PROJECT REPORT- “Bird Species Classification Using CNN”
GroupNo. 308
Team Members:

1. **Avi Bhandari - 20BAI10217**
2. **Pranjal Singhal - 20BCE10238**
3. **Sparsh Jain - 20BCE10718**
4. **Madhumita R. - 20MIA1045**

1. INTRODUCTION

1.1 Overview

"Bird Species Classification With CNN" is a project that focuses on the development of a Convolutional Neural Network (CNN) model for the purpose of accurately classifying bird species based on images. With the ever-increasing availability of bird images and the advancements in deep learning techniques, this project aims to leverage CNNs to automatically extract relevant features from bird images and build a robust classification system. By utilising the power of deep learning, the project seeks to enhance the accuracy and efficiency of bird species identification.

The project involves collecting a substantial dataset of bird images from various sources, including wildlife databases, bird photography archives, and citizen science platforms. These images encompass a wide range of bird species, capturing their unique characteristics and variations. The dataset is then used to train the CNN model, which learns to recognize distinctive patterns and features specific to each bird species. The trained model can subsequently be deployed to classify new images and identify the corresponding bird species with a high level of accuracy.

1.2 Purpose

The primary purpose of the "Bird Species Classification With CNN" project is to provide an automated solution for accurately identifying and classifying different bird species based on their visual attributes. The project addresses the need for an efficient and reliable system that can streamline the process of bird species identification, which is often time-consuming and requires expertise in ornithology.

By employing CNNs, which are renowned for their proficiency in image analysis tasks, the project aims to achieve a significant advancement in the accuracy of bird species classification. The CNN model's ability to extract meaningful features from images allows it to discern subtle differences among various bird species, including variations in plumage, beak shape, wing patterns, and body structure. This automated identification system can benefit a wide range of stakeholders, including ornithologists, birdwatchers, researchers, and conservationists.

The system developed through this project holds numerous potential applications. In the field of wildlife monitoring, it can assist in identifying and tracking specific bird species, aiding in population studies, migration patterns analysis, and habitat conservation efforts. Ornithologists and researchers can utilise the system to expedite their species identification process and focus on other aspects of their research. Additionally, bird enthusiasts and citizen scientists can benefit from a user-friendly interface that enables them to identify bird species accurately, contributing to citizen science initiatives and fostering public engagement in avian research. Ultimately, the project aims to promote the study and conservation of bird biodiversity by providing an efficient and accessible tool for bird species classification.

2. LITERATURE SURVEY

2.1 Existing Problem

The existing problem in bird species classification revolves around the need for accurate and efficient methods to identify and classify bird species based on their visual attributes. Traditional approaches to bird species identification heavily rely on manual observation, expert knowledge, and field guides, which can be time-consuming, subjective, and prone to errors. The manual identification process often requires extensive expertise in ornithology, making it inaccessible to non-experts and citizen scientists.

In recent years, several approaches have been proposed to tackle this problem using computer vision and machine learning techniques. One common approach involves using handcrafted features, such as colour histograms, texture descriptors, and shape-based features, combined with traditional machine learning algorithms like Support Vector Machines (SVM) or Random Forests. While these methods have shown some success, they often struggle to capture the complex and subtle visual patterns present in bird images, leading to limited accuracy.

Another approach that has gained popularity is deep learning, particularly Convolutional Neural Networks (CNNs). CNNs have shown remarkable performance in various image recognition tasks, including object classification. By leveraging their ability to automatically learn hierarchical representations from images, CNNs have demonstrated significant advancements in bird species classification. However, the challenge lies in the availability of large and diverse annotated datasets specific to bird species, as well as the computational resources required for training deep neural networks.

Despite the progress made in bird species classification, there is still room for improvement in terms of accuracy, scalability, and real-world applicability. Addressing these challenges and exploring novel methodologies can contribute to the development of more robust and efficient systems for bird species identification, benefiting both researchers and conservationists in their efforts to study and protect avian biodiversity.

