

TurtleNet: Explainable Turtle Species Classification Using Attention-Enhanced EfficientNetB4 and Ensemble Learning

Pragyat Jyoti Baruah¹, Bugge Prince Solomon Raj Kumar¹, Arun Jyoti Nath², Arnab Paul¹, Jarine Anjum¹, Akash Nath¹, Tirthanka Borah¹

¹Department of Computer Science and Engineering, Assam University, Silchar, India

²Department of Ecology and Environmental Science, Assam University, Silchar, Assam, India

Article Info

Article history:

Received Mar 17, 2026

Revised Apr 22, 2026

Accepted Jun 11, 2026

Keywords:

Turtle Species Classification

Computer Vision

EfficientNetB4

Ensemble Learning

Explainable AI

ABSTRACT

Turtles are remarkable organisms as they play a vital role in maintaining a balanced and healthy ecosystem. However, their existence is at risk due to growing threats of habitat loss, illegal trade and environmental degradation. To support conservation and monitoring efforts for these turtle species, this study proposes TurtleNet, a custom deep learning model designed to accurately identify turtle species with acute precision. The proposed model relies on a customized EfficientNetB4 architecture as a backbone with an enhanced lightweight attention mechanism of Squeeze and Excitation blocks, allowing it to focus on subtle visual features that distinguish closely related species. The model was trained and evaluated on our curated Turtle1 dataset as well as the images.cv turtle dataset. The performance comparison of TurtleNet with 11 individual state-of-the-art deep learning classification models showed its superior ability by classifying Asian turtle species with 97.25% accuracy. To further enhance classification performance, TurtleNet is integrated with the YOLOv8 model using a soft-voting ensemble strategy, thereby increasing the classification accuracy by 2.41%. The results demonstrate that TurtleNet has strong potential for automated species monitoring and conservation using non-invasive measures. Grad-CAM visualization is used to highlight discriminative morphological regions relevant to image identification and is used to interpret the feature selection process of the proposed model.

Copyright © 2026 Institute of Advanced Engineering and Science.

All rights reserved.

Corresponding Author:

Arnab Paul,

Department of Computer Science and Engineering,

Assam University, Silchar,

Assam, 788011, India.

Email: arnab.paul@aus.ac.in

1. INTRODUCTION

Turtles are ancient reptiles of the order Testudines [1] that are characterized by a cartilaginous or bony covering [2] that encloses their soft body parts [3]. Turtles play a vital role in both aquatic and terrestrial ecosystems worldwide, contributing to nutrient cycling and controlling aquatic plant populations [4]. India possesses an outstanding diversity of turtle species that inhabit rivers, wetlands, coastlines, and forests, acting as ecological health indicators. However, most turtle species are currently threatened by habitat destruction, pollution, illegal trade, and human consumption [3]. Therefore, effective conservation and continuous monitoring efforts are essential for their protection. This study aims to develop an efficient species classification framework tailored to turtles, to support conservation and monitoring efforts.

Automatic turtle species classification uses deep learning methods to identify species from images in a non-invasive way. This approach aids conservation by enabling quick and accurate identification through image-based analysis. Previous studies on turtle classification include the following: Works such as [5], [6], [7], [8], [9], [10] used deep learning approaches to extract features from turtle images for species classification.

While deep learning has revolutionized turtle species classification, the model proposed by [11] remains limited, as it relies on color histograms and chromaticity moments for feature extraction and employed machine learning methods like K-nearest neighbor (KNN) and Support Vector Machine (SVM) approaches for classification, thus achieving an accuracy of 74%. The study in [5] proposed an ensemble of three CNNs (InceptionV3, DenseNet201, VGG16) to improve the classification performance of the individual CNNs, achieving an accuracy of 74%. Similarly, [9] improved the performance of YOLOv8 using an ensemble of multi-view image processing with a CNN architecture and achieved 98% accuracy on the test dataset. Transfer learning was employed by [6], using the YOLOv5 framework for object detection and instance segmentation, thereby enabling enhanced sea turtle monitoring with 91.80% accuracy. They introduced instance segmentation to improve the task of detecting turtles in complex underwater environments. [7] achieved 82.80% accuracy by using ResNet18 architecture for identifying invasive turtle species in Korea. [8] performed the Sobel operator for edge and feature detection from turtle images and used a histogram for pixel intensity distribution analysis, achieving a classification accuracy of 90.40% using ResNet50. [10] used a CNN-based transfer learning with Inception-v3 for turtle species recognition.

There are 28 freshwater turtles and tortoises found in India [12], which include 18 IUCN Threatened turtle species [3]. Most turtle species in India are listed as either Endangered or Critically Endangered on the IUCN Red List. Native Indian species like *Nilssonina nigricans* are extinct in the wild [3]. As reported by [12], at least 13 of India's freshwater turtles, which included 8 IUCN Threatened species, were illegally harvested, with over 58,000 live individuals seized from illicit traders between 2011 and 2015. These challenges highlight the need for effective conservation and regular monitoring of turtle species in their natural habitats. Although India is home to a rich diversity of turtle species, automated identification systems for native Indian turtle species remain limited.

A deep learning-based framework for Indian turtle species classification has been proposed in this paper. This framework uses the EfficientNetB4 architecture with a Squeeze-and-Excitation (SE) attention block for feature extraction. It also employs transfer learning to improve the classification performance. The proposed model is also compared with multiple variants of CNNs, including InceptionV3 [13], ResNet101 [14], MobileNetV3 [15], ConvNeXt-Base [16], DenseNet169 [17], VGG16 [18], YOLOv8 [19], Inception-ResNet-V2 [20], ResNet152V2 [14], Xception [21], and VGG19 [18] during evaluation.

Two turtle image datasets are used to evaluate the models. The first dataset (Turtle1) consists of images of six turtle species found in Asia, with most species native to India. The dataset was manually curated using images collected from different online sources. The second dataset (images.cv turtle dataset [22]) consists of images of four turtle species and is used as a benchmark dataset for performance evaluation. An ensemble learning strategy combining TurtleNet and YOLOv8 is also employed to improve the classification performance. Previous studies have reported that ensemble learning often achieves better performance than individual models [5], [9]. The objective of this study is to develop an accurate computer vision-based framework for turtle species classification to support wildlife conservation.

The remainder of this paper is organized as follows. Section 2 describes the datasets and the proposed methodology. Section 3 presents the experimental results and discussion. Finally, Section 4 concludes the paper.

2. RESEARCH METHODOLOGY

2.1. Datasets

The framework utilizes two datasets: the Turtle1 dataset and the images.cv turtle dataset. The Turtle1 dataset comprises 2630 images of six native turtle species found in India, collected from online sources including GBIF (Global Biodiversity Information Facility), Indian Reptiles, Reptile Database, and iNaturalist. For non-commercial use, the dataset is available at <https://ieee-dataport.org/documents/turtle1>. Table 1 presents the class distribution of the Turtle1 dataset. It is a small-scale dataset and maintains a near-uniform distribution across all six classes to prevent model bias. The images.cv turtle dataset [22] consists of images of four different turtle species, as shown in Table 2. Both datasets consist of RGB images of varied sizes, and are further normalized and resized to 380×380 pixels.

