Optimizing precision in high variablity image dataset through adaptive ensemble of deep convolution neural network

Katam Veera Goutham, Omprakash S M, Sudeep Suresh, Sneha H R
Department of Information Science and Engineering
Nitte Meenakshi Institute of Technology, Bengaluru Karnataka, India
veeragoutham04@gmail.com, ompi12527@gmail.com, devadigasudeep03@gmail.com, snehahr1990@gmail.com

***Abstract*—Image classification in high-variability datasets presents challenges due to intra-class variability and inter-class similarity. This project explores a hybrid approach integrating NASNetLarge and EfficientNetB7 within an adaptive ensemble framework to improve classification precision. Baseline models are trained individually and compared to the hybrid system, which combines their diverse strengths for enhanced performance. Experiments on the CUB-200-2011 bird species dataset demonstrate significant improvements in accuracy and robustness. Hyperparameter variations are analyzed to optimize the architecture, showcasing the hybrid model's potential for fine-grained image classification in real-world scenarios.**

***Index Terms*—Image Classification, Variability, Deep Learning, Hybrid Models, Convolutional Neural Networks (CNNs), Ensemble Learning.**

I. Introduction

Bird classification is a challenging task in computer vision due to the vast diversity of bird species, their varying appearances, and subtle inter-species differences. [1] Deep learning advancements have enabled highly accurate models for complex classification tasks, including bird species identification. This project focuses on developing an efficient bird classification system through hybrid deep learning architectures.

The project begins by collecting a robust bird species dataset from Kaggle. [2] Data augmentation techniques, including rotation, zoom, shear, and horizontal flips, are applied to improve model generalization and ensure resilience under diverse conditions.

ResNet50, a pre-trained convolutional neural network (CNN), is used initially as a baseline model, achieving an accuracy of 75%. While effective, ResNet50's performance is limited by intra-class variability. To address this, a hybrid model integrating NASNetLarge and EfficientNetB7 is designed. By combining the outputs of these state-of-the-art architectures after global average pooling, the hybrid model enhances feature extraction and improves classification accuracy to approximately 78%.

To prevent overfitting and enhance robustness, techniques like dropout, early stopping, and learning rate reduction are incorporated. The model is trained for 50 epochs using categorical cross-entropy as the loss function and Adam optimizer. Performance is evaluated through accuracy and loss metrics, with visualizations plotted over epochs. The trained model is saved and tested on single input images, verifying real-world applicability.

This work highlights the effectiveness of hybrid deep learning architectures in solving intricate classification tasks, laying a strong foundation for further advancements in bird species identification and similar domains. [3]

Section II explores related works, Section III details the proposed methodology and experimental setup, Section IV presents results and analysis, and Section V concludes with discussions and future directions.

II. Related work

- The project ”Optimizing Precision in High Variability Image Datasets through Adaptive Ensembles of Deep Convolutional Neural Networks (CNNs)” aims to improve wildlife monitoring accuracy using adaptive CNN ensembles. This approach addresses challenges in camera trap datasets from the Wildlife Spotter project, featuring labeled and unlabeled images from South-central Victoria, Australia. By employing deep CNN architectures, transfer learning with ImageNet-pretrained models, and fine-tuning, the system achieves 96.6% accuracy in detecting animals and 90.4% in identifying the three most common species. The results highlight the potential of automated systems to enhance efficiency and data reliability in ecological research. [4]

- The research focuses on the application of deep convolutional neural networks (CNNs) for the recognition of animal species from ecological camera trap images. [5] [6] The primary goal is to classify three groups of herpetofaunal species—snakes, lizards, and toads/frogs—by leveraging advanced deep learning architectures such as VGG16 and ResNet50, alongside a self-trained CNN model. The study addresses the challenges posed by imbalanced datasets and employs various image preprocessing and augmentation techniques to enhance classification accuracy. Experiments demonstrated high accuracy in binary classification tasks, particularly for snakes and toads, while multi classification tasks achieved an accuracy of up to 87% with VGG16 and ResNet50 models. The research underscores the potential of automated

