IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 12, No. 10, October, 2025, pp. 3563-3576
Journal homepage: http://iieta.org/journals/mmep

Genetic Algorithm-Based Ensemble of Deep Architecture Classifiers for Performance Enhancement of Bird Species Classification

B. S. Chandrashekar[1*], H. S. Nagendraswamy[1], M. P. Pavan Kumar[2]

[1] Department of Studies in Computer Science, University of Mysore, Mysuru 570006, India
[2] Information Science and Engineering, Jawaharlal Nehru National College of Engineering, Shivamogga 577204, India

Corresponding Author Email: coderchandru@gmail.com

https://doi.org/10.18280/mmep.121022

Received: 24 June 2025
Revised: 10 September 2025
Accepted: 22 September 2025
Available online: 31 October 2025

Keywords:
genetic algorithm, deep architecture classifiers, performance enhancement, bird classification, probability score, Vision Transformer, EfficientNetV2L, classifier ensemble

ABSTRACT

Computer vision-based approaches to bird species classification have been extensively studied over the past few decades. Manually classifying bird species based on their appearance in real-time is a complex and time-consuming task. However, advancements in image-capturing technology and the integration of machine learning techniques have made it possible to automatically classify bird species and assist ecologists in studying bird population trends, identifying endangered species, and assessing the impacts of habitat loss, climate change, and pollution on bird populations. Considering this, the present research proposes an approach that integrates multiple deep architecture classifiers optimized via a genetic algorithm to enhance bird species classification performance. Eight state-of-the-art deep architecture classifiers, namely Vision Transformer (ViT), EfficientNetV2L, VGG16, VGG19, Xception, Inception, MobileNetV2, and ResNet152, were employed. Experiments were conducted using the Indian bird dataset. The genetic algorithm is employed to determine the best combining rule and select an ensemble of deep architectures with the highest fitness. The proposed ensemble method achieved a classification accuracy of 97% using a mean-based ensemble of VGG19, EfficientNetV2L, and ViT. The findings indicate that the proposed ensemble approach outperforms existing appearance-based bird species classification methods.

1. INTRODUCTION

Advancements in the field of science and technology in recent years have motivated researchers and technocrats to consider addressing some of the very important and real-time tasks that have a significant impact on society and natural ecosystems. One such task is the classification of bird species in real-time environments. Birds can be classified based on their appearance and sound. Many ecologists and environmentalists who are quite familiar with birds can perform classification based on sound and appearance, but this is a challenging and cumbersome task when the number of species is large. Furthermore, intra-class variations and between-class similarities among the species pose many more challenges in accurately classifying bird species. In this context, in the past few decades, many attempts have been made to propose efficient and effective bird species classification techniques. A review of the literature reveals that there is scope to improve the classification accuracy and address many challenges, such as birds in complex backgrounds, occlusion, different poses, images with noise, and poor intensity. The capabilities of existing approaches can be effectively integrated to build a more robust and effective model that can work in real time and address the challenges mentioned above.

In view of this, an attempt was made to propose a genetic algorithm-based ensemble of deep architecture classifiers to enhance the performance of classification. The proposed ensemble approach explores the best deep architecture classifiers and the best combining rules for integration. A database consisting of 41 bird species was created to evaluate the performance of the proposed ensemble classification approach. A total of 6150 images were considered in our experimental analysis.

The remainder of this paper is organized as follows. Section 2 presents a review of recent literature. Section 3 presents the proposed methodology. Section 4 presents the details of the experiments conducted and the results obtained on the dataset created as a part of this work, followed by conclusions and a few important references.

The key contributions of this work are listed as follows:

- Proposed a genetic algorithm-based ensemble of deep architecture classifiers for effective classification of bird species.
- Suggested a novel way of applying crossover and mutation operators for the genetic algorithm.
- Created a considerably large database of bird images to study the performance of the proposed methodology.

2. LITERATURE REVIEW

Several researchers have attempted to develop techniques for the effective classification of bird species. A review of these key contributions is essential for identifying research gaps, addressing the limitations of existing systems, and proposing methodologies to enhance efficiency and effectiveness. In light of this, the following paragraphs summarize the significant works related to appearance-based bird species classification.

2.1 Convolutional Neural Network and Vision Transformer-based approaches

In this section, Convolutional Neural Network (CNN) and Vision Transformer (ViT)-based approaches for bird species classification are discussed. The following paragraphs present a brief description and critical analysis of these approaches.

Chen et al. [1] proposed a Hierarchical Gated Network (HGNet) for fine-grained visual recognition. HGNet uses the interconnections among hierarchical classes. Long Short-Term Memory (LSTM) like mechanism is used to propagate dependencies among classes, which boosts classification accuracy. Experiments were conducted on datasets such as Stanford Dogs, CUB-200-2011, Aircraft, NABirds, iNaturalists, DeepFashion2, and DeepFashion. The proposed methodology achieved 88.7%, 88%, 92.8%, 86.4%, 78.2%, 58.5%, and 68.4% accuracy for the Stanford Dogs, CUB-200-2011, Aircraft, NABirds, iNaturalists, DeepFashion2, and DeepFashion datasets, respectively.

Branson et al. [2] proposed a classification approach based on pose-normalized Deep Convolutional Neural Networks (DCNNs). The architecture estimates the pose of a bird to compute the local image features for classification. A graph-based clustering algorithm and higher-order geometric warping functions were used to estimate the normalized bird pose. Experiments conducted on the CUB-200-2011 dataset showed an accuracy of 75.7%.

Jaderberg et al. [3] proposed a spatial transformer network by introducing a learnable module that could be integrated into any CNN architecture with very little modification. When incorporated, this module equips the CNN with the ability to handle image transformations, such as translation, scaling, and rotation. The experiments conducted on the CUB-200-2011 dataset achieved a classification accuracy of 84.1%.

Manna et al. [4] proposed various convolutional neural networks, namely InceptionV3, DenseNet 201, MobilenetV2, and ResNet152V2, for bird species classification based on images. The model was trained on 58388 images and tested on 2000 images belonging to 400 species. According to the study, ResNet152V2 achieved the highest accuracy of 95.45% and a loss of 0.8835. Similarly, DenseNet achieved a classification accuracy of 95.0% with a loss of 0.6845.

Zhang et al. [5] proposed a part-based CNN for fine-grained category detection to explicitly identify significant differences in the appearance of specific object parts. Semantic part localization was utilized for fine-grained categorization, with bounding box annotations considered during testing to address the object annotation challenges. The model leveraged convolutional features obtained through bottom-up region proposals, learning both whole-object detection and part detection while enforcing geometric constraints between the foreground and background of the interest. This approach enables fine-grained category detection using post-normalized representations. Experiments on the CUB-200-2011 dataset demonstrate a performance of 76%.

Huang et al. [6] proposed a recognition of endemic birds using deep-learning models. A transfer learning-based method using Inception-ResNetV2 was used, which outperformed all the other methods, namely InceptionV3, ResNet101, MobileNetV2, Xception, and ResNet101. A five-fold cross-validation was used to review the results. A total of 790 images of birds, consisting of 29 species, were used to evaluate the performance of the model. The proposed model achieved accuracies of 100% for bird identification and 98.39% for bird classification.

Wei et al. [7] proposed a mask CNN without fully connected layers, which localizes parts and selects descriptors for fine-grained bird species categorization. The fully convolutional network locates discriminative parts and generates weighted object masks. A three-stem-masked CNN model was used to build the descriptors. The experiment was conducted on the CUB-200-2011 and Birdsnap datasets. The proposed model achieves 85.7% of accuracy for the CUB-200-2011 dataset and 77.3% of accuracy for the Birdsnap dataset, respectively.

Kumar and Kondaveeti [8] proposed a bird species recognition using a hybrid hyperparameter optimization scheme (HHOS). A few selected models were trained in the HHOS, which enhanced the classification performance. The results showed that EfficientNetB0 achieved superior performance compared to the other classification models, with the highest accuracy of 99.12%. The authors increased the size of the CUB-200-2011 dataset from 11788 to 40000 by adding 29,000 more images, which are the augmented versions of the original dataset.

Ngo et al. [9] proposed a survey of different deep architecture classifiers, such as InceptionV3 and EfficientNetB4. The model was evaluated on a wide range of datasets, including CUB -200-2011, Kaggle 325 bird species, and Kaggle-510 bird species, and a self-generated dataset (100 bird species from Malaysia) was used in this study. The EfficientNetB4 model outperformed the models in terms of classification performance. For the CUB-200-2011 dataset, the result was 74% for the EfficientNetB4 model.