The Turtle1 dataset was divided into three subsets comprising 2053 training images, 286 validation images, and 291 testing images. The partitioning maintains a balanced class distribution across all subsets. No duplicate or overlapping images are present between the training, validation, and testing sets. This prevents data leakage and ensures a fair evaluation of the proposed model.

Table 1. Class distribution of the Turtle1 dataset

Species	No. of samples
<i>Amyda cartilaginea</i> (Asiatic Softshell Turtle)	418
<i>Cuora amboinensis</i> (South Asian Box Turtle)	426
<i>Cyclemys dentata</i> (Asian Leaf Turtle)	431
<i>Melanochelys trijuga</i> (Indian Black Turtle)	466
<i>Nilssonia gangetica</i> (Indian Softshell Turtle)	459
<i>Pangshura tentoria</i> (Indian Tent Turtle)	430
TOTAL	2630

The images.cv turtle dataset [22] is a publicly available benchmark dataset containing 5191 images of four sea turtle species. The dataset includes images captured under different environmental and lighting conditions. It is divided into three subsets consisting of 3637 training images, 791 validation images, and 773 testing images.

Table 2. Class distribution of the images.cv turtle dataset

Species	No. of samples
<i>Terrapene carolina</i> (Florida Box turtle)	1300
<i>Dermochelys coriacea</i> (Leatherback turtle)	1300
<i>Caretta caretta</i> (Loggerhead turtle)	1296
<i>Kinosternon scorpioides</i> (Scorpion mud turtle)	1295
TOTAL	5191

2.2. Model Architecture

The overall architecture of the proposed TurtleNet model is illustrated in Figure 1. The different components of the model are described in the following subsections.

2.2.1. Preprocessing

Image preprocessing is performed to reduce the variations present in the input images before feature extraction [23]. The images in the dataset differ in size, resolution, lighting conditions, orientation, and background. In this study, image resizing and normalization are applied during preprocessing. All input images are resized to 380×380 pixels. Subsequently, the pixel values are normalized by rescaling them with a factor of $1/255$ so that the pixel intensities lie within the range of $[0, 1]$.

2.2.2. Feature Extraction

The proposed model employs a deep learning-based transfer learning approach, utilizing a customized EfficientNetB4 [24] as the backbone, and integrating an SE attention mechanism [25] for feature extraction. For transfer learning, the top 14 layers of EfficientNetB4 are fine-tuned, while the remaining layers are kept frozen during training to retain the benefits of pre-learned general features from the ImageNet dataset [23]. The input to this feature extraction stage comprises pre-processed RGB images of size $380 \times 380 \times 3$, as illustrated in Figure 1. The input passes through a series of 8 mobile inverted bottleneck convolution (MBConv) blocks, starting with the first convolution layer (i.e., the stem convolution). The low-level feature maps are formed with the stem convolution, followed by the formation of mid-level features using all the MBConv blocks. These blocks progressively reduce spatial resolution while increasing channel depth, effectively extracting both low-level and mid-level semantic feature representations, as shown in Figure 2.

