

A Comparative Evaluation Of Deep Learning Architectures For Butterfly Species Classification In Natural Scene Images

Manjuchethan N^{1*}, H S Nagendraswamy¹, Somanna M¹, Thejashwini B L¹, Santhosh S²,
Chethana Kumara B M³

^{1*}Department of Studies in Computer Science, University of Mysore, Mysuru, India. [0009-0005-2003-1538], [0009-0001-2614-6169], [0000-0001-6186-5225], [0009-0005-7273-8754],
manjuchethan@gmail.com

²Department of Studies in Zoology, University of Mysore, Mysuru, India.
[0000-0002-0384-1636],

^{3*}Samsung Electro-mechanics, Bangalore, India. [0000-0003-0061-3651]

Abstract: Automated ecological species identification is a subject of intense study in computer vision research, which plays an important role in biodiversity monitoring, research, and conservation planning. The research work presented in this article introduces a large custom butterfly dataset consisting of 23537 images belonging to 102 different species, collected through extensive field photography and supplemented with selected images from reliable online sources. Furthermore, the dataset is preprocessed with annotations to generate a Region of Interest (RoI). A total of 54 state-of-the-art deep learning architectures, including classical Convolution Neural Networks (CNN), Modern CNN, Lightweight CNN models, attention-based models, and Vision Transformers, were evaluated on both segmented and non-segmented images of the dataset. A comparative analysis of all the deep learning models in terms of accuracy, precision, recall and F1-Score has been done. The non-uniform class distribution in the dataset motivates the use of weighted evaluation metrics along with overall accuracy, ensuring an impartial evaluation across all categories. Results indicate that modern CNN architectures consistently outperform other models on both segmented and non-segmented images in the dataset. In particular, ConvNeXt-Tiny achieved the best performance of a weighted F1-Score of 90.88% on segmented images and 84.8% on non-segmented images. This proves the effectiveness of object-focused preprocessing and demonstrates the superiority of modern CNN architectures for fine-grained butterfly species classification.

Keywords: Butterfly Species classification, Natural Scene Images, Segmented and Non-Segmented butterfly images, Deep Learning, Transfer Learning, Biodiversity Monitoring.

1. INTRODUCTION

Butterflies are the most attractive flying insects that belong to the group Lepidoptera with brightly colored wings. They play a significant role in the field of biological science, agriculture, and forest conservation by facilitating the reproduction of numerous plant species through pollination. The butterfly population indicates the health of the ecosystem. Due to inadequate safeguarding, some species have gone extinct, and conserving butterflies is crucial to protect the ecosystem.

Identification of butterfly species in the traditional approach is a challenging task that requires laborious work from zoologists and microscopists. Thus, environmentalists, naturalists, and researchers are showing interest in Fine-Grained Visual Butterfly Identification and Classification.

The development of artificial intelligence over the past decades has attracted researchers to the identification of animals, birds and insects using computer vision approaches. This encourages the researchers and motivates them towards the identification of butterfly species using natural language processing [1], deep learning [2], and machine learning approaches with a dataset of lower complexity. However, computer vision researchers have intensified their efforts in the identification and classification of butterflies using machine learning and deep learning approaches, which are failing to achieve satisfactory results due to a lack of appropriately labelled datasets from experts.

The literature review provides a complete overview of existing research conducted in the field of butterfly identification and classification. The following paragraphs present a few interesting and state-of-the-art works carried out by contemporary researchers in this direction.