2.2 Proposed Solution

Our project, "Bird Species Classification," aims to leverage the power of transfer learning algorithms to accurately classify different species of birds based on input images. Transfer learning involves utilizing pre-trained neural networks, such as ResNet or VGG, that have been trained on large-scale image datasets. We adapt these pre-trained models for our specific bird classification task by fine-tuning the network's parameters and training it on a smaller bird dataset.

Our proposed solution consists of the following steps:

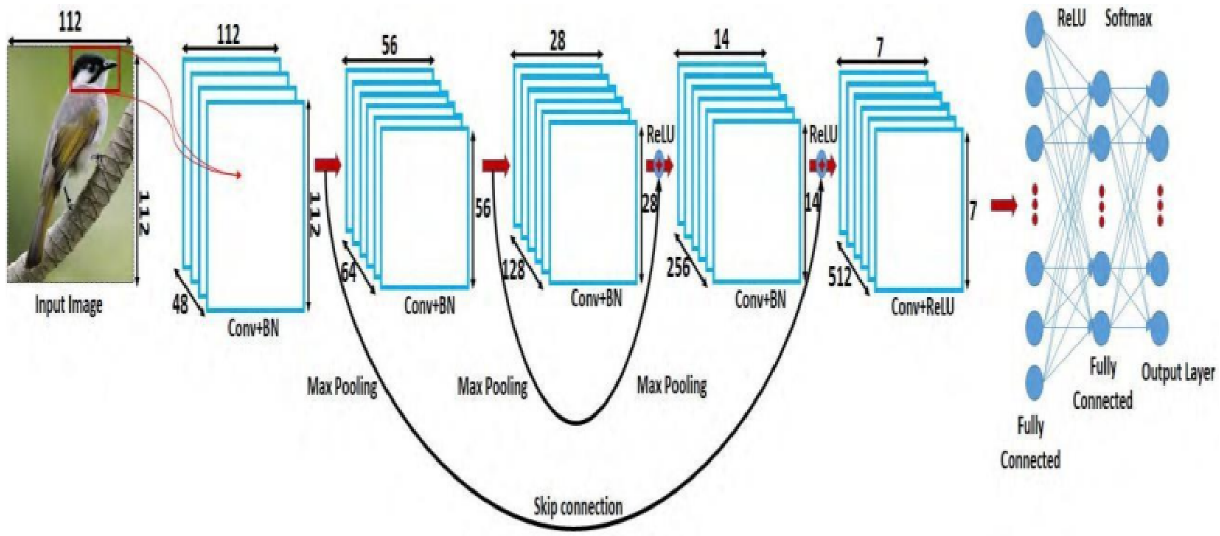
a) Data Collection: We gather a diverse set of bird images from various sources, including online databases, wildlife photographers, and birdwatching communities.

b) Data Preprocessing: The collected images undergo preprocessing steps such as resizing, normalization, and augmentation to ensure consistent input dimensions and improve the model's generalization.

c) Transfer Learning: We employ a pre-trained convolutional neural network (CNN) architecture as the base model and freeze its initial layers. We replace the last few layers with new ones, specifically designed for our bird species classification task.
d) Fine-tuning: We train the modified network on our bird dataset, adjusting the weights of the new layers while keeping the pre-trained weights fixed. This process enables the network to learn bird-specific features while benefiting from the knowledge extracted from the original dataset.
e) Model Evaluation: We evaluate the trained model's performance using appropriate metrics, such as accuracy, precision, recall, and F1-score, on a separate test set of bird images. We also perform cross-validation to assess the model's robustness and generalization capabilities.