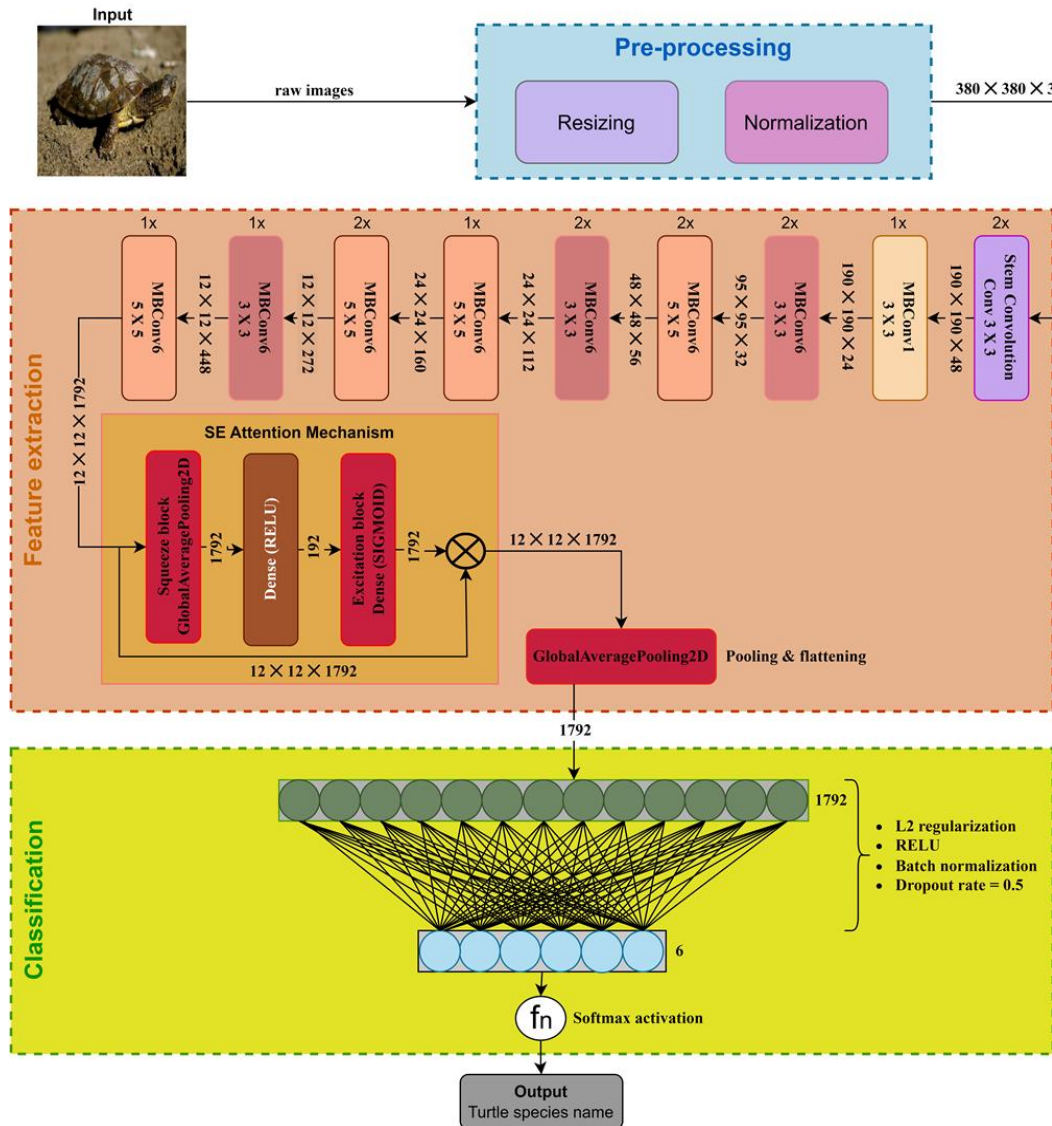


Figure 1. The detailed architectural pipeline of the proposed model

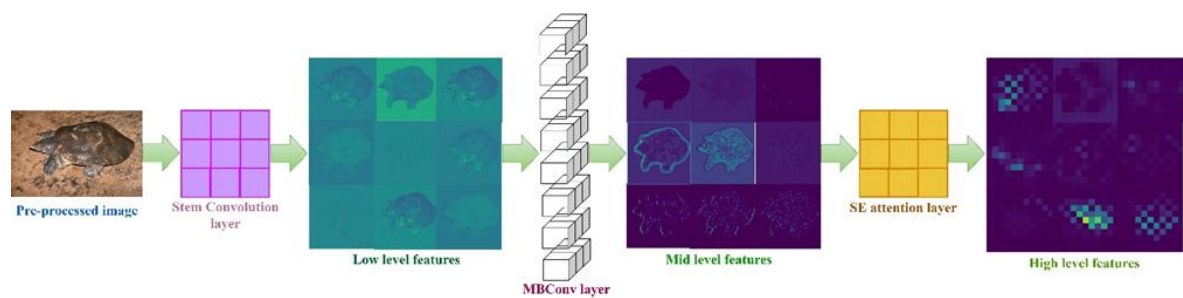


Figure 2. Layer-wise feature maps generated by TurtleNet

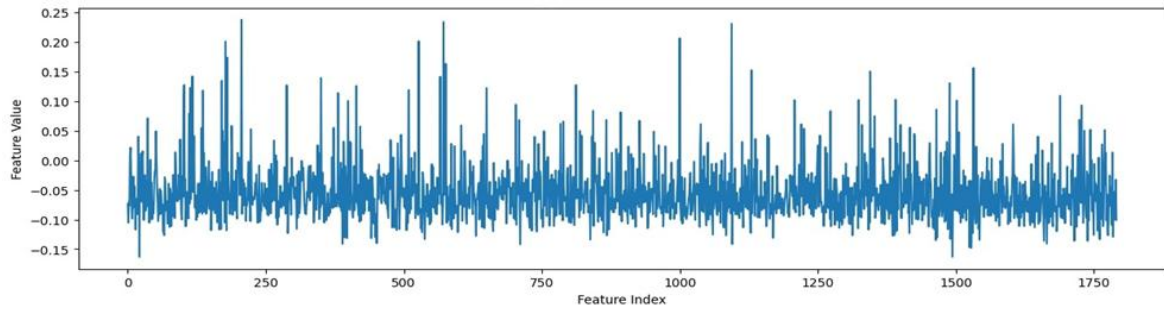


Figure 3. Distribution of flattened feature vectors

As shown in Figure 1, a custom Squeeze and Excitation (SE) block is placed after the final convolutional block of the EfficientNetB4 backbone and before the global average pooling layer. At this stage, the feature maps contain rich spatial and semantic information extracted by the backbone network. The SE block recalibrates the channel-wise feature responses by assigning higher weights to informative channels and lower weights to less informative ones. The refined feature maps are then forwarded to the subsequent classification stage. During the squeeze operation, global average pooling is applied to the mid-level feature maps, producing an output of vector size 1792. The excitation operation takes this vector as input and processes it through a dense layer with ReLU activation, followed by a sigmoid-activated layer, producing an output of the same dimensionality. The attention mechanism is completed by scaling the original $12 \times 12 \times 1792$ feature map with the excitation output, producing refined high-level feature maps, as shown in Figure 2. Following the SE attention layer, global average pooling is applied to reduce the spatial dimensions, resulting in a one-dimensional (1D) feature vector of size 1792 for classification. The feature vectors produced during this stage are illustrated using Figure 3.

2.2.3. Classification

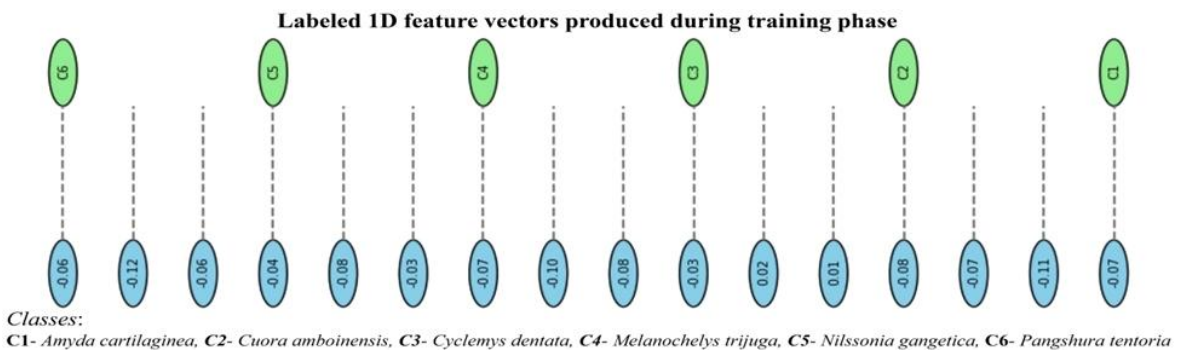


Figure 4. Visualization of class probability scores during classification

During the classification phase, the feature vectors obtained from the extraction stage are passed through a feedforward neural network (FFNN) comprising 256 dense units. A ReLU activation function is employed to introduce non-linearity, while L2 regularization is applied to mitigate overfitting. Batch normalization is applied after the dense layer to stabilize the training process. A dropout layer with a dropout rate of 0.5 is used to reduce overfitting. Finally, the output is passed through a softmax activation layer to obtain the probability scores for all turtle species. The class probability distribution is illustrated in Figure 4.

The Softmax function [10] used in the classification phase is defined as:

$$S_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad (1)$$

where z_j denotes the input score (logit) for class j , K is the total number of classes, and S_j represents the predicted probability for class j .

2.3. Ensemble Approach

An ensemble learning strategy is used to improve the classification performance of TurtleNet. TurtleNet and YOLOv8 [19] are trained separately and their predictions are combined during inference using a soft voting strategy. Previous studies have reported that ensemble learning often achieves better performance than individual models [26]. The classification variant of YOLOv8 provides an efficient backbone for feature extraction in image classification tasks [19].