979-8-3503-7311-0/24/$31.00 ©2024 IEEE

species identification in facilitating conservation efforts and monitoring endangered species through ecological camera trap images. [7]

- This paper discusses the use of deep convolutional neural networks (CNNs) for detecting extreme weather events in climate datasets. Traditional methods rely on human-defined thresholds for identifying these events, but this approach can be subjective and inconsistent. The authors developed a deep CNN system that can learn to identify patterns associated with extreme weather events such as tropical cyclones, atmospheric rivers, and weather fronts from large climate datasets. The system was trained using labeled data and optimized with a Bayesian-based hyper-parameter scheme, achieving an accuracy of 89% to 99%. [8]

- This paper presents PlaNet, a deep convolutional neural network (DCNN) model designed to identify diseases in plant leaves. The model aims to improve accuracy and robustness in recognizing various plant diseases from images. The researchers collected a diverse dataset of plant leaves affected by different diseases and used it to train the PlaNet model. They also implemented techniques such as data augmentation, transfer learning, and hyperparameter tuning to enhance the model's ability to generalize across different types of diseases and environmental conditions.
  The study compares PlaNet with other existing models and demonstrates its superior performance in terms of accuracy, precision, recall, and F1-score. The results showed that PlaNet achieved high accuracy in disease recognition, making it a reliable tool for early detection and management of plant diseases. The paper concludes that the integration of advanced CNN techniques can significantly improve the performance of disease detection systems in agriculture, potentially leading to better crop management and yield. [9]

- This project explores the use of deep convolutional neural networks (CNNs) and ensemble learning to improve classification accuracy in high-variability image datasets. By combining multiple CNN models in an adaptive ensemble framework, the approach enhances robustness and leverages the strengths of individual models. Generative adversarial networks (GANs) are employed to generate synthetic data, addressing challenges of limited labeled data.
  Focusing on medical imaging, the study investigates Alzheimer's disease classification using multi-modal MRI and PET data. The integration of CNNs, ensemble learning, and GANs aims to improve early detection and diagnosis, supporting timely therapeutic interventions. These methods show promise for advancing image classification across diverse, complex datasets. [10]

- This paper explores optimizing image classification in high-variability datasets through an adaptive ensemble of deep convolutional neural networks (CNNs). By combining multiple CNN models, the ensemble leverages their strengths while mitigating weaknesses, improving robustness and accuracy. Generative adversarial networks (GANs) generate synthetic data to address limited data availability, enhancing model training. Focusing on medical imaging, the study uses multi-modal MRI and PET data for Alzheimer's disease classification, aiming to improve early detection and diagnosis. Advanced optimization techniques like hyperparameter tuning and regularization ensure the models generalize well to unseen data. Rigorous evaluations and comparative analyses validate the framework's effectiveness, demonstrating its superiority in handling complex datasets. [11]

III. Proposed Method

The proposed bird species classification system [12] was developed through a structured approach involving data preparation, model construction, training, and evaluation. [6] The following steps were undertaken:

Dataset Collection

- **Data Collection:**
  A labeled dataset of bird images was obtained from Kaggle.The dataset was split into training and testing sets to ensure model evaluation on unseen data.
- **Data Augmentation:**
  To improve model generalization and handle variations in the dataset, data augmentation techniques were applied:
  Rescaling pixel values to [0, 1].
  Applying transformations such as rotation, shifting, shearing, zooming, and horizontal flipping.
  Separate data generators were created for training and testing, with only rescaling applied to the testing data.
- **Baseline Model (ResNet50)**
  ResNet50, pre-trained on ImageNet, was used to establish baseline performance, achieving 75% accuracy. ResNet50 provided strong initial results but faced limitations with highly variable datasets, necessitating a more robust hybrid approach. [13]
- **Hybrid Model Development**
  Two pre-trained models, NASNetLarge and EfficientNetB7, were selected for their strong feature extraction capabilities. The steps involved:
  Load both models pre-trained on ImageNet,excluding their top layers.
  Freeze their weights to retain learned features.
  Pass input images through both models.
  Apply Global Average Pooling to reduce feature dimensions.
  Concatenate the outputs of both models to form a combined feature vector.
  Add a dropout layer to prevent overfitting.