Bold et al. [10] proposed a multiple kernel learning framework for bird species classification using a CNN. A multiple kernel learning technique was used to combine features from audio and video data. The CUB-200-2011 dataset and audio data were combined for classification. 78.15% was the highest performance achieved using the proposed methodology.

Discriminative features for bird species classification were proposed by Pang et al. [11]. First, the images were cropped based on patches. The patches are used to form codebooks, and finally, the codebooks are used to generate intermediate features based on the sparse coding algorithm. Intermediate features were concatenated to form the final feature representation, which was used for training and classification. The CUB-200-2011 dataset was used in this study. The proposed model has achieved 64.6% of performance.

CNN works best in processing spatial patterns, while ViT works best in processing global context and long-range dependencies. Limitations of the ViT are high computational cost, low inductive bias, and a need for huge amounts of data. Whereas limitations of the CNN are high inductive bias, lower computational cost, and a weakness in handling global context.

CNN-based models utilize convolutional kernels that have

a local receptive field, meaning they process only a small window of pixels at a time. While stacking multiple layers allows the network to eventually see the entire image, its global understanding remains inefficient and implicit. This limitation can be addressed by transformer-based methods, which use a self-attention mechanism for a more holistic view.

However, transformer models are extremely dependent on large amounts of data and require high computational power. Consequently, the choice between a CNN and a ViT involves a trade-off: CNNs offer greater efficiency and lower computational demands, while ViTs deliver higher performance and superior holistic modelling, which requires adequate data and significant computational resources, as shown in Table 1.

Table 1. Literature review on Convolutional Neural Network and Vision Transformer-based approaches

Refs.	Architecture	Dataset	Accuracy	Analysis	Limitation
[1]	Hierarchical gate network for fine-grained visual recognition	Stanford Dogs, CUB-200-2011, Aircraft NABirds, and iNaturalists. DeepFashion2 and DeepFashion.	88.7%, 88%, 92.8%, 86.4%, 78.2%, 58.5%, and 68.4% accuracy for the Stanford Dogs, CUB-200-2011, Aircraft NABirds, iNaturalists, DeepFashion2, and DeepFashion datasets, respectively.	Interconnection among hierarchical classes and LSTM is used to propagate dependencies.	The ViT-based approaches are computationally expensive, low inductive bias, and require huge amounts of data. Whereas the CNN-based approaches have high inductive bias and are weak in handling global context.
[2]	Classification approach based on pose-normalized DCNNs	CUB-200-2011	75.7%	With DCNN, a graph-based clustering algorithm and higher-order geometric warping functions were used to estimate the normalized bird pose.	
[3]	Spatial transformer networks	CUB-200-2011	84.1%	A learnable mod named spatial transformer is used to handle image transformation, scaling, and rotation.	
[4]	Bird image classification using convolutional neural network transfer learning architectures	The model was trained on 58388 images and tested on 2000 images belonging to 400 species (own dataset).	ResNet152V2 achieved the highest accuracy of 95.45% and a loss of 0.8835. Similarly, DenseNet achieved a classification accuracy of 95.0% with a loss of 0.6845.	Various deep architecture classifiers are compared.	
[5]	Part-based R-CNNs for fine-grained category detection	CUB-200-2011	76%	The model leveraged convolutional features obtained through bottom-up region proposals, learning both whole-object detection and part detection while enforcing geometric constraints between the foreground and background of the interest.	
[6]	Recognition of endemic bird species using deep learning models	Own dataset	100% for bird identification and 98.39% for bird classification.	Various deep architecture classifiers' performance is compared.	
[7]	Bird species recognition using transfer learning with a hybrid hyperparameter optimization scheme (HHOS)	CUB-200-2011	The model achieved the highest accuracy of 99.12%. The authors increased the size of the CUB-200-2011 dataset from 11788 to 40000 by adding 29,000 more images, which are the augmented versions of the original dataset.	Augmentation is used to build more robust models.	
[8]	Bird species identification using deep learning on a GPU platform	CUB-200-2011	Google Net was used for this purpose, which performed with 88% classification accuracy.	Deep architecture classifiers named GooLeNet were used for this purpose, which proves that any deep learning model can be used.	
[9]	Mask-CNN: Localizing parts and selecting descriptors for fine-grained image recognition	CUB-200-2011 and Birdsnap datasets	85% of accuracy for the CUB-200-2011 dataset and 77.3% of accuracy for the Birdsnap dataset, respectively.	This study proves that the EfficientNet model is the best model for bird species classification.	
[10]	Bird species recognition system with a fine-tuned model	CUB-200-2011	74%	Fusion of audio and video enhances classification performance.	
[11]	Bird species classification with audio-visual data using CNN and multiple kernel learning	CUB-200-2011	78.15%	Images were cropped based on patches. The patches are used to form codebooks, and finally, the codebooks are used to generate intermediate features based on the sparse coding algorithm.	

Table 2. Literature review of genetic algorithm-based approaches

Ref.	Architecture	Dataset	Accuracy	Analysis	Limitation
[12]	A new ensemble learning methodology based on the hybridization of classifier ensemble selection approaches.	14 datasets	The best-performing chromosome was achieved	Ensemble learning-based genetic algorithm estimates the best performing chromosome based on a trade-off between accuracy and diversity.	Limitations of the genetic algorithm-based approaches are computationally expensive for training the model at each iteration, no guarantee of optimality, problem-specific parameter tuning, premature convergence, and are not suited for all kinds of solutions.
[13]	Optimizing DenseNet121 for waste classification using genetic algorithm-based down-sampling and data augmentation.	Own dataset	97%	To obtain the best hyperparameter configuration, multiple scenarios were tested with augmentation and different image dimensions.	

2.2 Genetic algorithm-based approaches

Genetic algorithm-based approaches for bird species classification were discussed in studies [12, 13]. The following paragraphs present a brief description and critical analysis of these approaches.

Mousavi and Eftekhari [12] proposed a technique for image classification based on the hybridization of classifier ensemble approaches. The model was evaluated using both static and dynamic ensemble methodologies, leveraging NSGA-II as a multi-objective genetic algorithm to optimize both error and diversity for selecting the best classifier ensemble. A Pareto optimal solution is suggested to balance the trade-off between these objectives. The experiments were conducted using six combined rules and 46 individual classifiers. The proposed approach outperformed all other ensemble methods across the 14 datasets in terms of classification accuracy.

Dharmawan et al. [13] proposed a novel waste classification approach that incorporates genetic algorithm hyperparameter optimization with data augmentation and data down-sampling. To obtain the best hyperparameter configuration, multiple scenarios were tested with augmentation and different image dimensions. This configuration enhances the accuracy from 94% to 97%.

Advantages of the genetic algorithm are that it supports global optimization, handles nondifferentiable and complex problems, is flexible and versatile, provides a set of good solutions, and is intuitive and inspired by nature.

Limitations of the genetic algorithm-based approaches are computationally expensive for training the model at each iteration, no guarantee of optimality, problem-specific parameter tuning, premature convergence, and are not suited for all kinds of solutions, as shown in Table 2.

2.3 Machine learning-based approaches

Machine learning-based approaches for bird species classification are discussed in studies [14-19]. The following paragraphs present a brief description and critical analysis of these approaches.

Kong and Fowlkes [14] proposed a low-rank bilinear pooling method for fine-grained classification. The study revealed that a high-dimensional bilinear feature vector, created by pooling the results of second-order local feature statistics, outperformed contemporary methods across various fine-grained bird classification tasks. To address the complexity of high-dimensional feature spaces, this study proposes covariance features and applies them to a low-rank bilinear classifier. The model was further compressed using a classifier co-decomposition approach, where a collection of bilinear classifiers was factorized into a common factor and compact per-class terms. This co-decomposition was implemented using two convolutional layers and was trained in an end-to-end architecture. Additionally, an effective initialization strategy was suggested to avoid the explicit training and factorization of larger bilinear classifiers. Several experiments were conducted on various public datasets to evaluate the efficacy of the proposed methodology, and the results demonstrated a superior performance. The experiment conducted on the CUB-200-2011 dataset achieved the highest accuracy of 84.21%.

Rai et al. [15] proposed various machine learning techniques for bird species classification. In this study, a CNN model was used for feature extraction, whereas Naive Bayes (NB), Support Vector Machine (SVM), Random Forest (RF), Linear Discriminant Analysis (LDA), Decision Tree (DT), and K-Nearest Neighbors (KNN), and Logistic Regression were employed for classification. Trail experiments were conducted on the CUB-200-2011 dataset, and it was reported that the KNN classifier achieved the highest classification accuracy of 33%. These studies highlight the progress made in appearance-based bird species classification and underscore the need for further improvements, particularly in handling complex pose variations and enhancing classification accuracy in real-world settings.