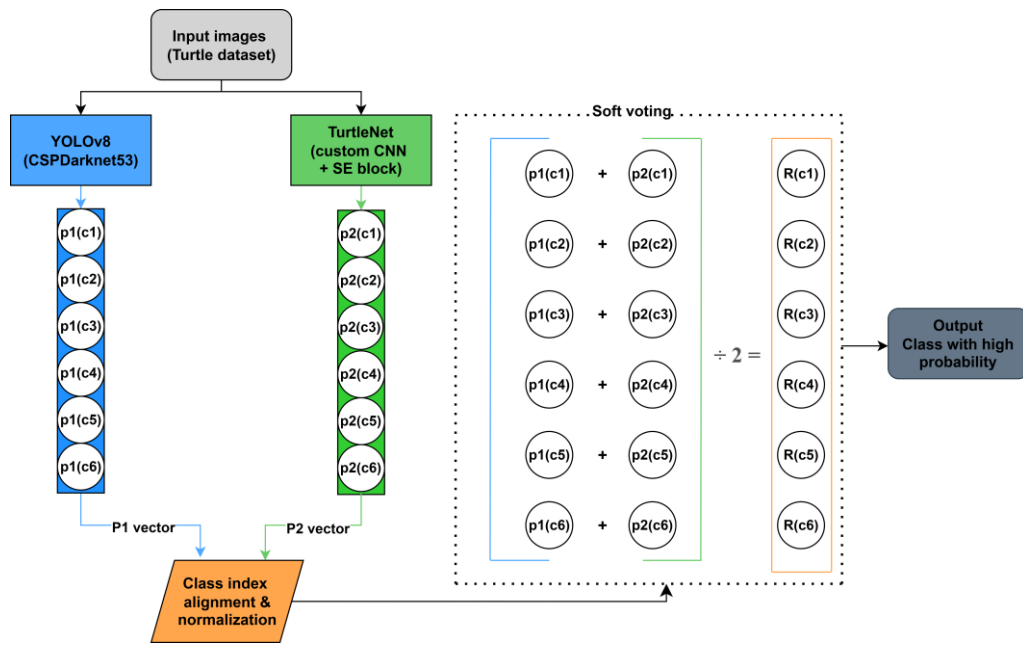


Figure 5. The proposed ensemble architecture

Figure 5 shows the class probability alignment and fusion process of the proposed ensemble framework. TurtleNet and YOLOv8 independently generate class probability vectors for each input image. The probability vectors are aligned using the same class order before fusion. The probability values are then normalized to maintain a common scale across both models. This prevents one model from dominating the fusion process because of differences in probability distributions. A soft voting strategy with equal weights is then applied to obtain the final prediction. The final class label is selected based on the highest combined probability. The class probability for class c is computed as:

$$P(c) = \frac{1}{n} \sum_{i=1}^n p_i(c) \quad (2)$$

where $P(c)$ denotes the final aggregated probability for class c , n is the number of models, and $p_i(c)$ represents the probability predicted by the i^{th} model for class c .

3. RESULTS AND DISCUSSION

The performance of the proposed TurtleNet is compared with eleven CNN-based classification models. These models are used as baseline models for comparison without additional fine-tuning. All models are trained and tested using the Turtle1 and images.cv turtle datasets. The performance comparison is presented in Table 3. Brief descriptions of the selected models are given below.

- InceptionV3 [13]: Employs factorized convolutions to improve computational efficiency and classification performance.
- ResNet101 [14]: A deep residual network that employs identity shortcut connections to facilitate efficient gradient propagation during training.
- MobileNetV3 [15]: A lightweight and computationally efficient CNN architecture designed for mobile and embedded vision applications.
- ConvNeXt-Base [16]: A modern CNN architecture inspired by transformer design principles and optimized for high image classification performance.
- DenseNet169 [17]: A densely connected CNN architecture where each layer receives feature maps from all preceding layers, improving feature reuse and gradient flow.
- VGG16 [18]: A classical CNN architecture that employs stacked convolutional layers followed by fully connected layers for image classification.
- YOLOv8 [19]: Primarily designed for object detection and supports image classification through its classification head and multi-scale feature extraction capability.

- Inception-ResNet-V2 [20]: Combines Inception modules with residual connections to construct a deep and computationally efficient network architecture.
- ResNet152V2 [14]: A very deep residual network that utilizes identity shortcut connections to facilitate stable training of deep architectures.
- Xception [21]: A deep CNN architecture that improves feature extraction efficiency using depthwise separable convolutions.
- VGG19 [18]: An extension of VGG16 with additional convolutional layers, providing increased representational capacity.

Table 3. Performance comparison of the models used in the study

Models	Turtle1 Dataset				images.cv Turtle Dataset			
	Accuracy (%)	Precision	Recall	F1 Score	Accuracy (%)	Precision	Recall	F1 Score
InceptionV3 [13]	88.85	0.92	0.89	0.89	90.82	0.91	0.91	0.91
ResNet101 [14]	90.88	0.92	0.91	0.91	86.93	0.87	0.87	0.87
MobileNetV3 [15]	92.44	0.94	0.92	0.92	24.71	0.25	0.25	0.25
ConvNeXt-Base [16]	84.88	0.86	0.85	0.85	80.72	0.81	0.81	0.81
DenseNet169 [17]	94.50	0.95	0.94	0.94	88.62	0.89	0.89	0.89
VGG16 [18]	93.24	0.93	0.93	0.93	78.91	0.79	0.79	0.79
YOLOv8 [19]	95.88	0.96	0.96	0.96	82.28	0.82	0.82	0.82
Inception-ResNet-V2 [20]	88.51	0.89	0.88	0.89	91.33	0.91	0.91	0.91
ResNet152V2 [14]	90.38	0.92	0.90	0.90	85.77	0.86	0.86	0.86
Xception [21]	91.07	0.92	0.91	0.91	90.56	0.91	0.90	0.90
VGG19 [18]	91.22	0.92	0.91	0.91	82.41	0.82	0.82	0.82
TurtleNet (Proposed)	97.25	0.97	0.97	0.97	91.46	0.91	0.91	0.91

As shown in Table 3, TurtleNet achieved the highest classification accuracy on the Turtle1 dataset. It also achieved precision, recall and F1-score values of 0.97. The high classification performance can be attributed to the combination of the EfficientNetB4 backbone and the SE attention block. EfficientNetB4 uses a compound scaling strategy to balance the network depth, width, and input resolution while maintaining parameter efficiency [27]. The SE attention block further enhances the extracted feature maps by assigning higher weights to important feature channels. This helps the model distinguish subtle morphological differences among turtle species and improves the overall classification performance.

A similar trend is observed for the images.cv turtle dataset. As shown in Table 3, TurtleNet achieved the highest classification accuracy of 91.46%. The model also achieved precision, recall and F1-score values of 0.91. It consistently outperformed the other deep learning models. These results indicate that TurtleNet can effectively classify turtle species under different conditions.