Include a dense output layer with softmax activation for multi-class classification.

- **Hyperparameter Tuning**
The model was optimized with the following configurations:
  Optimizer: Adam optimizer with a learning rate of 0.0001.
  Early Stopping: Stop training if validation loss does not improve for 10 consecutive epochs.
  Learning Rate Reduction: Reduce the learning rate by half if validation loss stagnates for 5 epochs.
  Loss Function: Categorical cross-entropy.
  Evaluation Metric: Accuracy.

- **Model Training:**
The hybrid model was trained for 20 epochs using the augmented training dataset. Validation was performed simultaneously to monitor overfitting and track performance.
- **Evaluation and Visualization:**
Accuracy and loss were tracked over epochs for both training and validation sets. Graphs were plotted to visualize the trends of accuracy and loss, ensuring a clear understanding of the model's performance. [14]
- **Model Saving and Testing:**
The trained model was saved for future use. It was then tested on single input images to evaluate its ability to classify unseen data effectively.

HYBRID MODEL TRAINING PROCESS

- **Model Architecture Definition:**
A hybrid CNN model architecture was created by integrating NASNetLarge and EfficientNetB7, leveraging their pre-trained weights from ImageNet. This hybrid design harnesses the powerful feature extraction capabilities of both models. To ensure optimal performance, the top (fully connected) layers of each model were removed, retaining only the convolutional and pooling layers for feature extraction. Input images were passed through both models, and their outputs were combined using global average pooling. This concatenated feature vector was further processed by dense layers with dropout to reduce overfitting. The final layer used softmax activation for multi-class classification. This architecture enabled the model to focus on diverse and complex image features, improving classification accuracy on high variability datasets. [15]

- **Hybrid Deep Learning Model:**
The core of the proposed system is a hybrid deep learning model that integrates two high-performing architectures: NASNetLarge and EfficientNetB7. [16] The hybrid model is designed as follows:
  Feature Extraction: Both pre-trained models independently extract features from the input images.

In Figure 1, a detailed depiction of the architecture is presented, showcasing the components and intricacies of the models. The representation elucidates the Hybrid Model. This illustration offers a visual understanding of the approach employed in the study.

```
Model: "model_1"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_6 (InputLayer)           [(None, 331, 331, 3  0           []
                                )]

 NASNet (Functional)            (None, 11, 11, 4032  84916818    ['input_6[0][0]']
                                )

 efficientnetb7 (Functional)    (None, 11, 11, 2560  64097687    ['input_6[0][0]']
                                )

 global_average_pooling2d_2 (Gl  (None, 4032)        0           ['NASNet[0][0]']
 obalAveragePooling2D)

 global_average_pooling2d_3 (Gl  (None, 2560)        0           ['efficientnetb7[0][0]']
 obalAveragePooling2D)

 concatenate_9 (Concatenate)    (None, 6592)         0           ['global_average_pooling2d_2[0][0
                                                                 ]',
                                                                  'global_average_pooling2d_3[0][0
                                                                 ]']

 dropout_1 (Dropout)            (None, 6592)         0           ['concatenate_9[0][0]']

 dense_1 (Dense)                (None, 200)          1318600     ['dropout_1[0][0]']

==================================================================================================
Total params: 150,333,105
Trainable params: 1,318,600
Non-trainable params: 149,014,505
```

Fig. 1. Hybrid model details.