Yang et al. [16] proposed an improved transfer learning methodology for the classification of protected Indonesian birds based on images. In this study, batch normalization dropout fully connected (BNDFC) layers were proposed, which can be incorporated into any CNN network to further enhance classification accuracy. This study was conducted using an Indonesian bird dataset. The experiment was conducted using MobilenetV2 CNN, and on average, the performance was enhanced by 19.88% accuracy, 24.43% F-measure, 17.93% G-mean, 23.41% of sensitivity, and 18.76% precision. The highest accuracy achieved was 88.07% for the validation set.

Naranchimeg et al. [17] proposed a multi-model bird species classification for using audio-visual data. CUB-200-2011 was used to build the visual data module, and originally collected audio data were used for building the audio module.

The overall classification performance was enhanced by combining audio and video data using a CNN. The model achieved a classification accuracy of 78.9%.

Yang et al. [18] proposed an automatic bird image classification with feature enhancement and contrastive learning. The proposed methodology includes multiscale feature fusion to extract information at different scales. An attention feature enhancement module was integrated to address occlusion and noise. Furthermore, the Siamese network was used to compare two images belonging to the same class and different classes. The CUB-200-2011 dataset

was used in the study. With only 5% of the training data, the model performed fairly well, with a recognition accuracy of 65.2%. Limitations of the machine learning-based approaches depend mainly on feature engineering, struggles with unstructured data, the curse of dimensionality, and sensitivity to data scaling, as shown in Table 3.

Table 3. Literature review on machine learning-based approaches

References	Architecture	Dataset	Accuracy	Analysis	Limitation
[14]	Low-rank bilinear pooling for fine-grained classification	CUB-200-2011	84.21%	A high-dimensional bilinear feature vector, created by pooling the results of second-order local feature statistics, outperformed contemporary methods across various fine-grained bird classification tasks. To address the complexity of high-dimensional feature spaces, this study proposes covariance features and applies them to a low-rank bilinear classifier.	Machine learning approaches depend mainly on feature engineering, struggle with unstructured data, the curse of dimensionality, and are sensitive to data scaling.
[15]	Analysis of learning techniques: bird species classification	CUB-200-2011	33%	Various classifiers were studied for the purpose of bird species classification.	
[16]	An improved transfer-learning for image-based species classification of protected Indonesian birds	Own dataset	88.07%	Batch normalization, dropout, and fully connected (BNDFC) layers were proposed, which can be incorporated into any CNN.	
[17]	Cross-domain deep feature combination for bird species classification with audio-visual data	CUB-200-2011	78.9%	A multi-model bird species classification was proposed, which enhances the overall classification accuracy.	
[18]	Automatic bird species recognition from images with feature enhancement and contrastive learning	CUB-200-2011	65.2%	multiscale feature fusion to extract information at different scales. An attention feature enhancement module was integrated to address occlusion and noise. Furthermore, the Siamese network was used to compare two images belonging to the same class and different classes.	

2.4 Optimization-based approaches

Optimization-based approaches for classification were discussed in studies [19, 20]. The following paragraphs present a brief description and critical analysis of these approaches.

Kittler et al. [19] proposed a theoretical framework for combining multiple classifiers using various combinations of rules to improve pattern classification performance. Each classifier utilizes a unique pattern representation technique to make decisions. Four decision classifiers, such as structural, Gaussian, neural network, and hidden Markov model, were studied. The accuracy of the proposed classifier ensemble was evaluated through handwritten digit classification, which achieved a classification accuracy of 98.19% with the median combining rule. The approach is computationally expensive and suffers from the risk of overfitting and noise sensitivity.

Kumar and Kondaveeti [20] proposed an explainable artificial intelligence to increase the clarity of deep architecture classifiers for bird species classification. Popular deep architecture classifiers such as EfficientNetB0, DarkNet53, DenseNet201, SqueezeNet, InceptionResNetV2, and NasNetLarge were trained using CUB-200-2011. The authors have claimed a highest accuracy of 99.51% for EfficientNetB0. Explanations given by XAI are approximations and incomplete, as the single statement rarely tells the complete story. As explanations are filtered through cognitive biases. Explainable models are less accurate, create a false sense of trust, and are hard to audit, as shown in Table 4. In the present study, an optimal chromosome has been achieved that outperforms all the individual classifiers. It's the best combination of the available classifiers, which has enhanced the classification performance. The classifiers have not overfitted because of the availability of the validation split. The genetic algorithm has been best suited in the present scenario. The ensemble model using a genetic algorithm does not overcome the individual limitations; instead, it enhances the overall performance of the ensemble.

Table 4. Literature review on other optimization-based techniques

References	Architecture	Dataset	Accuracy	Analysis	Limitation
[19]	On combining classifiers	Handwritten digit classification dataset	98.19%	This work is very useful in introducing the combining rules for combination.	High computational cost, risk of overfitting, and sensitive to noise.
[20]	Towards transparency in AI: Explainable bird species image classification for ecological research	CUB-200-2011	EfficientNetB0 has achieved an accuracy of 80.421% on the CUB-200-2011 dataset.	This work helps in introducing explainable AI in the field of bird species classification.	XAI is incomplete as the single statement rarely tells the complete story. As explanations are filtered through cognitive biases, Explainable models are less accurate, create a false sense of trust, and are hard to audit.

3. METHOD

The methodology proposed in this study aims to identify a deep architecture classifier combination that can improve the accuracy of classifying an unknown bird image into one of the known classes, as per the training. The concept of a genetic algorithm was explored to identify the best deep architecture classifier combination. The steps involved in the proposed methodology are depicted in Figure 1 and described in the following paragraphs.

Initially, the methodology begins by splitting the images of the benchmark dataset into training, validation, and testing portions. The images in the training set were used to train the deep architecture classifiers, and those in the validation set were used to fine-tune the trained models.

Once the deep architecture classifiers are trained with the required number of iterations and are found to perform better in terms of classification accuracy, the training process is stopped. The classification ability of the trained individual deep architecture classifiers was tested using images in the test set, and their performance was recorded. It was observed that the models showed good performance for some classes and poor performance for some other classes. It has also been observed that the model that has shown good classification performance for one class does not show such good performance for some other classes. Hence, it is evident that no single model can provide an overall good classification accuracy for the considered dataset. Therefore, it is thought that ensembling the models and taking the collective decision enhances the accuracy of classification. However, to optimize the classifier combination, it is suggested to explore the concept of a genetic algorithm that produces the best overall classification accuracy for the considered dataset.

The following presents a brief description of the various deep architecture classifiers used in the proposed research work and the details of the selection of the best combination through a genetic algorithm-based approach.