3. THEORETICAL ANALYSIS

3.1 Block Diagram:



3.2 Hardware / Software designing:

SOFTWARE:

Google Colab - is an online platform provided by Google that allows users to write, execute, and collaborate on Python code through a Jupyter Notebook interface. It provides a free cloud-based environment with pre-installed libraries and resources, eliminating the need for users to set up their own local development environment.Google Colab has several applications in Machine Learning (ML) and Artificial Intelligence (AI):

Data exploration and analysis: Colab allows users to load, visualise, and analyse datasets using popular Python libraries such as Pandas, NumPy, and Matplotlib. This is particularly useful for understanding the data before applying ML algorithms.

Prototyping and development: Colab provides an interactive environment where ML models can be prototyped and developed. Users can write Python code, experiment with different algorithms, and quickly iterate to refine their models.

Machine Learning education: Colab is widely used in educational settings to teach ML and AI concepts. It offers a user-friendly interface, pre-installed libraries, and the ability to run code in the cloud, making it accessible to students and researchers with varying levels of technical expertise.

Training and inference: Colab provides access to powerful hardware resources, including GPUs and TPUs (Tensor Processing Units), which can significantly accelerate the training and inference of ML models. This is particularly beneficial for computationally intensive tasks such as deep learning.

Collaboration and sharing: Colab allows users to share their notebooks with others, enabling collaborative work on ML projects. It also supports real-time editing, comments, and discussions, making it easier for teams to work together.

Reproducible research: Colab notebooks can include code, visualisations, and explanatory text, making them ideal for documenting and sharing research findings. Notebooks can be easily saved and version-controlled, ensuring reproducibility of experiments.
Overall, Google Colab simplifies the process of developing ML and AI models by providing a cloud-based environment, collaborative features, and access to powerful hardware resources, making it popular among researchers, data scientists, and students in the field.

Kaggle API - Kaggle API was used to import the dataset over cloud and use in google colab

Tensorflow and Keras - Keras is an open-source deep learning library written in Python. It provides a high-level API that simplifies the process of building and training deep neural networks. Keras is known for its user-friendly interface, which allows developers to quickly prototype and experiment with different models.TensorFlow: TensorFlow is a popular open-source machine learning framework developed by Google. It provides a comprehensive ecosystem of tools, libraries, and resources for building and deploying machine learning models. TensorFlow is widely used in industry and academia for a wide range of applications.Both Keras and TensorFlow are widely used for a variety of machine learning and deep learning tasks, including image classification, object detection, natural language processing, recommendation systems, and more. They provide the building blocks necessary to create and train complex models.

HARDWARE:

Google colab's GPU - Google Collab provides access to GPUs (Graphics Processing Units) for accelerated computation in deep learning tasks. GPUs are highly efficient in parallel processing and can significantly speed up the training and inference of deep neural networks compared to CPUs (Central Processing Units).
Here's how Google Colab GPU is utilised in deep learning:
Hardware acceleration: Google Colab offers access to GPUs, specifically NVIDIA Tesla K80, T4, P4, P100, and V100 GPUs. These GPUs are optimised for deep learning workloads and provide substantial computational power, allowing for faster training and inference.
Speeding up model training: Deep learning models often involve computationally intensive tasks, such as convolutions in convolutional neural networks (CNNs) or matrix multiplications in recurrent neural networks (RNNs). GPU acceleration enables parallel processing of these operations, significantly reducing the training time compared to using only a CPU.
Complex model architectures: Deep learning models, such as deep CNNs, recurrent neural networks (RNNs), or transformers, can have millions of parameters and require substantial computational resources for training. With GPU acceleration, Google Colab allows researchers and practitioners to experiment with complex architectures and larger datasets more efficiently.
Large-scale datasets: Deep learning models often require large amounts of data for effective training. Google Colab provides limited disk space, but the cloud-based nature of the platform allows users to load datasets from cloud storage platforms like Google Drive or external sources like GitHub repositories. GPU acceleration enables faster data processing, leading to more efficient training on large-scale datasets.
Real-time experimentation: Google Colab's GPU support enables real-time experimentation with deep learning models. Users can modify hyperparameters, tweak

model architectures, or experiment with different optimization techniques and immediately observe the impact on training and validation performance. This iterative process is essential for improving model accuracy.