  Global Average Pooling: The extracted features are pooled to reduce dimensionality while preserving the most significant information.
  Feature Combination: The pooled features from both models are concatenated to create a rich feature representation.
  Dropout Layer: A dropout layer is introduced to mitigate overfitting. **Classification Layer:** A dense output layer with softmax activation predicts the probabilities of each bird species.
- **Training Process:**
The hybrid model is trained using the augmented training dataset. Loss Function: Categorical cross-entropy is used to minimize classification errors. Optimizer: Adam optimizer ensures efficient weight updates. Regularization: Early stopping and learning rate reduction on plateau are applied to prevent overfitting and improve training efficiency.

.

- **Evaluation and Testing:**
During training, the model's performance is evaluated using validation data, tracking metrics such as accuracy and loss. After training, the model is tested on unseen images to assess its classification accuracy and robustness. Visualizations, including graphs of accuracy and loss, are generated to analyze the training process.
- **Output:**
During training, the model's performance is evaluated using validation data, tracking metrics such as accuracy and loss. After training, the model is tested on unseen images to assess its classification accuracy and robustness. Visualizations, including graphs of accuracy and loss, are

generated to analyze the training process. [17]

IV. Results And Discussions

Software and Tools Used

Anaconda Navigator

- **Description:** Anaconda Navigator is a desktop graphical user interface included in the Anaconda distribution. It simplifies package management and deployment for data science projects. It allows users to easily create, manage, and switch between Python environments and install necessary libraries.
- **Role in Project:** Used to manage the Python environment and install essential libraries like TensorFlow, NumPy, and OpenCV for the bird classification system.

Python

- **Description:** Python is a high-level, interpreted programming language widely used for machine learning, deep learning, and data analysis. Its extensive library support and easy syntax make it ideal for implementing deep learning models.
- **Role in Project:** Used as the primary programming language for writing the code to preprocess data, build the hybrid model, train it, and evaluate its performance.

Jupyter Notebook

- **Description:** Jupyter Notebook is an open-source web-based interactive environment for running and documenting Python code. It allows users to combine live code, visualizations, and explanatory text in a single document.
- **Role in Project:** Used to write and execute code for the entire bird classification pipeline, from data augmentation to model building, training, and evaluation, while documenting the workflow effectively.

TensorFlow

- **Description:** TensorFlow is an open-source deep learning framework developed by Google. It provides pre-built models, libraries, and tools for building and deploying machine learning models.
- **Role in Project:** Used to build the hybrid model using NASNetLarge and EfficientNetB7 architectures, train the model, and manage hyperparameters and optimization processes.

OpenCV (cv2)

- **Description:** OpenCV (Open Source Computer Vision Library) is a popular library for computer vision tasks. It provides tools for image preprocessing, manipulation, and analysis.
- **Role in Project:** Used to preprocess images, such as resizing, and for loading single input images during testing.

Hardware and Software Requirements

1) Software Requirements:

- **Operating System:** Windows 10 or later / Ubuntu 18.04 or later / macOS.
- **Programming Language:** Python 3.7 or later.
- **Libraries/Tools:** TensorFlow 2.x, OpenCV, Jupyter Notebook, NumPy, Matplotlib.
- **Environment Management Tool:** Anaconda Navigator.

2) Hardware Requirements:

- **Processor:** Minimum: Intel Core i5 (or equivalent). Recommended: Intel Core i7 or AMD Ryzen 7 with 4+ cores for faster computation.
- **RAM:** Minimum: 8 GB. Recommended: 16 GB or more for efficient model training and data handling.
- **GPU:** Minimum: NVIDIA GTX 1050 or equivalent for basic model training. Recommended: NVIDIA RTX 3060 or higher for faster training and model deployment.
- **Storage:** Minimum: 50 GB free space. Recommended: SSD with at least 100 GB free space for storing datasets, models, and outputs.
- **Other Requirements:** Stable internet connection for downloading datasets and pre-trained models. Display with a minimum resolution of 1920x1080 for better visualization.