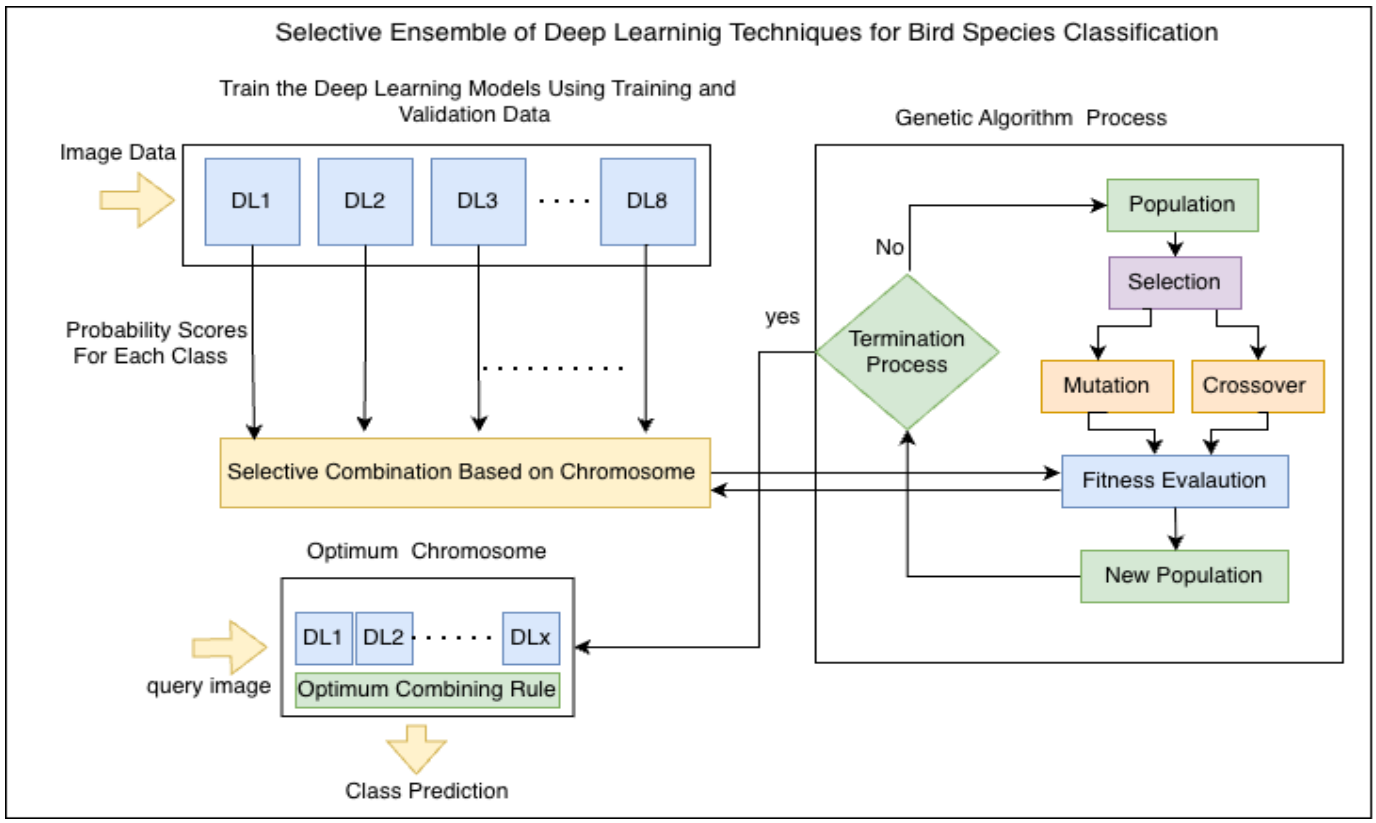

Figure 1. Architecture of the proposed genetic algorithm-based ensemble of deep learning models for appearance-based bird classification

3.1 The deep learning model's overview

Several deep CNN and transformer-based models have been proposed for various machine learning tasks, such as segmentation, classification, and prediction. All these models vary in architecture but have certain common features. The 1st layer in the deep architecture is the input layer, which accepts the image data, followed by a convolution layer, which is responsible for extracting meaningful features by performing convolutional operations. The subsequent layers, also called hidden layers, perform different convolution operations and derive a high-level abstraction from the input image. The number of hidden layers depends on the architecture that is designed to solve a particular problem. The final layer of the deep architecture was a fully connected layer that produced a complete set of features extracted from the image for characterization. The rectified linear unit (ReLu) activation function was used in all layers except the final layer, and the Softmax activation function was used in the final layer for classification.

Xception: A CNN architecture that depends completely on depth-wise separable convolution layers [21]. This is an extension of the inception model, where the inception modules are substituted with depth-wise separable convolution layers. The Xception model is organized into three blocks, namely, the entry, middle, and exit flows with skip connections around the thirty-six layers. The entry flow extracts low-level features from the input image, the middle flow progressively extracts

higher-level features from the image, and the exit flow refines these features for final predictions. This hierarchical structure aids in learning hierarchical representations and in the flow of data through the network.

VGG16: Deep CNN model used for image classification tasks [22]. The network was composed of 16 layers of artificial neurons, which were responsible for incrementally processing image information and enhancing the accuracy of its predictions. VGG16 uses convolution layers with a 3 × 3 filter and a stride of 1, which are in the same padding and max pool layer of a 2 × 2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the architecture. It had two fully connected layers at the end, followed by a Softmax activation function for the output.

VGG19: A variant of the VGG model, which consists of 19 layers, is characterized by its simplicity and uniform architecture [22]. The network is composed of a series of convolutional layers, followed by max-pooling layers, and several fully connected layers at the end. The use of small 3 × 3 convolutional filters throughout the network allows for a deeper architecture, while maintaining a relatively low number of parameters. A ReLU was used to introduce nonlinearity and improve the computational time and classification ability of the model.

ResNet152: A convolutional neural network that exploits the concepts of residual learning to dampen the degradation of deep neural networks and skip connections [23], which adds new inputs to the network and generates new outputs. This idea enables the model to be trained more deeply and achieve better classification accuracy without increasing the complexity of the model. ResNet152 has been proven to be the best model in terms of its classification accuracy among the ResNet family members, and hence it is considered in this study.

MobileNetV2: Simple but lightweight and efficient convolutional neural network model for mobile vision applications [24]. MobileNetV2 is widely used in many applications, including object detection, fine-grained classification, semantic segmentation, and localization. The MobileNetV2 model replaces depth-wise convolutions with standard convolutions to build a much lighter model with fewer parameters when compared with other networks to come up with a lightweight deep CNN.MobileNetV2 introduced two new global hyperparameters, the width multiplier and resolution multiplier, that allow model developers to trade off latency or accuracy for speed and small size, depending on their requirements.

InceptionNet: A convolutional neural network designed for image-classification tasks [25]. The inception architecture combines convolutional filters of different sizes in parallel to extract features from the input image at different scales. The output of each filter is then concatenated and sent to the next layer, where the process is done multiple times. This approach allows the network to capture local and global features of the input image while minimizing the number of parameters. The inception modules allow the network to learn temporal and spatial features from the input data because the module is composed of pooling and small convolutional layers. The aim was to make the model more L and faster. It has been widely used in various applications such as object detection, face recognition, and image classification.

EfficientNetV2L: A type of convolutional neural network that can be trained faster and has better parameter efficiency than previous models [26]. Both parameter efficiency and training speed were jointly optimized using a neural architecture search (NAS). Although EfficientNetV2L is larger in its family, it still outperforms the state-of-the-art method by enriching the search space with newer operations, such as fused-MBConv.

ViT: A transformer-based logic derived from Natural Language Processing (NLP) is applied in the field of computer vision. By inserting a new multi-layer perceptron as the head of the encoder, a ViT was used to classify birds based on the images. Training the ViT on large datasets produces better results than the CNN-based architecture in the field of image classification.

3.2 Ensemble of deep architecture classifiers using a genetic algorithm

A genetic algorithm is an optimization and search technique used to find the best ensemble of deep architecture classifiers in the field of image classification. The optimization model reduces the search space to determine the most optimal architecture with the highest fitness. The genetic algorithm is based on evolution and natural selection, which is inspired by the field of biology and incorporates the knowledge of survival of the fittest to return the chromosome with the highest performance.

From the literature survey, it can be inferred that the ensemble learning system has achieved significant improvement in classification accuracy, and many ensemble learning techniques have also been introduced by Mousavi and Eftekhari [12]. It is well known that the concept of genetic algorithms has also been explored to determine the optimal solution. Based on this observation, the deep architecture classifiers presented in the previous section were assembled using a genetic algorithm. Different combining rules, as shown in Table 5, were used for ensembling. Table 5 presents six different combining rules and the corresponding procedures used to ensemble the deep architecture classifiers. For a given sample, the class-wise probabilities obtained by the individual learning models were combined based on the combining rule, and the final decision for classification was made accordingly. Table 6 presents the various deep architecture classifiers used in the proposed methodology for ensembling, and their corresponding bits in the chromosome.

Table 5. Combining rules and the procedure for ensembling deep architecture classifiers

No.	Combining	Description	Procedure	Encoding
1	Majority	Majority	Class Title	000
2	Maximum	Maximum rule	Supporting	001
3	Minimum	Minimum rule	Supporting	010
4	Mean	Mean rule	Supporting	011
5	Median	Median	Supporting	100
6	Product	Product rule	Supporting	101

Table 6. Deep architectures used in the experiment and their encoding bits on the chromosome

Deep	Encoding
Xception	1
Inception	2
MobileNetV2	3
ResNet152	4
VGG19	5
VGG16	6
EfficientNetV2	7
ViT	8

Figure 2. Diagram representing the structure of a chromosome

3.3 Representation of a chromosome

The concept of a genetic algorithm is based on the representation of chromosomes. In the present study, a chromosome is a pattern of bits that represents a list of deep architecture classifiers and their combination rules. A chromosome was defined with 11 binary bits, where the first 3 bits are used to define combining rules, and the remaining 8 bits are used for the inclusion or exclusion of 8 different deep architecture classifiers considered in this work. The six different combining rules mentioned in Table 5 can be uniquely identified by 3 bits. Figure 2 shows the structure of the chromosomes.