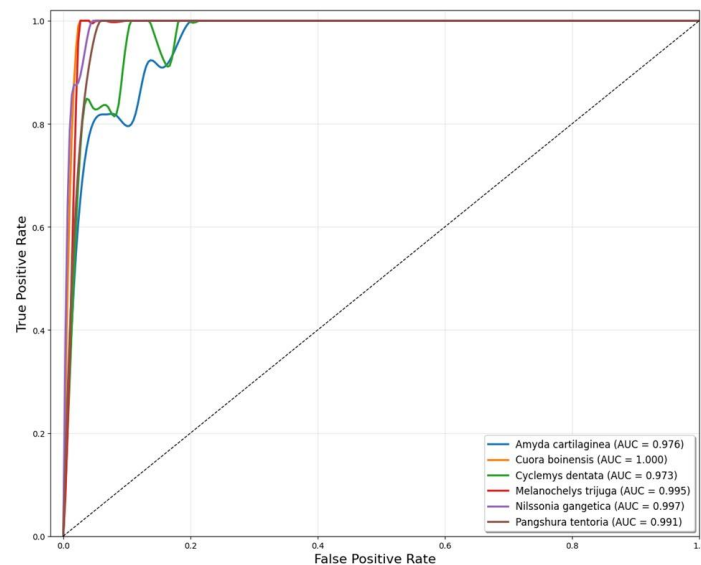


Figure 6. Multiclass ROC curve of the proposed TurtleNet model on the Turtle1 dataset

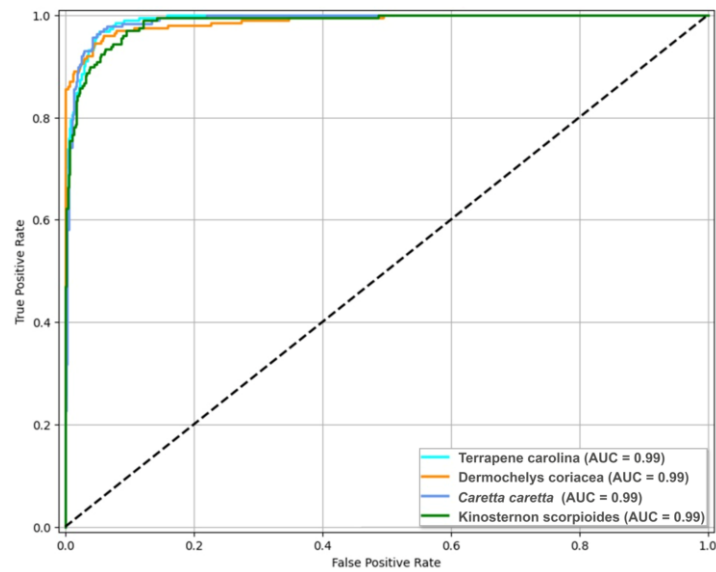


Figure 7. Multiclass ROC curve of the proposed TurtleNet model on the images.cv turtle dataset

Figure 6 and Figure 7 present the ROC curves and AUC scores of TurtleNet on the Turtle1 and images.cv turtle datasets. TurtleNet achieved an AUC score of 0.99 for all classes on the images.cv turtle dataset. For the Turtle1 dataset, the AUC scores ranged from 0.973 to 1.00. The high AUC scores demonstrate the effectiveness of the proposed model in classifying different turtle species.

3.1. Analysis of Training and Validation Performance

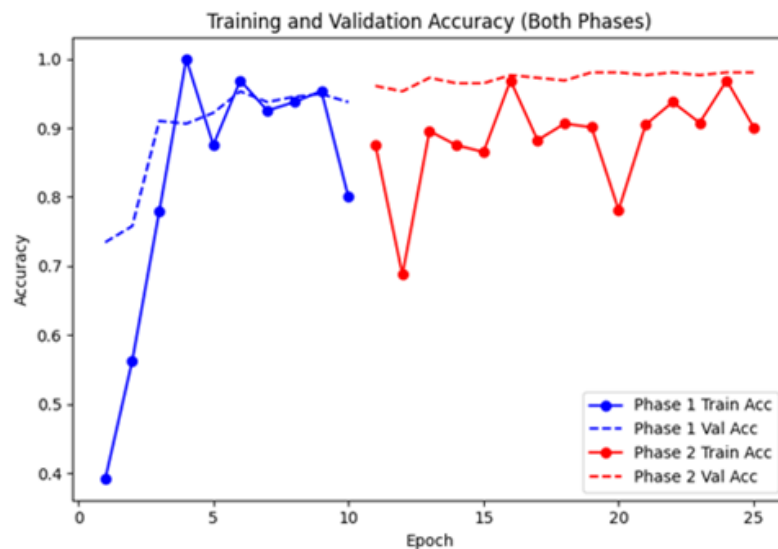


Figure 8. Training and validation accuracy curves of TurtleNet on the Turtle1 dataset

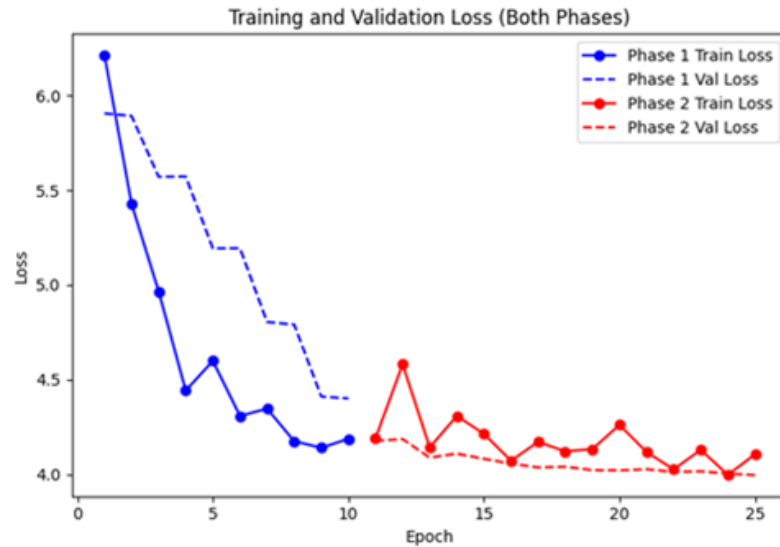


Figure 9. Training and validation loss curves of TurtleNet on the Turtle1 dataset

Figure 8 and Figure 9 show the training and validation accuracy and loss curves of TurtleNet on the Turtle1 dataset during the two training phases. In Phase 1 (blue), the EfficientNetB4 backbone learns the image features and achieves more than 90% validation accuracy after a few epochs. The training and validation loss also decrease continuously during this phase. In Phase 2 (red), the SE attention block further improves the feature representation. As a result, the validation accuracy becomes more stable and only minor variations are observed in the loss curves. The training and validation curves remain close throughout the training process, indicating good generalization with minimal overfitting.

Figure 10 and Figure 11 show the training and validation accuracy and loss curves of TurtleNet on the images.cv turtle dataset. The validation accuracy remains consistently high throughout both training phases. The training and validation loss decrease gradually with increasing epochs. The close agreement between the training and validation curves suggests good generalization with minimal overfitting.