By integrating these software tools and ensuring the appropriate hardware setup, the proposed system achieves efficient bird classification with a seamless workflow from data preprocessing to deployment.

Results with Snapshots

Figure 2 The left plot shows the loss of the model over 50 epochs for both the training and validation datasets. Initially, the loss decreases sharply, indicating that the model is learning effectively. As the epochs progress, the training loss continues to decline, but the validation loss begins to plateau and diverge slightly.

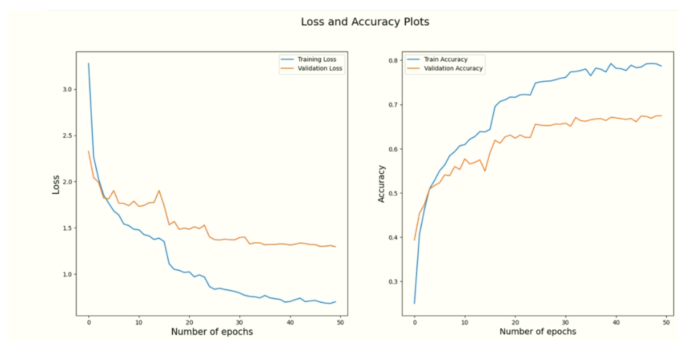

Fig. 2. Loss-Accuracy Curve

The right plot illustrates the accuracy of the model during training and validation. Both training and validation accuracy increase steadily at the start, demonstrating effective learning. However, after a certain point, the validation accuracy levels off at around 68-70%, while the training accuracy continues to rise, reaching approximately 80%.

Figure 3 The confusion matrix illustrates the model's performance across multiple classes, with the diagonal representing

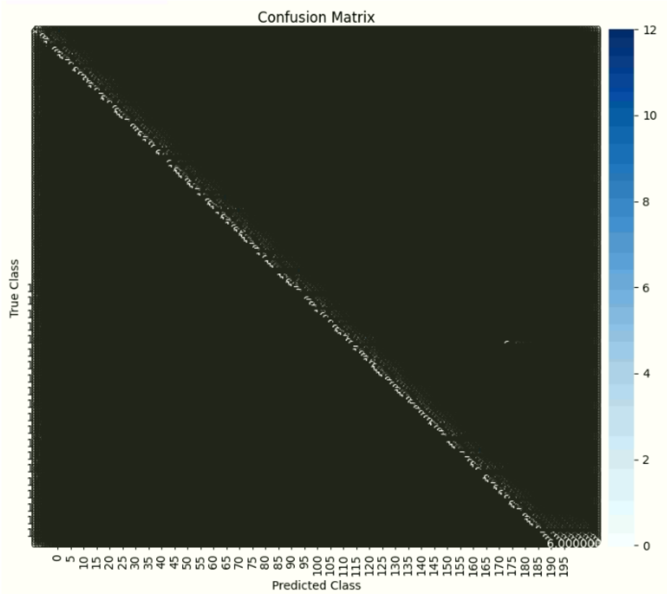

Fig. 3. Confusion Matrix

correct predictions and off-diagonal elements indicating misclassifications. A strong diagonal pattern, as seen here, signifies that the model generally classifies samples accurately, with minimal errors. The sparse misclassification suggests good performance but highlights specific areas where the model struggles to differentiate between certain classes. Addressing these misclassifications could further enhance accuracy, potentially through more data augmentation or refining the model architecture. The color intensity reflects the frequency of predictions, reinforcing the focus on diagonal dominance for successful classification.

Figure 4 The image shows feature maps extracted from an early convolutional layer of a neural network, visualizing how the model detects edges, textures, and patterns from the input image of a bird. Each grid cell represents activations from different filters, highlighting distinct features such as outlines, wing shapes, and contrast regions. These low-level features are essential for deeper layers to build more complex representations, aiding in accurate classification. The diversity in feature maps signifies the network's capacity to learn multiple perspectives from a single image.