Josiah Wang et al. [1] proposed learning models for object recognition from natural language descriptions. The study examines the use of NLP in visual object identification for detailed object categories, such as animal and plant species. A generative model is used to generate textual terms learnt from templates of visual attributes. Experiments comparing human performance with the proposed method on a dataset of 10 butterfly categories showed moderate accuracy, comparable to that of non-negative English speakers in the human experiment. This approach achieved an overall balanced accuracy of 72% for native English speakers and 51% for non-native English speakers. Seung-Ho Kang et al. [2] proposed a novel back-propagation convolutional neural network for the identification of butterfly species. This approach utilises branch length similarity entropy (BLS) as an input feature, focusing on butterfly wing shapes. The proposed system outperformed on binary images as inputs, guaranteeing consistent feature sizes and robustness to image variations, achieving 86% accuracy with a BLS entropy network on a dataset of 268 specimen images from seven butterfly species and 540 images of 15 butterfly classes. Nur Nabila Kamaron Arzar et al. [3] proposed a method of identifying butterfly species using a convolutional neural network with a pre-trained architecture, focusing on the GoogleNet model. Experiments were conducted on a dataset that comprises only 120 images with 4 butterfly species commonly found in Asia, and achieved a classification accuracy of 97.5%. The study highlights the importance of noise in the samples and the scalability of the sample size within the dataset. Ruoyan Zhao et al. [4] developed a butterfly recognition system on a dataset of 5,695 butterfly images, comprising 4,270 specimen images of 111 species and 1,425 ecological photos of 17 species, highlighting the challenges of intra-class and inter-class similarity. The study employed the Faster R-CNN (Region-based Convolutional Neural Network) approach for the automatic identification of butterflies, achieving a classification accuracy of approximately 70.4% on ecological photos of butterfly species. Hari Theivaprakasham [5] introduced a method to identify Indian butterflies using a deep convolutional neural network-based architectures. Experiments were conducted on a newly created dataset of 315 species with 34024 images collected through self-photography, the Flickr API, and the Google search engine. The challenges of the conventional method, such as wing-venation, colour, and shape are discussed. Applied data augmentation techniques using the fastai Python library function, then experimented on pre-trained deep architectures, and classification results are tabulated using the model's top k (k=1,3,5) most confident predictions, which provide probabilistic results. Aijiao Tan et al. [6] proposed a rapid fine-grained classification of butterflies using the FCM-KM and Mask R-CNN techniques, experimenting with five different species of butterflies and a total sample of 1,188 images. An adaptive image enhancement algorithm based on fuzzy sets, optimised by the Fruit Fly optimisation algorithm, which improves the quality of the image before classification. The FCM-KM method selects the optimal clustering numbers and centers, whereas Mask R-CNN fusion extracts and classifies butterfly images. This technique outperforms existing algorithms such as CNN (Convolutional Neural Network), ResNet, and Mask R-CNN, with a recognition accuracy of 83.62%. This research highlights the expansion of the dataset to enhance the generalisation ability. Ayad Saad Almryad et al. [7] introduced the automatic identification of field butterflies by selected variants of convolutional neural networks. VGG16, VGG19, and ResNet50 were employed to classify butterfly species. Used a dataset of 44,659 images of 104 classes of butterfly species from the Butterfly Monitoring and Photography Society, collected in Turkey. Experimented on

a selected species with 17,769 images of 10 classes of the butterfly due to fewer samples. Achieved a testing accuracy of 79.5% from VGG-16. Fathimathul Rajeena P. P. et al. [8] The research aims to recognise butterfly species using image-based identification with a transfer- learning neural network approach, using a dataset of 10,035 images from 75 butterfly classes collected from Kaggle. Considering variations in orientation, angle, size, occlusion, and complex background, and to avoid overfitting, data augmentation has been performed to increase the sample size and improve the model's generalisation. VGG16, VGG19, MobileNet, Xception, ResNet50, and InceptionV3 with transfer learning architectures are used to evaluate precision, recall, F-measure, and accuracy. Claims that Inception V3 achieves the highest accuracy of 94.66 %, outperforming all other models. The study highlights the need for further research on a detailed taxonomy in butterfly species classification. Halit Çetiner [9] proposed a novel lightweight CNN for automatic classification of butterfly species, aiming for a reliable, fast, and cost-effective system. The dataset consists of 10 classes of butterfly species, consisting of 832 images. Experimentation was conducted with and without data augmentation, yielding validation accuracies of 93.41% and 91.72%, respectively, highlighting the importance of a balanced dataset. Rumana Yasmin et al. [10] proposed butterfly detection and classification techniques: a review of various methods for classifying and detecting butterflies, focusing on deep learning, machine learning, and artificial neural network models, which were collected from research information from 2012 to 2020 in the identification of butterfly species. It also describes potential avenues for further research. Diniati Ruaika et al. [11] proposed an ensemble learning-based ResNet-50 for butterfly classification to overcome the lower accuracy obtained with architectures such as the basic CNN and the Pre-trained ResNet-50. A dataset of 75 species, consisting of 9285 training images, 375 testing images, and 375 validation images, is used with a sequence transformation function to increase the number of samples, a part of data augmentation. This approach achieves 95% accuracy and 93% accuracy without data augmentation, respectively. Abdul Fadlil et al. [12] proposed butterfly image identification and classification using multilevel thresholding segmentation and a CNN with an AlexNet architecture, respectively. A dataset of 419 butterfly images with 40 classes is experimented with and without segmentation, and obtained classification accuracy of 83% and 62%, respectively. Based on the result obtained, they conclude that butterfly images without segmentation yield the highest accuracy comparatively. Idir Filali et al. [13] introduced a new segmentation process based on quantum mechanics and implemented it on a multilayered graph structure using the Schrödinger equation in a propagation process across graph layers, which is guided by background and foreground information with the local contrast method with a dataset of 832 ecological images (Leeds butterfly dataset) [1]. Proposed Single-layer Graph Quantum cuts (SGQ) achieve the highest segmentation accuracy compared to other state-of- the-art methods. Hafeez Anwar [14] proposed a large-scale comparison across a broad range of modern CNN and Transformer models, using 64 architectural types on a public butterfly dataset that consists of ecological photos of butterflies. Claims that modern Vision Transformers (ViTs) outperform other deep learning architectures. From the above literature, it has been observed that few attempts have been made to propose the detection, classification and segmentation of butterfly species using computer vision and machine learning approaches, which are of immense value for ecologists and zoologists to identify the butterfly species. However, the study indicates that although some attempts were made to collect the dataset, there are major gaps, such as a lack of properly labelled data and fewer samples. To justify computer vision and machine learning classification techniques, a dataset was collected through fieldwork with a zoologist to identify butterfly species, considering pose variation. The proposed work provides details on the newly created butterfly