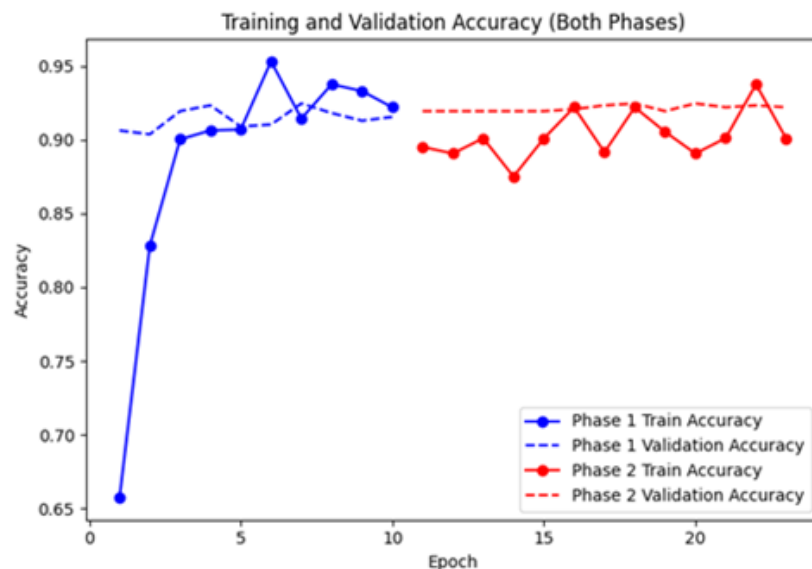


Figure 10. Training and validation accuracy curves of TurtleNet on the images.cv turtle dataset

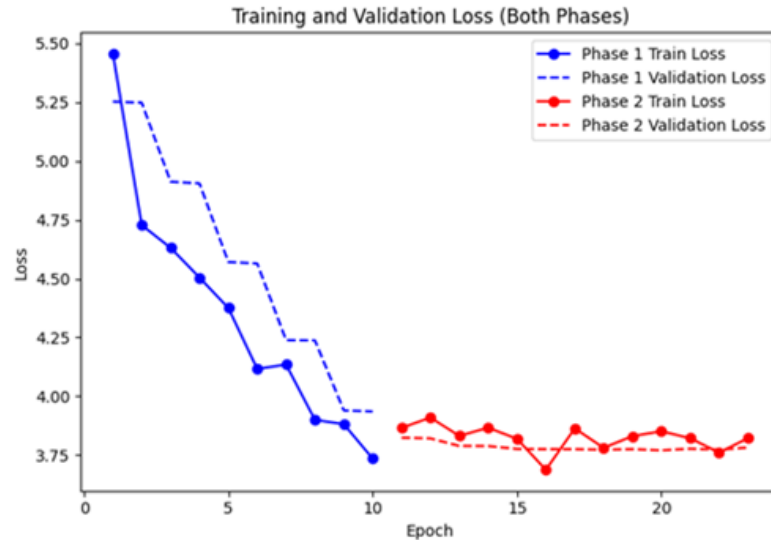


Figure 11. Training and validation loss curves of TurtleNet on the images.cv turtle dataset

Overall, the training and validation curves shown in Figure 8 to Figure 11 remain consistent for both datasets. The validation accuracy remains high, while the training and validation loss decrease gradually during the training process. These results show that TurtleNet learns effectively and performs well on both datasets.

3.2. Computational Complexity Analysis of the SE Attention Block

Table 4. Computational complexity comparison of TurtleNet

Metric	With SE block	Without SE block	Relative increase
Parameters	18,085,989	17,684,581	2.27%
Inference Time (ms/image)	95.47	93.33	2.29%

Table 4 presents the computational complexity of TurtleNet with and without the SE attention block. The SE attention block increases the parameter count by only 2.27% and the inference time from 93.33 ms/image to 95.47 ms/image. Although the computational cost increases slightly, the proposed model achieves better classification performance. This shows that the SE attention block improves the effectiveness of TurtleNet with only a marginal computational overhead.

3.3. Computational Analysis and Trade-offs in Ensembling

The ensemble model is evaluated on the Turtle1 dataset. Table 5 compares the performance of YOLOv8 (M1), TurtleNet (M2) and the ensemble model. The ensemble model achieved the highest classification accuracy of 99.66%. It also achieved precision, recall and F1-score values of 0.99. The ensemble model improved the classification accuracy by 3.78% over YOLOv8 and 2.41% over TurtleNet. The improved performance can be attributed to the aggregation of class probability vectors generated by TurtleNet and YOLOv8 through a soft voting strategy.

Table 5. Performance comparison of YOLOv8, TurtleNet, and ensemble model on the Turtle1 dataset

Models	Accuracy (%)	Precision	Recall	F1 Score
M1 (YOLOv8)	95.88	0.96	0.96	0.96
M2 (TurtleNet)	97.25	0.97	0.97	0.97
Ensemble (M1 + M2)	99.66	0.99	0.99	0.99

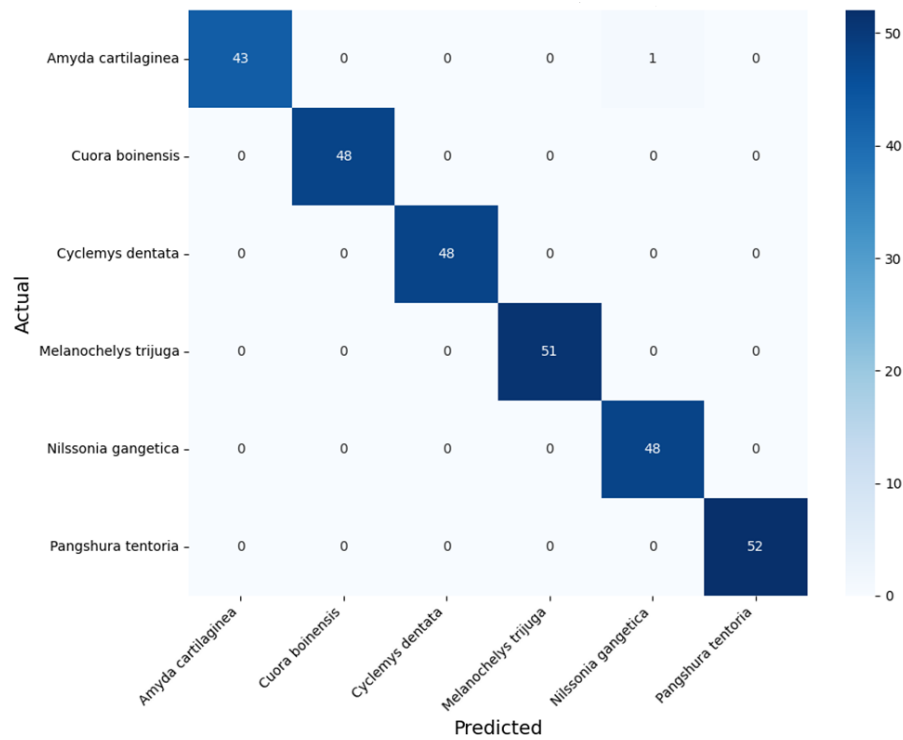


Figure 12. Confusion matrix of the proposed ensemble model

Figure 12 shows the confusion matrix of the proposed ensemble model on the Turtle1 testing dataset. The ensemble model correctly classified 290 out of 291 testing samples, with only one misclassification. This indicates that the proposed ensemble model can accurately distinguish turtle species with similar morphological characteristics.