Discussions

A. Interpretation of Results

The results obtained from the experiments provide valuable insights into the effectiveness of the applied techniques in enhancing classification precision and handling high variability within the CUB-200 bird species dataset. The interpretation of results is as follows:

- **Accuracy and Intra-Class Variability** The achieved accuracy metric serves as an essential indicator of overall model performance. However, in the context of high-variability datasets, accuracy alone may not fully reflect the model's capability. The model's ability to correctly classify bird species with subtle inter-species differences and variations in pose, lighting, and background is crucial for robust performance.

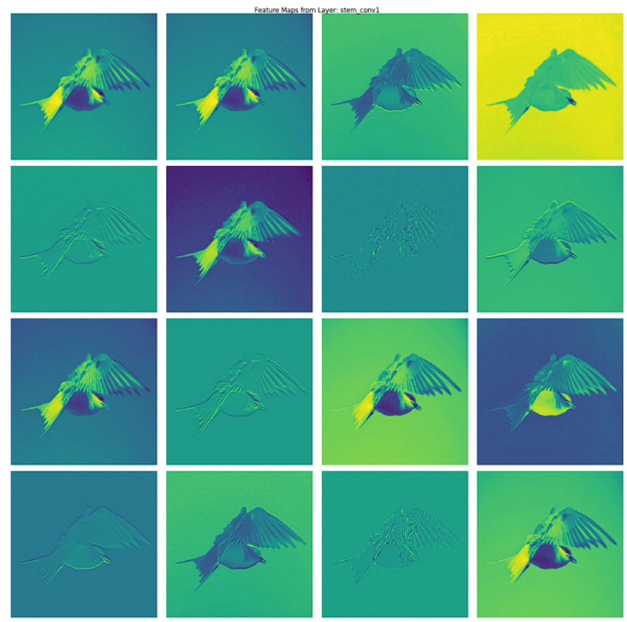

Fig. 4. Feature Map

- **Balancing Trade-offs** While improving recall ensures the model correctly identifies a broader range of bird species, it is equally important to manage trade-offs with precision to minimize false positives. Achieving a balance between precision and recall is vital to avoid misclassification of similar-looking bird species. The F1 score, which harmonizes both precision and recall, serves as a more comprehensive evaluation metric, reflecting the model's ability to generalize effectively across diverse bird species.

B. Limitations

Despite the promising results, several limitations were observed in the current implementation:

- **Generalization to New Datasets**
  The hybrid deep learning models, trained and evaluated on the CUB-200-2011 bird species dataset, may face challenges when applied to external datasets with different characteristics. Additional experimentation on varied datasets is necessary to validate the robustness and generalizability of the proposed framework.
- **Hyperparameter Optimization**
  While hyperparameters were fine-tuned for this project, further optimization using advanced techniques, such as Bayesian optimization or grid search, could potentially improve model performance and stability across epochs.
- **Incorporation of Additional Techniques**
  The hybrid model utilized NASNetLarge and Efficient-

NetB7 for feature extraction. Exploring additional architectures or techniques, such as self-supervised learning or fine-grained classification approaches, may enhance the model's capability to handle inter-class and intra-class variability.

- **Ethical and Practical Considerations**
The methodology assumes access to substantial computational resources and labeled data, which may not be feasible in all real-world scenarios. Ensuring the practical usability of the system and its alignment with ethical standards, including equitable dataset representation, remains a priority for future work.