dataset with proper labelling across different lighting conditions, and on the classification of butterfly species using different deep learning models, with evaluation of the best architecture for butterfly classification.

The rest of the paper is organised as follows. Section II presents the methodology, including dataset construction, data preprocessing, dataset segregation, and deep learning-based evaluation. Section III presents the experimentation analysis. Section IV presents the discussion, followed by a conclusion and future avenues in this direction.

2. METHODOLOGY

The methodology provides an overview of butterfly species classification, which involves data construction, preprocessing, data segmentation, experimentation with different deep learning models, and evaluation. The overall architecture of the proposed methodology is shown in Fig.1, and a detailed explanation of each step is provided in the following subsections.

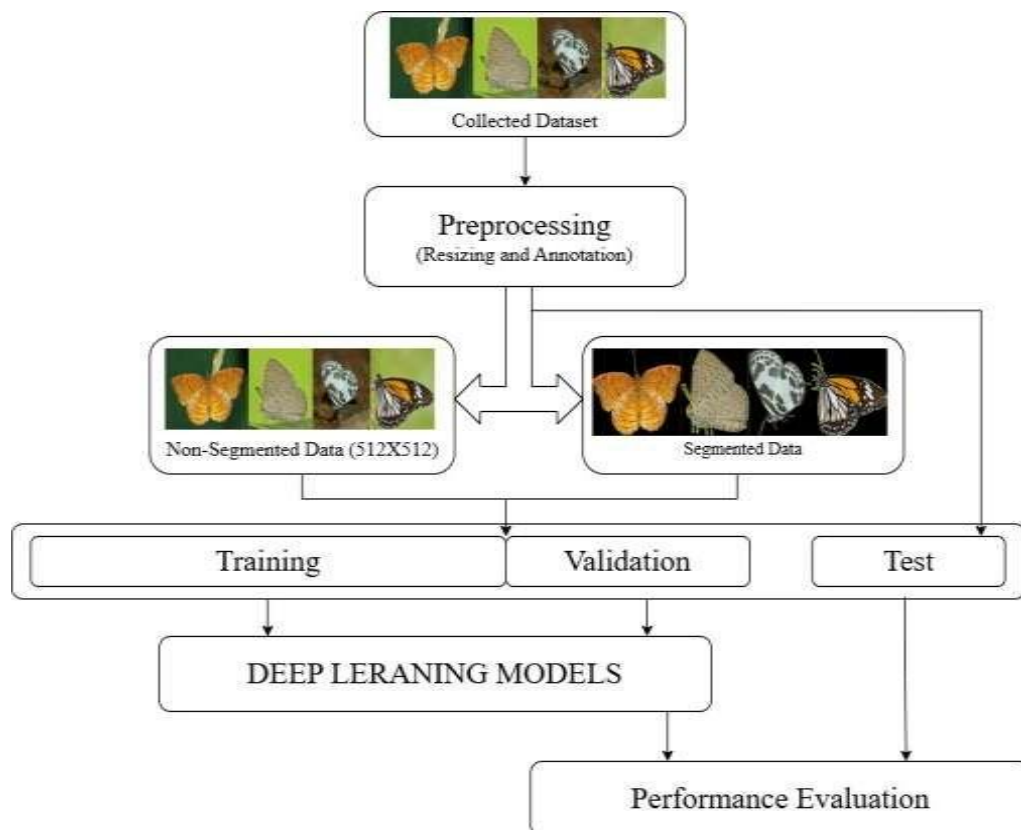


Fig.1 Framework for proposed research work.