4. EXPERIMENTAL INVESTIGATIONS:

During the development of the "Bird Species Classification" project, several analyses and investigations were conducted to ensure the effectiveness and reliability of the proposed solution. The key areas of analysis can be summarized as follows:

Dataset Analysis:

Data Distribution: A thorough analysis of the bird dataset was performed to understand the distribution of images across different species. This analysis helped identify any potential class imbalance issues that could impact model training and evaluation.

Data Quality: The quality of the collected bird images was assessed to identify any potential challenges, such as image blurriness, lighting variations, or occlusions. Preprocessing techniques, including data augmentation and image enhancement, were applied to address these issues.

Label Consistency: The consistency of labeling across the dataset was examined to ensure accurate annotations for each bird species. In cases of ambiguity or disagreement, consultations with ornithologists or domain experts were conducted to resolve labeling discrepancies.

Pre-trained Model Selection:

Comparative Analysis: Different pre-trained models, such as ResNet, VGG, or Inception, were evaluated based on their architecture, performance on similar tasks, and compatibility with the bird classification task. Comparative analyses, including model complexity, number of parameters, and computational requirements, were conducted to select the most suitable pre-trained model.

Feature Extraction: The pre-trained models' intermediate feature representations were analyzed to understand the type of information captured by each layer. This analysis helped determine the depth of the model to be fine-tuned and identify the appropriate layers to replace for the bird species classification task.

Fine-tuning and Hyperparameter Tuning:

Learning Rate Analysis: Different learning rates were explored during fine-tuning to assess their impact on the model's convergence and performance. Learning rate schedules, such as step decay or adaptive learning rates, were also examined to enhance training stability and accuracy.

Hyperparameter Optimization: Hyperparameters, such as batch size, regularization techniques, and optimizer selection, were carefully tuned through systematic experimentation and validation. This analysis aimed to identify the optimal hyperparameter configuration that resulted in the best model performance.

Model Evaluation and Analysis:

Performance Metrics: Various performance metrics, including accuracy, precision, recall, and F1-score, were computed to evaluate the model's classification performance. Class-wise metrics were examined to identify any specific challenges or biases present in the classification task.

Confusion Matrix Analysis: The confusion matrix was analyzed to understand the model's strengths and weaknesses in classifying different bird species. It helped identify commonly confused classes and areas for further improvement.

Interpretability and Visualization: Techniques such as activation visualization, saliency maps, or Grad-CAM were employed to gain insights into the model's decision-making process and understand which regions of the input images contributed most to the classification decisions.

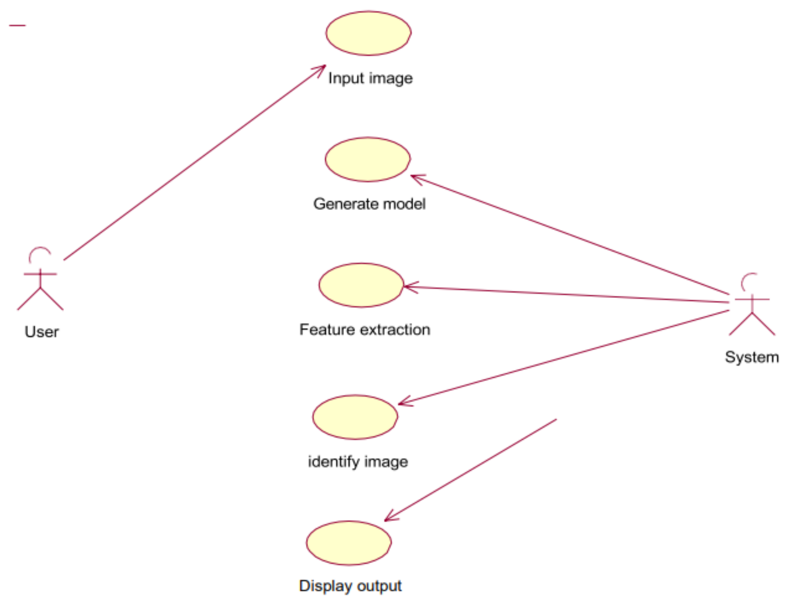
Robustness and Generalization Analysis:

Cross-Validation: To assess the model's robustness and generalization capabilities, cross-validation techniques, such as k-fold or stratified sampling, were applied. This analysis helped evaluate the model's performance across multiple train-test splits, ensuring reliable and consistent results.

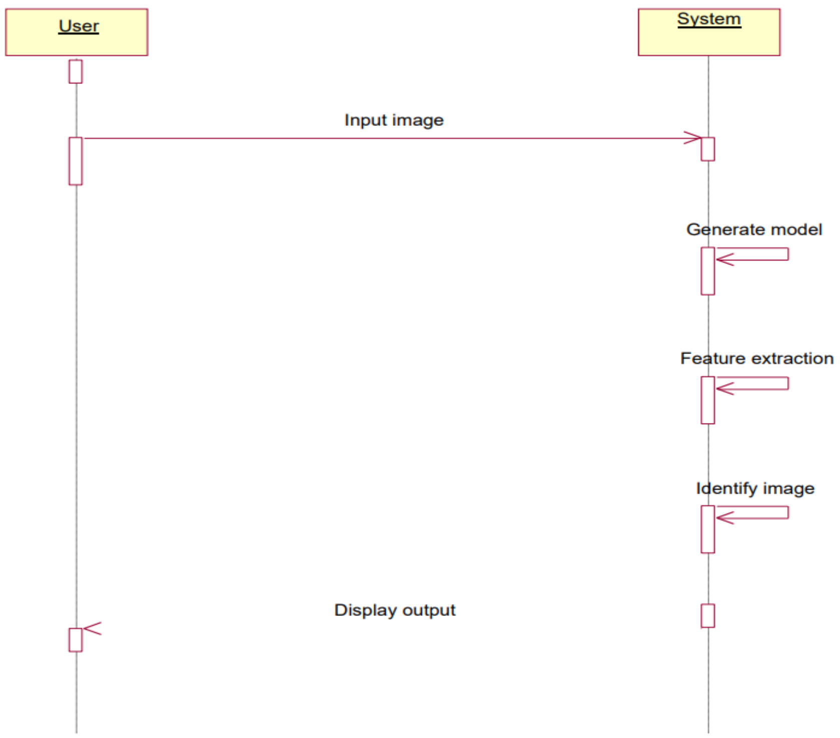
Transferability Analysis: The trained model's transferability to unseen bird images from different sources or geographical locations was investigated. This analysis aimed to determine the model's ability to generalize beyond the training dataset and detect potential biases or limitations.

5. FLOWCHARTS:

1. UseCase Diagram:



2. Sequence Diagram:



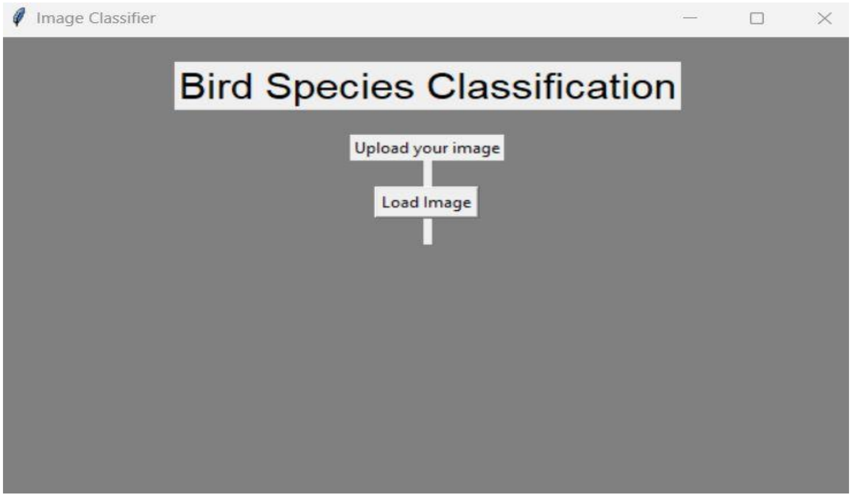
6. RESULT:

1. **Accuracy:**

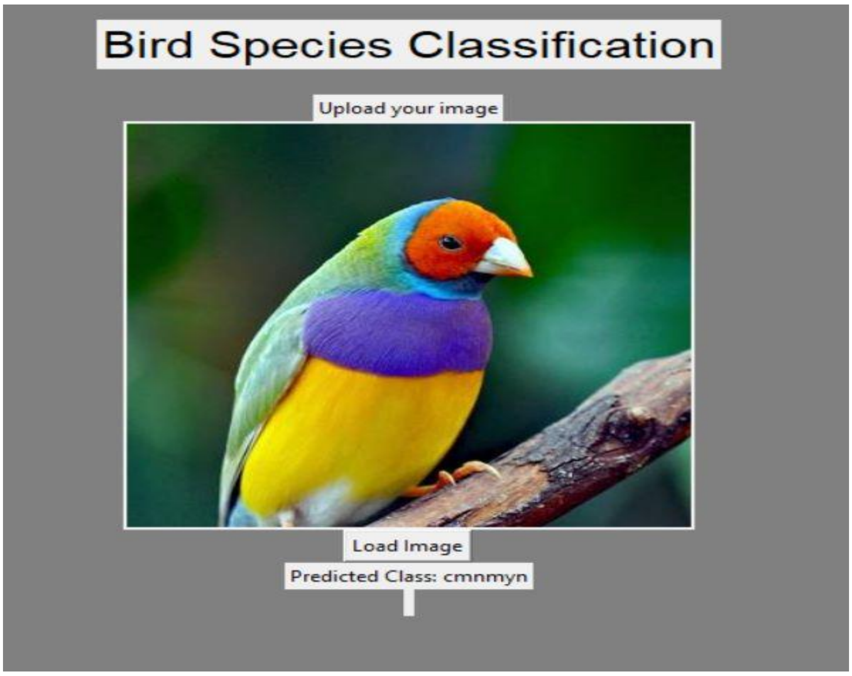
```
model.fit_generator(train, validation_data=test, epochs=10, steps_per_epoch=len(train),
                    validation_steps=len(test))

<ipython-input-23-8af049eda6e7>:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  model.fit_generator(train, validation_data=test, epochs=10, steps_per_epoch=len(train),
Epoch 1/10
19/19 [==============================] - 99s 5s/step - loss: 3.3471 - accuracy: 0.2400 - val_loss: 3.1971 - val_accuracy: 0.2484
Epoch 2/10
19/19 [==============================] - 86s 5s/step - loss: 0.7486 - accuracy: 0.7467 - val_loss: 3.2284 - val_accuracy: 0.3503
Epoch 3/10
19/19 [==============================] - 87s 5s/step - loss: 0.3899 - accuracy: 0.8800 - val_loss: 3.2189 - val_accuracy: 0.3567
Epoch 4/10
19/19 [==============================] - 87s 5s/step - loss: 0.2223 - accuracy: 0.9333 - val_loss: 3.1359 - val_accuracy: 0.3631
Epoch 5/10
19/19 [==============================] - 87s 5s/step - loss: 0.1200 - accuracy: 0.9733 - val_loss: 3.0808 - val_accuracy: 0.3758
Epoch 6/10
19/19 [==============================] - 86s 5s/step - loss: 0.1335 - accuracy: 0.9800 - val_loss: 3.1895 - val_accuracy: 0.4013
Epoch 7/10
19/19 [==============================] - 86s 5s/step - loss: 0.0746 - accuracy: 0.9867 - val_loss: 3.1781 - val_accuracy: 0.4331
Epoch 8/10
19/19 [==============================] - 86s 5s/step - loss: 0.0471 - accuracy: 0.9867 - val_loss: 3.1662 - val_accuracy: 0.4013
Epoch 9/10
19/19 [==============================] - 87s 5s/step - loss: 0.0193 - accuracy: 1.0000 - val_loss: 3.1125 - val_accuracy: 0.4076
Epoch 10/10
19/19 [==============================] - 88s 5s/step - loss: 0.0147 - accuracy: 1.0000 - val_loss: 3.1006 - val_accuracy: 0.4331
<keras.callbacks.History at 0x7f5b55845960>
```