C. Future Directions

The discussion of results and limitations provides a foundation for future research directions. Areas for further exploration include:

- **Exploring Base Models Separately**
Future work involves evaluating the performance of individual base models, such as NASNetLarge and EfficientNetB7, to establish detailed benchmarks and understand their independent strengths and limitations.
- **Integration of Multiple Models**
Expanding the hybrid approach by integrating two or more state-of-the-art models and comparing the performance of these integrated systems against the base models. This will help assess the efficacy of ensemble techniques in improving precision and robustness.
- **Hyperparameter Optimization**
Conducting extensive experiments with various hyperparameters to analyze their impact on model performance. Techniques such as grid search, random search, or Bayesian optimization can be used to identify optimal configurations.
- **Evaluation on Diverse Datasets**
Applying the proposed models to additional datasets with varied characteristics to validate their generalization capabilities and adaptability to different domains.
- **Impact of Parameter Variations**
Investigating how variations in parameters, such as layer depth, dropout rates, and learning rates, affect classification accuracy, training time, and overall model efficiency.

REFERENCES

[1] Yo-Ping Huang and Haobijam Basanta. Recognition of endemic bird species using deep learning models. *IEEE Access*, 2021.

[2] C. Chalmers, P. Fergus, S. Wich, et al. Removing human bottlenecks in bird classification using camera trap images and deep learning. *Remote Sens.*, 2023.

[3] B. Divya, M. A. V. Raju, and D. D. D. SURIBABU. Automatic bird species identification using deep learning. In *2022 International Conference on Data Science and Intelligent Computing (ICDSIC)*, 2022.

[4] Hung Nguyen, Sarah J Maclagan, Tu Dinh Nguyen, Thin Nguyen, Paul Flemons, Kylie Andrews, Euan G Ritchie, and Dinh Phung. Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring. In *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 40–49. IEEE, 2017.

[5] M. H. Abdulameer, Huda W. Ahmed, and Israa S. Ahmed. Bird image dataset classification using deep convolutional neural network algorithm. In *2022 International Conference on Data Science and Intelligent Computing (ICDSIC)*, 2022.

[6] Yo-Ping Huang and Haobijam Basanta. Bird image retrieval and recognition using a deep learning platform. *IEEE Access*, 2019.

[7] Sazida Binta Islam, Damian Valles, Toby J Hibbitts, Wade A Ryberg, Danielle K Walkup, and Michael RJ Forstner. Animal species recognition with deep convolutional neural networks from ecological camera trap images. *Animals*, 13(9):1526, 2023.

[8] Yunjie Liu, Evan Racah, Mr Prabhat, Joaquin Correa, Amir Khosrowshahi, David Lavers, Kenneth Kunkel, Michael Wehner, and William Collins. Application of deep convolutional neural networks for detecting extreme weather in climate datasets. May 2016.

[9] Munish Khanna, Law Kumar Singh, Shankar Thawkar, and Mayur Goyal. Planet: A robust deep convolutional neural network model for plant leaves disease recognition. *Multimedia Tools and Applications*, 83(2):4465–4517, 2024.

[10] Robert Logan, Brian G Williams, Maria Ferreira da Silva, Akash Indani, Nicolas Schcolnicov, Anjali Ganguly, and Sean J Miller. Deep convolutional neural networks with ensemble learning and generative adversarial networks for alzheimer's disease image data classification. *Frontiers in Aging Neuroscience*, 13:720226, 2021.

[11] Shuqin Wang. Design of real-time object recognition algorithm based on deep convolution neural network. pages 533–538, Dec 2023.

[12] P. Anusha and Kundurthi ManiSai. Bird species classification using deep learning. In *2022 International Conference on Intelligent Computing and Control Systems (ICICCSP)*, 2022.

[13] Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54, 2019.

[14] Pavithra Jampa, Uday Kiran Kolluri, et al. Image classification using deep learning models in computer vision. In *2024 International Conference on Intelligent Multimedia Computing (ICIMC)*, 2024.

[15] Abeer S Desuky and Sadiq Hussain. An improved hybrid approach for handling class imbalance problem. *Arabian Journal for Science and Engineering*, 46:3853–3864, 2021.

[16] S. Gopalakrishnan, B. Naveen, Lochan R. Krishnan, et al. Bird species detection using deep learning. *Indian Journal of Computer Science and Technology*, 2024.

[17] A. Varghese. Identification of bird species using a deep learning technology. 2021.