2.1. Dataset Preparation

To strengthen the dataset, images and their information are collected through legitimate internet sources such as Butterflies of India [15], Butterfly Circle [16], and iNaturalist [17], further verified with proper labelling from the experts and the dataset was pre-processed. Initially, all images are manually examined to identify and remove samples that may negatively affect feature extraction and model performance. This ensures its overall quality and stability for computer vision tasks. The collected samples, their labels, and statistics are presented in Table 1.

Table 1: Class-wise distribution of butterfly species in the dataset.

Sl.No	Species Name	Count	Sl.No	Species Name	Count
1	Angled Costor	377	52	Dark Grass Blue	336
2	Apefly	158	53	DarkBlueTiger	138
3	Banded Bluepierrot	181	54	DarkGlassyTiger	43
4	Baronet	524	55	Dart Palm Dart	270
5	Black Rajah	324	56	Double Banded Crow	203
6	Black spotted grass jewel	129	57	Elbowed Pierrot	88
7	Black VeinTiger	122	58	Fivebar Swordtail	151
8	Blue Glassy Tiger	132	59	Forget me not	229
9	Blue mormoon	196	60	Gaint Red eye	164
10	Blue Pansy	248	61	Gaudy Baron	334
11	Blue Tiger	416	62	Glad Eye Bush Brown	136
12	Bright Babul Blue	183	63	Glassy Tiger	135
13	Chestnut angled	176	64	Gram blue	294
14	Chocolate Pansy	186	65	Grey Pansy	145
15	Chocolate_Tiger	136	66	GreyGlassyTiger	132
16	CommonJay	275	67	Guava Blue	186
17	CommonMime	273	68	Indian Cupid	112
18	Common_Albatros	164	69	Indian Sunbeam	248
19	Common_Bandedd_Awl	297	70	Jocker	430
20	Common_Banded_Peacock	119	71	Large Oakblue	370
21	Common_Baron	265	72	Lemon Pansy	152
22	Common_Blue_bottle	251	73	Lesser Grass Blue	534
23	Common_Bush_Brown	112	74	Lime Swallowtail	306
24	Common_Bush_hopper	283	75	Long tailed blue	250
25	Common_cerulean	210	76	Long-brand Bushbrown	164
26	Common_Crow	144	77	MalabarBandedPeacock	119
27	Common_Emigrant	195	78	MalabarBandedSwallowtail	50
28	Common_Evening_Brown	421	79	MalabarRaven	50

29	Common_Four_Ring	795	80	MalabarRose	126
30	Common_Grass_Dart	203	81	Monkey Puzzle	471
31	Common_Grass_Yellow	262	82	Mottled Emigrant	480
32	Common_Gull	183	83	Nigger	244
33	Common_Hedge_Blue	263	84	One Spot Grass yellow	116
34	Common_Jezebel	106	85	Orange-spotted Grass Jewel	128
35	Common_Lascar	194	86	Painted Lady	231
36	Common_Leopard	181	87	Pale Grass Blue	623
37	Common_Lime	168	88	ParisPeacock	164
38	Common_Line_Blue	293	89	Peacock_Pansy	258
39	Common_Mormon	364	90	Pea_Blue	545
40	Common_Pierrot	350	91	Plain_Cupid	267
41	Common_Rose	186	92	Plain_Tiger	286
42	Common_Sailar	353	93	Red_Helen	177
43	Common_Three_ring	203	94	Southern_Birdwing	222
44	Common_Tiger	154	95	Spot_Sword_Tail	241
45	Common_Tree_Brown	124	96	Swamp_Tiger	83
46	Common_Yellow_breasted_Flat	107	97	Tailed_Jay	208
47	Crimson Rose	141	98	Tailless_Line_Blue	153
48	Crimson tip	219	99	Tamil_Yemon	126
49	Danaid eggfly	271	100	Towny_Coster	190
50	Dark Brand Bush Brown	119	101	White_Orange_Tip	171
51	Dark cerulean	351	102	Zebra_Blue	181

Table 1 provides the details about 102 butterfly species and the number of samples collected for each species. The number of samples in each class varies significantly, indicating the class imbalance in the dataset. This variation in sample distribution reflects real-world ecological data collection challenges. Uneven distribution of samples may significantly influence the training efficiency of deep learning models and subsequently the classification results. Standard evaluation metric results may mislead class-wise classification. In order to address this issue, in the proposed research, experimentation was carried out with evaluation metrics such as weighted precision, weighted recall, and weighted F1-score, along with the standard accuracy. The detailed evaluation methods are discussed in the following sections. A sample of each class from the dataset is shown in Fig. 2.