3.4 Selection of the best deep architecture classifiers ensemble

Algorithm 1: Selection of Best Deep Architecture Classifier Ensemble

Input: (i) population size: S
(ii) point of crossover 'p1',
mutation rate 'p2', structure of the chromosome, fitness metric, and number of generations 'N'.
Output: Best performing chromosome
Method:

1. Randomly choose the chromosomes based on the population size and include the chromosomes that are expected to perform well.
2. Evaluate the fitness of each of the chromosomes.
3. Loop start:
   - From the population, randomly pick two chromosomes and apply binary tournament selection.
   - Apply crossover or mutation with equal chances.
   - Add newly generated chromosomes to the existing population.
   - Rank all the chromosomes based on fitness.
   - Select the top chromosomes based on fitness for the next generation.
   - End the loop when all the chromosomes in the population do not change consistently for 4 generations.

   Loop end:
4. Return the chromosome with the highest fitness (classification accuracy).

The concept of a genetic algorithm is used to identify the best combination of deep architecture classifiers to improve classification accuracy. According to the structure of the chromosome mentioned above, if the brute-force method is used to find the optimal chromosome structure, then the algorithm should compute all 1536 (28 × 6) possible combinations for eight different deep architecture classifiers and six different combining rules. However, the proposed genetic algorithm performs only 300 possible combinations and reduces overall complexity. In the proposed genetic algorithm, classification accuracy is used as the fitness metric, as mentioned in Algorithm 1.

In prior ensemble methods, crossover and mutation operations are performed in sequence (mutation followed by crossover), but in the proposed work, these operations are performed in parallel. By applying mutation and crossover operators in parallel, the genetic algorithm searches the local optima (solutions with lower hamming distance) using a lower mutation rate and searches the wider search space with the crossover operator. But in the prior work, mutation followed by crossover searches for diverse solutions (solutions with higher hamming distance).

3.4.1 Binary tournament selection

Binary tournament selection is a methodology used in the genetic algorithm for the purpose of selecting individuals based on their fitness. Two chromosomes were randomly selected from the population, and the accuracy of both chromosomes was estimated based on the combined rule and selected deep architecture classifiers. The computed accuracy of classification was assigned as the fitness of the chromosomes; the fitness of both individuals was compared, and chromosomes with the highest fitness were returned as the output of the binary tournament selection in Algorithm 2.

Algorithm 2: Binary Tournament Selection

Input: Randomly select two chromosomes from the population 'P.'
Output: Chromosome with the higher fitness
Method:

1. let p1, p2 ∈ P, where p1 = r and(P) and p2 = rand(P)
2. A1= pattern(p1) and A2 = pattern(p2) where A1 and A2 are the pattern of the chromosomes
3. fitness1 = F(A1), fitness2 = F(A2), where F(.) returns the accuracy of the classification for the given pattern.
4. $Pi = \begin{cases} p1 & \text{if fitness 1} > \text{fitness 2} \\ p2 & \text{otherwise} \end{cases}$
5. Return Pi

3.4.2 Crossover

Crossover is a genetic operation used to combine genetic information to produce new offspring, as shown in Algorithm 3. Two chromosomes selected using the tournament selection technique were subjected to recombination. In the present experiment, a one-point crossover was used to estimate the recombination, where all the genes were exchanged to produce new offspring after the crossover point. Example: Let

Chromosome A = 00000000000, Chromosome B = 11111111111; if the point of crossover is the 3rd bit, then the newly generated offspring are A = 00111111111 and B = 11000000000. Here, all the genes after the reference point are exchanged.

Algorithm 3: Crossover

Input: Population, chromosome size 'N.'
Output: Output architecture of the two chromosomes
Method:

1. Let p_1 and p_2 be the two chromosomes selected based on two binary tournament selection algorithms.
2. P be the point of crossover such that p = rand (0, N) where rand () is a random function
3. $y_1[i] = \begin{cases} p_1[i] \text{ if } i < P \\ p_2[i] \; otherwise \end{cases}$ where i = 0…, N
4. $y_2[i] = \begin{cases} p_2[i] \text{ if } i < P \\ p_1[i] \text{ otherwise} \end{cases}$ where i = 0…, N
5. return y_1 and y_2.

3.4.3 Mutation

Mutation is a genetic operator used to produce offspring by altering the chromosomes. A chromosome selected using the tournament selection technique was subjected to mutations. In this experiment, bits of the chromosome were flipped based on the mutation rate. While processing a chromosome, each bit is compared with the value generated by a random function in the interval [0,1]. If the function generates a value less than the mutation rate, then the bit is flipped. Thus, by flipping the bits in the chromosomes, the mutation enabled us to find a better chromosome as presented in Algorithm 4. The higher the mutation rate, the higher the chance of flipping the bits. Hence, the mutation rate is directly related to the Hamming distance between chromosomes before and after the mutation application.

Algorithm 4: Mutation

Input: Input chromosome 'C', mutation rate 'P', chromosome size 'N'
Output: Output Chromosome
Method:

1. Let p be the chromosome selected based on two binary tournament selection algorithms.
2. $y[i] = \begin{cases} p[i] & \text{if } \text{rand}([0,1]) > P \\ 1 - p[i] & \text{otherwise} \end{cases}$, where, i = 0, …, N
3. Return y

3.4.4 Production of the new generation

During the execution of the genetic algorithm, the new generation replaces the old generation based on the fitness values of the chromosomes. If the population size is 'N,' the population size doubles when new offspring are generated and added to the present population in Algorithm 5. To pass the chromosomes to a newer generation, all the chromosomes are first ranked based on fitness; further, the top N chromosomes are considered as the newer population, as the evolution methodology is based on the survival of the fittest.

During the workflow of the genetic, whenever new chromosomes are generated or modified, they are added to the chromosome bank. This repository helps to observe all chromosomes that were searched by the genetic algorithm. This bank is very helpful in displaying the top 10 chromosomes.

Algorithm 5: Production of New Generation

Input: Population of the present generation $P = \{p_1, p_2, p_{3\ldots.}\, p_m\}$
Output: Population of the next generation of size 'N', $P_i = \{p_1, p_2, \; p_{3\ldots.}\, p_n\}$ where n < m
Method:

1. $A_i = accuarcy\ (p_i)$, where, $i = 0, \ldots,$ m, where A_i is the accuracy of each of the chromosomes 'i' and p_i is the i^{th} chromosome of the population P.
2. $Fitness_i = A_i$ where accuracy is assigned as the Fitness of each chromosome
3. P = Sort $(p_1, p_2, \; p_{3\ldots.}\, p_m)$, where, P is the sorted population in decreasing order of the fitness $A_i \geq A_{i+1}$ where $i = 0, \ldots,$ m.
4. X = $\{p_1, p_2, \; p_{3\ldots.}\, p_n\} = \{p_1, p_2, \ldots p_n \ldots p_m\}$ such that n < m
5. return X, where X is the population of the next generation.

3.4.5 Dealing with erroneous chromosomes

During the execution of each step of the genetic algorithm, there is a chance of generating an erroneous chromosome. As there are 3 bits allotted for the encoding of the combining rules, the three bits can generate eight possible combinations, where there are only six combining rules. The remaining two possible combinations are erroneous because they do not represent any combining rules. Furthermore, during crossover and mutation operations, an erroneous chromosome may be generated. To address such situations, exception handling is necessary.

As per the principles of the genetic algorithm, whenever offspring are generated by crossover or mutation, they are added to the existing population. If n_1 is the population size and n_2 is the number of chromosomes generated by crossover or mutation operations, then the total number of chromosomes to be processed becomes $n_3 = n_1 + n_2$. However, before passing all these chromosomes to the next generation, the chromosomes are sorted, and the top n << n_3 chromosomes are passed to the next generation.

4. EXPERIMENTATION AND RESULTS

4.1 Dataset

Figure 3. Example images from the UOM-IBID (Indian Bird Image Dataset)
Note: Dataset Link: https://www.kaggle.com/datasets/coderchandruu/birds-found-in-india.