Table 6. Computational complexity metrics comparison for TurtleNet, YOLOv8, and the ensemble model

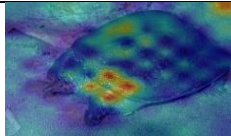
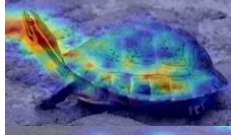
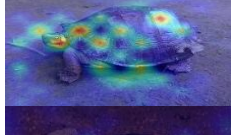
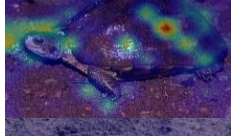
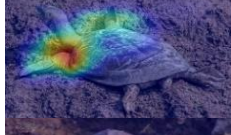
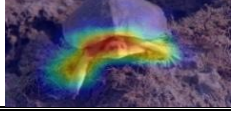
Model	Parameters (Millions)	FLOPs (G)	Inference time (ms/image)	GPU Memory Usage (MB)
M1 (YOLOv8)	2.72	0.21	3.02	43.02
M2 (TurtleNet)	18.54	9.02	547.36	1952.98
Ensemble (M1 + M2)	~21.26	~9.23	~550.38	~1996

Table 6 compares the computational complexity of YOLOv8, TurtleNet, and the ensemble model. The ensemble model achieved higher classification accuracy than the individual models. However, it also increased the parameter count, FLOPs, inference time, and GPU memory usage. TurtleNet achieved high classification accuracy with moderate computational cost, while YOLOv8 required lower computational resources. The ensemble model improved the overall classification performance at the cost of additional computational overhead. Therefore, the proposed ensemble model is suitable for applications where classification accuracy is the primary requirement. For resource-constrained or real-time applications, the individual models may be a more suitable choice.

3.4. Grad-CAM-Based Visualization for Model Interpretability

Grad-CAM visualizations were generated for correctly classified samples of six turtle species to interpret the discriminative regions utilized by TurtleNet during prediction. As shown in Table 7, TurtleNet focuses on important morphological regions such as the head and neck stripe patterns, carapace scute arrangements, softshell textures, and limb morphology. These regions correspond to the characteristic features of different turtle species.

Table 7. Original images and corresponding Grad-CAM visualizations generated by TurtleNet

Species	Correctly Classified Image	Grad-CAM Visualization
<i>Amyda cartilaginea</i>		
<i>Cuora amboinensis</i>		
<i>Cyclemys dentata</i>		
<i>Melanochelys trijuga</i>		
<i>Nilssonia gangetica</i>		
<i>Pangshura tentoria</i>		

For *Amyda cartilaginea*, the activations were concentrated over the carapace surface and dorsal spots; *Cuora amboinensis* exhibited strong attention around the head and anterior shell regions; and *Cyclemys dentata* showed prominent responses along the shell geometry and marginal edges. Similarly, *Melanochelys trijuga* emphasized shell and head regions containing characteristic patterns, while *Nilssonia gangetica* demonstrated activations around the neck and forelimb regions with distinctive skin textures. In the case of *Pangshura tentoria*, the model focused primarily on facial and limb regions associated with distinctive shape features. Across several samples, minor activations were also observed in surrounding soil, rock, or ground regions. Such peripheral attention may arise from contextual correlations present in the training data or from the spatial diffusion behavior of Grad-CAM visualizations, rather than direct dependence on background information. Overall, the Grad-CAM results demonstrate that TurtleNet predominantly relies on biologically relevant morphological traits for species-level classification.

4. CONCLUSION

This study presented TurtleNet, an explainable deep learning framework for turtle species classification using an attention-enhanced EfficientNetB4 architecture with Grad-CAM visualization. The framework integrates transfer learning with a Squeeze-and-Excitation (SE) attention block to improve feature extraction for morphologically similar turtle species. The proposed framework was evaluated on the Turtle1 dataset and the benchmark images.cv turtle dataset. TurtleNet achieved classification accuracies of 97.25% and 91.46% on the Turtle1 and images.cv turtle datasets, respectively. The proposed model also outperformed several existing deep learning models.

The SE attention block improved the classification performance with only a small increase in computational cost. Grad-CAM visualizations highlighted the image regions used by the model during prediction. Most of these regions corresponded to important morphological characteristics such as the shell structure, head pattern, and limb features. The training and validation curves remained consistent across both datasets and indicated good generalization with minimal overfitting.

An ensemble framework was further developed by combining TurtleNet and YOLOv8 using a soft voting strategy. The ensemble achieved a classification accuracy of 99.66% on the Turtle1 dataset. Although the ensemble increased the computational cost, it provided better classification performance than the individual

models. Future work will focus on incorporating Vision Transformers, exploring additional ensemble strategies, and evaluating the framework on more diverse turtle datasets to further improve its robustness and generalization capability.

5. CODE AVAILABILITY

The implementation code for the proposed TurtleNet framework and associated experimental procedures is publicly available at: <https://github.com/7RajkumarB/Turtle-Species-Classification>.