Fig.2 Visual overview of the butterfly dataset.

Since the dataset contains images from an ecological background, it can be clearly observed that foliage, soil, flowers, and branches are in the background, as well as real-world challenges, including background clutter, partial occlusion, shadow effects, and scale variation. Additionally, intraclass and interclass variations were observed due to differences in the gender, wing orientation and lightning conditions.

2.2 Preprocessing

Preprocessing is one of the important stages of preparing data for further processing. It includes data cleaning, removal of long shots (pictures taken from a long distance), duplicates, incorrect class labels and partially captured images. This elimination process enhances the overall quality of the dataset and reduces noise and bias. This method ensures that retained images provide sufficient details for computer vision operations. Finally, 23547 images were selected and resized to 512X512. From the selected images, the RoI was manually extracted by polygon clipping annotation using the LabelMe tool. The annotation workflow in LabelMe generates JSON files containing detailed metadata, including polygon coordinates with object class labels, and segmentation boundaries. This helps in the segmentation of butterfly regions with high- quality ground truth masks. Fig. 3 shows the RoI extraction using polygon clipping.



Fig.3 Extraction of RoI using Polygon Clipping.

The butterfly's complex ecological context is defined by a polygonal boundary around the RoI. The clipping algorithm then masks out all the pixels located outside the polygon, thereby isolating only the butterfly region. This method is particularly effective for extracting the region of interest from butterfly images, which focuses on the butterfly's characteristics, such as colour and structure.

2.3 Dataset Segregation

Dataset segregation ensures the systematic partitioning of a dataset into training, validation, and test sets. The dataset is split into training (70%), validation (10%), and testing (20%) sets, preserving class-wise structure and ensuring randomization in training and validation, and kept identical across segmented and non-segmented data, which prevents the training bias. This is helpful for robust model training, hyperparameter tuning, and unbiased performance evaluation of deep learning-based image classification methods.

2.4 Deep Learning Models

Deep Learning models are a subset of machine learning and artificial intelligence models, distinguished by their ability to learn complex patterns through deep neural network architectures. It uses multilayer artificial neural networks inspired by the human brain to efficiently analyse large-scale, complex, unstructured data for computer vision tasks. Deep learning models used for classification can be broadly classified into Classical CNN models, Lightweight CNN models, modern CNN models, Attention-Enhanced CNN models, and Transformer-based models.

2.4.1 Classical CNN models

Classical CNN models are fundamental deep learning architectures developed for image recognition, classification, and computer vision tasks, which use stacked convolutional layers for feature extraction, down-sampling, and classification. Prominent classical CNN architectures are Inception, VGG, Xception, ResNet, and DenseNet. These architectures primarily capture local spatial patterns using fixed receptive fields and progressively aggregate them into higher-level semantic features.

2.4.2 Light-weight CNN model

Lightweight CNN models are designed to minimize computational efficiency, memory and cost, which can be deployed on resource-constrained devices such as mobiles and embedded systems. Some of the Light weight CNN architectures are MobileNet, ShuffleNet, SqueezeNet, and (Mobile Neural Architecture Search) MNASNet. These models use depth-wise separable convolution, which resolves the conventional convolution into depth-wise and pointwise convolution layers, resulting in a significant reduction in the number of parameters and floating-point operations (FLOPs). They provide a strong balance between performance and efficiency, making them ideal for real-time monitoring applications.

2.4.3 Modern CNN

Modern CNN models are highly polished deep learning architectures and designed for automatic and efficient learning of special features. These models are effectively used for classification, detection and segmentation tasks. Modern CNNs effectively capture texture and edge features in images with small convolutional filters, which enhances the classification performance. Some state-of-the-art architectures include ResNeXt,

Advanced-ResNet, EfficientNet, ConvNeXt, and RegNet. These are specially designed for classification, detection and semantic segmentation. Low-level layers detect simple patterns such as edges, lines, and textures, while high-level layers combine the features to detect complex structures.

