges and opportunities. *Journal of Applied Ecology*, 58(2), 291-303. https://doi.org/10.1111/1365-2664.13753

[23] R. S., & others. (2018). Automated detection of poaching activities using deep learning models. *Conservation Biology*, 32(6), 1377-1388. https://doi.org/10.1111/cobi.13197