REFERENCES

- [1] A. G. J. Rhodin *et al.*, "Global Conservation Status of Turtles and Tortoises (Order Testudines)," *Chelonian Conservation and Biology*, vol. 17, no. 2, p. 135, Dec. 2018, doi: 10.2744/CCB-1348.1.
- [2] H. Nagashima, S. Kuraku, K. Uchida, Y. Kawashima-Ohya, Y. Narita, and S. Kuratani, "Body plan of turtles: an anatomical, developmental and evolutionary perspective," *Anat Sci Int*, vol. 87, no. 1, pp. 1–13, Mar. 2012, doi: 10.1007/s12565-011-0121-y.
- [3] S. Singh, S. Badola, and M. Fernandes, "The Identification (ID) Cards: Tortoises and Freshwater Turtles of India." TRAFFIC, India Office, 2023. Accessed: Feb. 23, 2025. [Online]. Available: https://www.fin.awsassets.panda.org/downloads/tortoise_and_freshwater_turtles_id_cards_2023.pdf
- [4] J. Wyneken, K. J. Lohmann, and J. A. Musick, Eds., "Predators, Prey, and the Ecological Roles of Sea Turtles," in *The Biology of Sea Turtles, Volume III*, 0 ed., CRC Press, 2013, pp. 268–303. doi: 10.1201/b13895-15.
- [5] R. Faurina, A. Wijanarko, A. F. Heryuanti, S. I. Ishak, and I. Agustian, "Comparative study of ensemble deep learning models to determine the classification of turtle species," *Comput. Sci. Inf. Technol.*, vol. 4, no. 1, pp. 24–32, Mar. 2023, doi: 10.11591/csit.v4i1.p24-32.
- [6] J.-W. Baek, J.-I. Kim, and C.-B. Kim, "Deep learning-based image classification of sea turtles using object detection and instance segmentation models," *PLoS ONE*, vol. 19, no. 11, p. e0313323, Nov. 2024, doi: 10.1371/journal.pone.0313323.
- [7] J.-W. Baek, J.-I. Kim, and C.-B. Kim, "Deep learning-based image classification of turtles imported into Korea," *Sci Rep*, vol. 13, no. 1, p. 21677, Dec. 2023, doi: 10.1038/s41598-023-49022-3.
- [8] P. Sukyen, K. Chuiad, and P. Chaisiriprasert, "Sea Turtles Species Classification With a Modified ResNet-50 Deep Learning Model," in *2024 8th International Conference on Information Technology (InCIT)*, Chonburi, Thailand: IEEE, Nov. 2024, pp. 554–559. doi: 10.1109/InCIT63192.2024.10810619.
- [9] W. Songpan, T. Chaianunporn, K. Lomthaisong, and S. Keithmaleesatti, "Automated classification in turtles genus Malayemys using ensemble multiview image based on improved YOLOv8 with CNN," *Sci Rep*, vol. 14, no. 1, p. 24993, Oct. 2024, doi: 10.1038/s41598-024-76431-9.
- [10] J. Liu, M. Wang, L. Bao, X. Li, J. Sun, and Y. Ming, "Classification and Recognition of Turtle Images Based on Convolutional Neural Network," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 782, no. 5, p. 052044, Mar. 2020, doi: 10.1088/1757-899X/782/5/052044.
- [11] W. R. Da Paixao, T. M. Paixao, M. C. B. Da Costa, J. O. Andrade, F. G. Pereira, and K. S. Komati, "Texture Classification of Sea Turtle Shell Based on Color Features: Color Histograms and Chromaticity Moments," *IJAIA*, vol. 9, no. 2, pp. 55–67, Mar. 2018, doi: 10.5121/ijaia.2018.9205.
- [12] U. Mendiratta, V. Sheel, and S. Singh, "Enforcement seizures reveal large-scale illegal trade in India's tortoises and freshwater turtles," *Biological Conservation*, vol. 207, pp. 100–105, Mar. 2017, doi: 10.1016/j.biocon.2017.01.023.
- [13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 2015, *arXiv*. doi: 10.48550/ARXIV.1512.00567.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2015, *arXiv*. doi: 10.48550/ARXIV.1512.03385.
- [15] A. Howard *et al.*, "Searching for MobileNetV3," 2019, *arXiv*. doi: 10.48550/ARXIV.1905.02244.
- [16] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," 2022, *arXiv*. doi: 10.48550/ARXIV.2201.03545.
- [17] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," 2016, *arXiv*. doi: 10.48550/ARXIV.1608.06993.
- [18] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2014, *arXiv*. doi: 10.48550/ARXIV.1409.1556.
- [19] R. Varghese and S. M., "YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness," in *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, Chennai, India: IEEE, Apr. 2024, pp. 1–6. doi: 10.1109/ADICS58448.2024.10533619.
- [20] D. Yuehua and P. Huilin, "Improved garbage image classification for inception-Resnet-V2," in *2022 Asia Conference on Algorithms, Computing and Machine Learning (CACML)*, Hangzhou, China: IEEE, Mar. 2022, pp. 477–481. doi: 10.1109/CACML55074.2022.00087.
- [21] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," 2016, *arXiv*. doi: 10.48550/ARXIV.1610.02357.
- [22] image.cv, "Sea turtle Labeled Image Dataset." image.cv, Sea turtle. Accessed: Feb. 12, 2025. [Jpg]. Available: <https://images.cv/dataset/sea-turtle-image-classification-dataset>

- [23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. Burges, L. Bottou, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
- [24] C.-Y. Zhu *et al.*, "A Deep Learning Based Framework for Diagnosing Multiple Skin Diseases in a Clinical Environment," *Front. Med.*, vol. 8, p. 626369, Apr. 2021, doi: 10.3389/fmed.2021.626369.
- [25] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT: IEEE, Jun. 2018, pp. 7132–7141. doi: 10.1109/CVPR.2018.00745.
- [26] A. Mohammed and R. Kora, "A comprehensive review on ensemble deep learning: Opportunities and challenges," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 2, pp. 757–774, Feb. 2023, doi: 10.1016/j.jksuci.2023.01.014.
- [27] J. Jareño, G. Bárcena-González, J. Castro-Gutiérrez, R. Cabrera-Castro, and P. L. Galindo, "Automatic labeling of fish species using deep learning across different classification strategies," *Front. Comput. Sci.*, vol. 6, p. 1326452, Feb. 2024, doi: 10.3389/fcomp.2024.1326452.

BIOGRAPHY OF AUTHORS



Pragyat Jyoti Baruah received the Bachelor of Technology, and Master of Technology degree in Computer Science and Engineering from Assam University, Silchar. Currently, he is a Research Scholar in the department of Computer Science and Engineering at Assam University, Silchar. His area of research includes Machine Learning, Artificial Intelligence, Natural Language Processing, Computer Vision, Data Science, GIS and Remote Sensing.



Bugge Prince Solomon Raj Kumar received the Bachelor of Technology degree in Computer Science and Engineering from Assam University, Silchar. His research interests include Deep Learning, Computer Vision, and Machine Learning. He is passionate about leveraging innovative technologies for the betterment of society and is enthusiastic about research and continuous learning in emerging fields.



Arun Jyoti Nath received his Doctorate of Philosophy from the department of Ecology and Environmental Science. He received his Post-Doctoral degree from The Ohio State University, USA. He is a Professor in the department of Ecology and Environmental Science at Assam University, Silchar. His area of research includes Forest Ecology and Management, Agroforestry Systems, Soil Ecology, Adaptation and Mitigation Strategy for Climate Change, Biocarbon Sequestration. He was the recipient of DST Young Scientist, DST National Associateship, DBT Overseas Associateship, DBT Unit of Excellence awards. Notably, he has made significant contribution to the field, as evidenced by his multiple research articles published in prestigious journals indexed by Scopus and SCI. He is also recognized under the 'top 2% scientists in the world', according to a global survey published by Elsevier in 2025.



Arnab Paul received his Master of Technology from the department of Computer Science and Engineering degree at Tezpur University. He earned his Doctorate of Philosophy from Assam University, Silchar. Since 2011, he has been serving as an Assistant Professor in the department of Computer Science and Engineering at Assam University, Silchar. His research interests include Machine Learning, Artificial Intelligence, Natural Language Processing, and Trust & Reputation Models. He has published several research papers in reputed journals and conferences and has contributed to various externally funded research projects.



Jarine Anjum received the Bachelor in Technology degree in Computer Science and Engineering from Assam University, Silchar. Her research interests include Machine Learning, Deep Learning, Computer Vision, Image Processing, and Data Science. She worked as an intern in a Web Development project at Jadavpur University, Kolkata during 2024. She holds a certification from the 2nd IEEE CIS Summer School on Contemporary Advancement in Multi Objective Optimization (CAMOO), 2024.



Akash Nath is currently pursuing the Bachelor of Technology in Computer Science and Engineering at Assam University, Silchar. His research interests include Artificial Intelligence, Machine Learning, Computer Vision, and Natural Language Processing, with a focus on developing practical AI solutions for real-world challenges.



Tirthanka Borah received the Bachelor of Technology, and Master of Technology degree in Computer Science and Engineering from Assam University, Silchar. Currently, he is a Research Scholar in the department of Computer Science and Engineering at Assam University, Silchar. His area of research includes Computer Vision, Natural Language Processing, GIS and Remote Sensing.