Table 7. Distribution of samples for training, validation, and testing for each species (90% train, 5% validation, 5% test split)

Total Images in Each Class	Images Used for Training	Images Used for Validation	Images Used for Testing
150	135	7	8

The Indian Bird Image Dataset, named as UOM-IBID, was created and used to study the performance of the proposed methodology. The dataset was created by collecting the images from legitimate Internet sources. The dataset consists of 6150 bird images belonging to 41 classes; each class has 150 images. There were significant intra-class variations in the dataset, as a single bird was captured at different angles and poses. The dataset is quite complex when compared to available datasets, as the birds are captured with complex backgrounds, bird images are occluded with other objects, and the birds are captured in different poses and angles. Figure 3 shows a few example images taken from the dataset.

Table 7 shows the distribution of training, validation, and test splits used to study the performance of the proposed model.

4.2 Experimental setup

Several experiments were conducted on the UOM-IBID dataset to study the performance of the proposed methodology and fine-tune the parameters to enhance the classification accuracy and robustness of the classification approach. The following subsections provide details of the parameters used for training the model.

4.2.1 Parameters used for training deep architecture classifiers

All the bird images (Figure 3) are resized to a size of (224, 224, 3). A batch size of 32 was used in the experiment. 'Include top' parameter was set as False, as the fully connected layers were not included. Stochastic gradient descent (SGD) was used as the optimizer of the model. Accuracy was used as the metric for the optimization of the model. The SGD learning rate was initialized as 0.001, the SGD decay as 1×10^{-6}, and the SGD momentum as 0.9. The callbacks of the model were declared with 'Restore best weights' as TRUE and with a patience of 5. The categorical_crossentropy was used as a loss function.

Table 8. Classification results for UOM-IBID (Indian Bird Image Dataset)

Training/Testing	Train-90%	Train-80%	Train-70%
vs.	val-5%	val-10%	val-15%
DL Models	Test-5%	Test-10%	Test-15%
Xception	85.67%	82.11%	82.93%
ResNet152	88.11%	84.07%	81.44%
MobileNetV2	85.67%	84.88%	81.87%
Inception	84.15%	81.30%	81.12%
EfficientNetV2L	91.46%	87.32%	86.53%
VGG 16	82.62%	74.63%	70.73%
VGG 19	84.15%	74.47%	74.76%
Vision Transformer	95.76%	94.38%	93.96%

4.2.2 Results on different ratios of training and testing data

Several experiments were conducted to study the classification accuracy of the proposed methodology by considering the parameters considered in the previous sections. Table 8 presents the results of the experiments conducted for the deep architecture classifiers considered in this study with different percentages of training and testing for bird images.

4.2.3 Parameters used in the genetic algorithm

Some parameters associated with the genetic algorithm proposed in this study need to be initialized and fine-tuned during its execution.

New chromosomes are generated in each iteration by performing genetic operations, such as mutation and crossover. It has been observed that new chromosomes generated in this manner will be stable after 10 generations. Therefore, the maximum number of iterations required to produce stable chromosomes in this experiment was 10. Thus, the search space computation using a genetic algorithm is equal to the product of the number of generations and population size (10×30). As $1536 > 300$, the search space for the selection of the best classifier ensemble is reduced using a genetic algorithm when compared with the brute force technique.

The population size should be as large as possible so that a larger diversity of chromosomes can be considered in the experiment. The population size should be as small as possible so that the product of the number of generations and population size is smaller than the computations done by the brute force technique. Hence, the population size is a trade-off between these two conditions. In the experiment, the population size was empirically chosen as 30, as it optimizes the trade-off criteria. The point of crossover is chosen as 1, as it is the standard value used in most genetic algorithm implementations.

The mutation rate should be as large as possible so that diverse solutions are explored with fewer generations. The mutation rate should be chosen as small as possible so that solutions with local optima can be explored. Hence, the mutation rate is empirically chosen as 0.5, as it optimizes the trade-off criteria. Table 9 lists the parameters used in the experiments.

Table 9. Parameters used in the genetic algorithm

Parameters	Values in the Experiment
Population size	30
Point of crossover	1
Mutation rate	0.5
Fitness metric	Accuracy of classification
The number of generations is decided by observing no change in the population elements consistently for x generations	x = 4

Table 10. Performance of individual deep architecture classifiers and their ensembles for 90% training, 5% validation, and 5% test data split

Deep Architecture Classifiers	Individual Performance	Performance of Classifier Ensemble
Xception	85.67%	96.95%
ResNet152	88.10%	
MobileNetV2	85.67%	
Inception	84.15%	
EfficientNetV2L	91.46%	
VGG16	82.62%	
VGG19	84.15%	
Vision Transformer	95.76%	

Performance of the individual deep architecture classifiers and their ensemble is presented in Table 10. It can be observed from the results that the ensemble of deep architecture classifiers using the proposed genetic algorithm has enhanced the accuracy of the classification by selectively choosing the right combination of deep architecture classifiers and the right combining rule. Although it is possible to come up with the right combination of deep architecture classifiers with brute force methodology, it is a very cumbersome task, and the complexity of such a technique is quite high. Time complexity can be drastically reduced by applying a genetic algorithm, as discussed. Figure 4 represents a graph depicting training and validation loss, and Figure 5 represents a graph depicting training and validation accuracy for the VGG-16 deep Architecture classifier. These graphs evaluate that the model is not overfitting or underfitting while training. Figure 6 presents the graphical representation of different deep architecture classifiers and their ensemble, clearly showing that the ensemble outperforms individual classifiers.

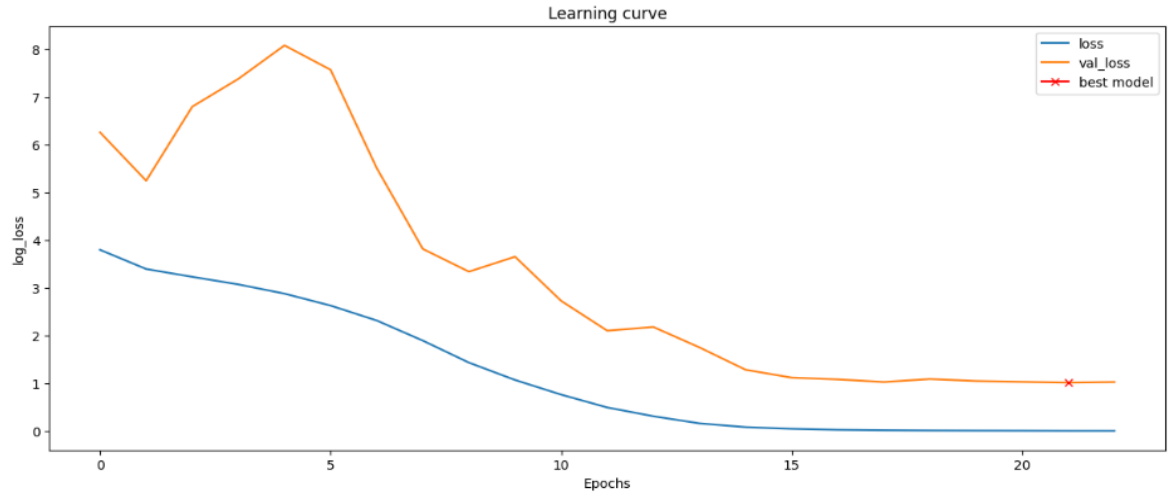

Figure 4. Loss v/s epochs for VGG-16 deep architecture classifier (70% Training, 15% Validation, and 15% Test split)

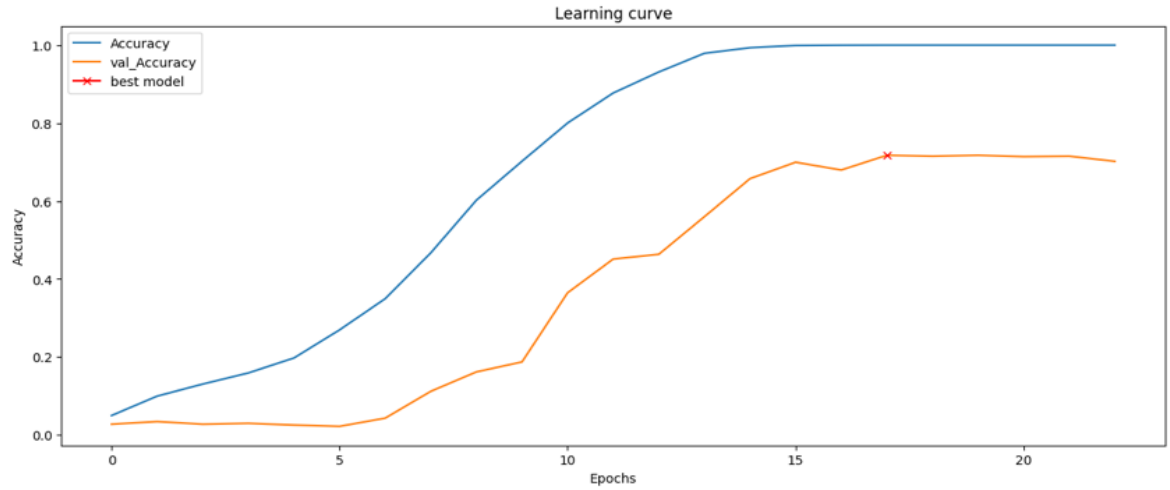

Figure 5. Accuracy v/s epochs for VGG16 deep architecture classifier (70% Training, 15% Validation, and 15% Test split)