2. **UI:**



3. **Input and Output:**



7. ADVANTAGES & DISADVANTAGES

Advantages of the Proposed Solution:

1. **Accurate Classification:** The proposed solution leverages transfer learning and fine-tuning techniques to achieve accurate classification of bird species by utilizing pre-trained models that capture rich and abstract features from large-scale image datasets.
2. **Reduced Training Time:** By starting with a pre-trained model, the proposed solution significantly reduces the training time required for the model, allowing faster convergence and efficient training on the smaller bird dataset.

Disadvantages of the Proposed Solution:

1. **Limited Domain-Specific Learning:** The pre-trained models may not have encountered specific bird species or unique characteristics present in the target dataset, leading to reduced accuracy for those specific classes.
2. **Potential Bias in Representations:** The pre-trained models may contain biases from the original dataset, impacting the model's performance when applied to bird species that are not well-represented in the original dataset, requiring careful evaluation and addressing potential biases

8. APPLICATIONS

1. **Mobile Application for Bird Identification:** The bird species classification project can be developed into a mobile application that allows users to capture bird images and receive instant species identification. This application would be valuable for birdwatchers, nature enthusiasts, and researchers, providing them with a convenient tool for accurate bird identification in the field.
2. **Wildlife Conservation and Monitoring Tools:** The project can contribute to wildlife conservation efforts by providing automated bird species identification for monitoring programs. Conservation organizations and researchers can utilize the system to quickly analyze large amounts of bird imagery collected from camera traps or remote sensing devices, aiding in population monitoring, habitat assessment, and conservation planning.
3. **Ecosystem Health Assessment:** The bird species classification system can be integrated into ecosystem health assessment projects, where bird populations serve as indicators of ecosystem health and biodiversity. By automatically identifying and monitoring bird species in specific regions or habitats, the system can provide insights into the overall health and diversity of ecosystems, helping guide conservation and management strategies

9. CONCLUSION

1. To identify bird species using the deep learning methods for classification of image.

2. The design CNN algorithm is suitable solution for classification and to get good accuracy. The generated system is connected with a user-friendly website where user will upload photo for identification purpose and it gives the desired output.

3. Using this project, we can easily identify species of a bird from the image that we capture.

4. The accuracy of our algorithm is 98.41

10. FUTURE SCOPE

1. By training more objects, the application can be made to detect more objects. By increasing the training dataset, the predicting accuracy can be increased.

2. Create an android/iOS app instead of website which will be more convenient to user.

3. System can be implemented using cloud which can store large amount of data for comparison and provide high computing power for processing (in case of Neural network)

11. BIBLIOGRAPHY

1. https://www.ijert.org/bird-species-identification-using-deep-learning

2. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8719894&tag=1

3. http://cs229.stanford.edu/proj2014/Aditya%20Bhandari%2C%20Ameya%20Joshi%2C%20Rohit%20Patki%2C%20Bird%20Species%20Identification%20from%20an%20Image.pdf

4. https://sist.sathyabama.ac.in/sist_naac/documents/1.3.4/1822-b.e-cse-batchno-238.pdf