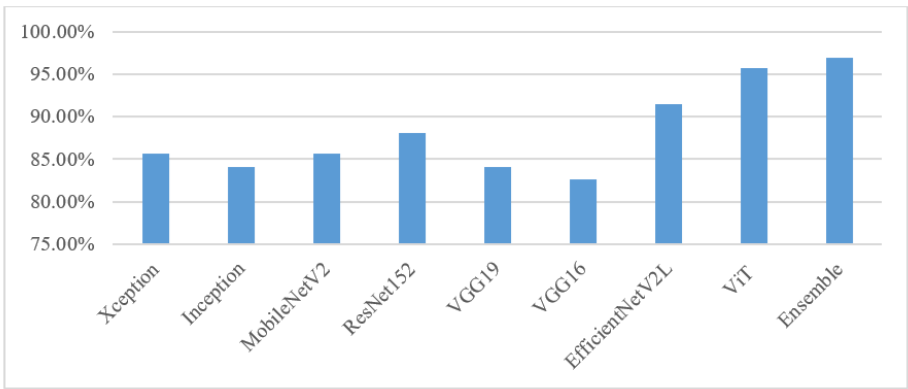

Figure 6. Bar graph representing the accuracy of different deep architecture classifiers

Table 11. Best performing chromosomes in terms of fitness

	Encoded String	Combining Rule	Deep Architectures Classifiers Included
First best chromosome	'00001011011'	Mean	VGG19, Efficentnetv2L, Vision Transformer

Table 12. Best performing top 10 chromosomes for UOM-IBID

No.	Chromosome	Accuracy
1	'00001011011'	96.95%
2	'00000011011'	96.64%
3	'00000111101'	96.64%
4	'00001111011'	96.64%
5	'00000011100'	96.64%
6	'00001011010'	96.34%
7	'10001011011'	96.34%
8	'00000011001'	96.34%
9	'00000011010'	96.34%
10	'00000111011'	96.34%

4.3 Analysis of the fit chromosome

In this context, the fitness of any chromosome is measured in terms of its classification performance by combining the different deep architecture classifiers considered for the analysis and the combining rule. The best combination of the deep architecture classifier ensemble and the best combining rules was analyzed by conducting experiments and fine-tuning the parameters of the genetic algorithm. Experimental results revealed that the combination of a VGG19, EfficientNetV2L, and ViT with a mean combining rule produced the highest accuracy of 97%. The details are presented in Table 11.

From the experiments, the best-performing top 10 chromosomes were observed and are presented in Table 12. The structure of each chromosome indicates the presence (1) or absence (0) of a particular deep architecture classifier and the combining rules for ensembling. For example, chromosome 10000011101 indicates that the deep architecture classifiers considered for ensembling are Xception, EfficientNetV2L, and ViT, as mentioned in 6, and the 'product' combining rule as described in Table 5. From Table 12, it can be observed that all the top 10 chromosomes have the 7th and 8th bit as one, which indicates that the combination of EfficientNetV2L and ViT has shown a better classification accuracy.

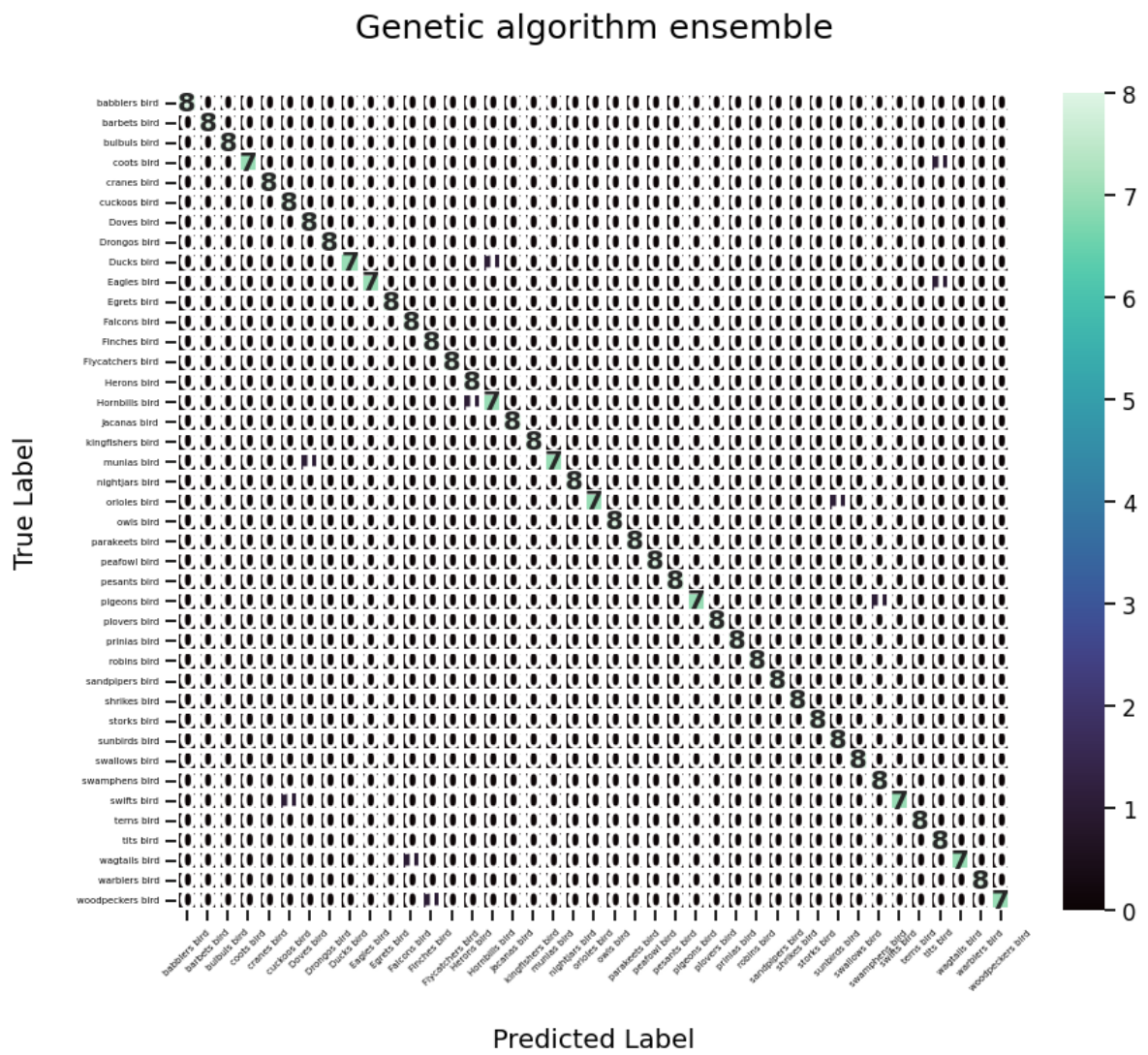

Figure 7. Confusion matrix of the proposed deep architecture classifier ensemble
Note: Source code link: https://github.com/chandrashekar6/Genetic-algorithm-ensemble-for-classification.

The top chromosomes work better because they have the highest fitness. The researcher does not have any control over the output of the genetic algorithm, and the genetic algorithm decides the output. The pattern of the best-performing chromosome states that the combination of VGG19, ViT, and EfficientnetV2L with the Mean combining rule is most suited for bird species classification. Individually, ViT and EfficientNetV2L performed very well. Hence, any combination of these two models performs better. Table 12 presents the top 10 chromosomes, where all the chromosomes have ViT and EfficientNetV2L in common.

Performance of the proposed ensemble classification approach was also studied by conducting several experiments and computing class-wise precision, recall, and F-measures, and the results are presented in Table 13. From the results, it was observed that for most of the classes, the results are good, with high precision, recall, and F-measure. However, for a few classes, the results require improvement. Figure 7 presents the confusion matrix of the proposed deep architecture classifier ensemble.

Table 13. Class-wise precision, recall, and F1-score obtained for the Indian bird dataset using the best classifier ensemble

No.	Bird Class	Precision	Recall	F1-Score	Support
1	Babblers	1	1	1	8
2	Barbets	1	1	1	8
3	Bulbuls	1	1	1	8
4	Coots	1	0.88	0.93	8
5	Cranes	1	1	1	8
6	Cuckoos	0.89	1	0.94	8
7	Doves	0.89	1	0.94	8
8	Drongos	1	1	1	8
9	Ducks	1	0.88	0.93	8
10	Eagles	1	0.88	0.93	8
11	Egrets	1	1	1	8
12	Falcons	0.89	1	0.94	8
13	Finches	0.89	1	0.94	8
14	Flycatchers	1	1	1	8
15	Herons	0.89	1	0.94	8
16	Hornbills	0.88	0.88	0.88	8
17	Jacanas	1	1	1	8
18	Kingfishers	1	1	1	8
19	Munias	1	0.88	0.93	8
20	Nightjars	1	1	1	8
21	Orioles	1	0.88	0.93	8
22	Owls	1	1	1	8
23	Parakeets	1	1	1	8
24	Peafowl	1	1	1	8
25	Pheasants	1	1	1	8
26	Pigeons	1	0.88	0.93	8
27	Plovers	1	1	1	8
28	Prinias	1	1	1	8
29	Robins	1	1	1	8
30	Sandpipers	1	1	1	8
31	Shrikes	1	1	1	8
32	Storks	1	1	1	8
33	Sunbirds	0.89	1	0.94	8
34	Swallows	1	1	1	8
35	Swamphens	0.89	1	0.94	8
36	Swifts	1	0.88	0.93	8
37	Terns	1	1	1	8
38	Tits	0.80	1	0.89	8
39	Wagtails	1	0.88	0.93	8
40	Warblers	1	1	1	8
41	Woodpeckers	1	0.88	0.93	8

5. CONCLUSION

A novel deep architecture classifier ensemble using a genetic algorithm for bird species classification was proposed. The genetic algorithm significantly reduced the computational search space for identifying the best deep architecture classifier ensemble compared to the brute-force approach.

The process of using selective deep architecture classifiers among the eight was performed using a genetic algorithm. The mechanism of the genetic algorithm was applied by assigning the accuracy of classification as the fitness. Binary Tournament selection was used to find the fitter parents during the execution of the genetic algorithm, which selected the fit chromosome among two randomly selected chromosomes.

Mutations and crossovers were applied in parallel with equal probabilities. Based on the experiments, it can be concluded that a combination of two deep architecture classifiers, namely VGG19, EfficientNetV2L, and ViT, produced the highest classification accuracy of 97% for the mean combining rule. This combination was computed only once and stored in a knowledge base for the classification of unknown bird images. The proposed model fails with poor-quality images, and there is still scope for researchers to work in this direction.

REFERENCES

[1] Chen, Y., Song, J., Song, M. (2022). Hierarchical gate network for fine-grained visual recognition. Neurocomputing, 470: 170-181. https://doi.org/10.1016/j.neucom.2021.10.096

[2] Branson, S., Van Horn, G., Belongie, S., Perona, P. (2014). Bird species categorization using pose normalized deep convolutional nets. arXiv preprint arXiv:1406.2952. https://doi.org/10.48550/arXiv.1406.2952

[3] Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K. (2015). Spatial transformer networks. arXiv preprint arXiv:1506.02025. https://doi.org/10.48550/arXiv.1506.02025

[4] Manna, A., Upasani, N., Jadhav, S., Mane, R., Chaudhari, R., Chatre, V. (2023). Bird image classification using convolutional neural network transfer learning architectures. International Journal of Advanced Computer Science and Applications, 14(3): 854-864. https://doi.org/10.14569/IJACSA.2023.0140397

[5] Zhang, N., Donahue, J., Girshick, R., Darrell, T. (2014). Part-based R-CNNs for fine-grained category detection. In Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, Springer, Cham. https://doi.org/10.1007/978-3-319-10590-1_54

[6] Huang, Y.P., Basanta, H. (2021). Recognition of endemic bird species using deep learning models. IEEE Access, 9: 102975-102984. https://doi.org/10.1109/ACCESS.2021.3098532

[7] Wei, X. S., Xie, C. W., Wu, J. (2016). Mask-CNN: Localizing parts and selecting descriptors for fine-grained image recognition. arXiv preprint arXiv:1605.06878. https://doi.org/10.48550/arXiv.1605.06878

[8] Kumar, S.V., Kondaveeti, H.K. (2024). Bird species recognition using transfer learning with a hybrid

hyperparameter optimization scheme (HHOS). Ecological Informatics, 80: 102510. https://doi.org/10.1016/j.ecoinf.2024.102510

[9] Ngo, C.Y., Chong, L.Y., Chong, S.C., Goh, P.Y. (2023). Bird species recognition system with fine-tuned model. International Journal on Advanced Science, Engineering & Information Technology, 13(5): 1719-1726. https://doi.org/10.18517/ijaseit.13.5.19030

[10] Bold, N., Zhang, C., Akashi, T. (2019). Bird species classification with audio-visual data using CNN and multiple kernel learning. In 2019 International Conference on Cyberworlds (CW), Kyoto, Japan, pp. 85-88. https://doi.org/10.1109/CW.2019.00022

[11] Pang, C., Yao, H., Sun, X. (2014). Discriminative features for bird species classification. In ICIMCS '14: Proceedings of International Conference on Internet Multimedia Computing and Service, Xiamen China, pp. 256-260. https://doi.org/10.1145/2632856.2632917

[12] Mousavi, R., Eftekhari, M. (2015). A new ensemble learning methodology based on hybridization of classifier ensemble selection approaches. Applied Soft Computing, 37: 652-666. https://doi.org/10.1016/j.asoc.2015.09.009

[13] Dharmawan, T., Auliya, Y.A., Retnani, D.A., Bukhori, S., Zarkasi, M., Ataama, I. (2025). Optimizing DenseNet121 for waste classification using genetic algorithm-based downsampling and data augmentation. Mathematical Modelling of Engineering Problems, 12(5): 1711-1717. https://doi.org/10.18280/mmep.120526

[14] Kong, S., Fowlkes, C. (2017). Low-rank bilinear pooling for fine-grained classification. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, pp. 7025-7034. https://doi.org/10.1109/CVPR.2017.743

[15] Rai, P.K., Chaturvedi, S., Katiyar, S. (2019). Analysis of learning techniques: Bird species classification. International Journal of Computer Applications, 178(17): 12-16. https://doi.org/10.5120/ijca2019918969

[16] Yang, C.L., Harjoseputro, Y., Hu, Y.C., Chen, Y.Y. (2022). An improved transfer-learning for image-based species classification of protected Indonesians birds. Computers, Materials & Continua, 73(3): 4577-4593. https://doi.org/10.32604/cmc.2022.031305

[17] Naranchimeg, B., Zhang, C., Akashi, T. (2018). Cross-domain deep feature combination for bird species classification with audio-visual data. arXiv preprint arXiv:1811.10199. https://doi.org/10.48550/arXiv.1811.10199

[18] Yang, F., Shen, N., Xu, F. (2024). Automatic bird species recognition from images with feature enhancement and contrastive learning. Applied Sciences, 14(10): 4278. https://doi.org/10.3390/app14104278

[19] Kittler, J., Hatef, M., Duin, R.P., Matas, J. (2002). On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(3): 226-239. https://doi.org/10.1109/34.667881

[20] Kumar, S.V., Kondaveeti, H.K. (2024). Towards transparency in AI: Explainable bird species image classification for ecological research. Ecological Indicators, 169: 112886. https://doi.org/10.1016/j.ecolind.2024.112886

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

[22] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. https://doi.org/10.48550/arXiv.1704.04861

[23] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, pp. 1-9. https://doi.org/10.1109/CVPR.2015.7298594

[24] Tan, M., Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning, pp. 6105-6114.

[25] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, Heigold, G., Gelly, S., Jakob Uszkoreit, J., Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929. https://doi.org/10.48550/arXiv.2010.11929

[26] Sitepu, A.C., Liu, C.M., Sigiro, M., Panjaitan, J., Copa, V. (2022). A convolutional neural network bird's classification using north American bird images. International Journal of Health Sciences, 6(S2): 15067-15080. https://doi.org/10.53730/ijhs.v6nS2.8988