2.4.4 Attention-Enhanced CNN

Attention-enhanced CNNs are specially restructured neural networks that prioritise certain features or regions and suppress noise for accurate prediction. Attention-enhanced CNNs improve the performance by integrating attention mechanisms such as CBAM, SE blocks, or self-attention. These models are primarily used for deepfake detection by reducing feature redundancy and enhancing the extraction of key information.

2.4.5 Transformer-Based Models

Transformer-based vision models focus on self-attention mechanisms, dispensing with recurrence and convolutions entirely. Attention approximation refers to a set of techniques designed to make a self-attention mechanism. Vision Transformer (ViT) and Swin Transformer are state-of-the-art self-attention mechanisms that capture the features of distant pixels that are semantically related and establish relationships across the entire image. ViT and Swin Transformer are used for computer vision tasks such as image classification, object detection, and segmentation.

Based on the deep learning model categories discussed, the proposed research experimented on some state-of-the-art architectures from each category, which are suitable for the classification of butterfly species.

Initially, Classical CNN architectures such as Xception, Inception variants (V1, V3), VGG variants (16, 19), ResNet variants (18, 34, 50, 101, 152), and DenseNet variants (121, 161, 169, 201), are investigated.

Secondly, from Light-weight CNN models, ShuffleNetV2, SqueezeNet, MobileNet variants (MobileNet, MobileNet-V1, V2, V3-small, V3-large), MNASNet variants (0.5,1.0) are investigated.

Thirdly, from Modern CNN models variants of ResNeXt, variants(50,101) of Advanced-ResNet, EfficientNet variants (B0, B1, B2, B3, B4, B5, B6, B7), variants of ConvNeXt variants (Tiny, Small, Base), and RegNet variants (X-400MF,800MF,1.6GF, Y-400MF, Y-800MF, Y1.6GF).

Fourthly, from Attention-enhanced CNN models: SE-Net variants (SE-ResNet-50, 101, 152, 154).

Finally, from Transformer-based Models ViT-B/16, Deit-Small, and Swin-Tiny.

A total of 54 Deep CNN architectures are evaluated, and the results are discussed in the subsequent section.

3. EXPERIMENTATION AND ANALYSIS

Experimental analysis refers to the detailed evaluation and interpretation of experimental results to understand model performance, compare methodologies, and draw meaningful conclusions. In the proposed research work, a large set of deep architectures is evaluated on both the segmented and non-segmented butterfly datasets. A unified experimental protocol was adopted across all classification architectures; all images were normalised using the ImageNet mean and Standard deviation to ensure a consistent input distribution and stable convergence during training. A transfer learning strategy was employed by initialising the network with pretrained weights, and feature extraction layers were frozen to preserve minute visual representations. The modified classifier consisted of a fully connected layer with 128 neurons, followed by a ReLU

activation function, a dropout layer with a rate of 0.5 to mitigate overfitting, and a final output layer corresponding to the total number of 102 target butterfly classes. Model training was performed using the Adam optimizer with a learning rate of 1×10^{-3} with the combination of a categorical cross-entropy loss function, which is appropriate for multi-class classification problems. Furthermore, an early stopping mechanism was applied by monitoring validation accuracy with a patience of five epochs, and the model parameters yielding the best validation performance were retained for subsequent evaluation on the test dataset.

Model performance is evaluated using accuracy, weighted precision, weighted recall, and weighted F1-score to address class imbalance in the multi-class dataset.

The mathematical formulations of evaluation metrics used in deep architectures are as follows.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy measures the overall model performance by calculating the number of correctly classified samples divided by the total number of samples.

$$Weighted\ Precision = \sum_{i=1}^c \left(\frac{n_i}{N} \times P_i \right)$$

The weighted precision reflects the reliability of the model's positive predictions. It is computed by calculating the precision for each class and then weighting it by that class's sample proportion. Subsequently, classes with higher samples contribute more to the overall precision score, making this metric particularly suitable for a multi-class imbalanced dataset.

$$Weighted\ Recall = \sum_{i=1}^c \left(\frac{n_i}{N} \times R_i \right)$$

The weighted recall evaluates the model's ability to correctly identify samples from each class. It is computed by calculating the recall for each class and weighting by its balanced samples in that class. Subsequently, classes with a larger number of samples contribute more to the overall recall score, making this metric particularly suitable for evaluating performance on imbalanced multi-class datasets.

$$Weighted\ F1\ Score = \sum_{i=1}^c \left(\frac{n_i}{N} \times F1_i \right)$$

The weighted F1-score provides a balanced measure by integrating precision and recall. It is computed by calculating the F1-score for each class and weighting it by the proportion of samples in that class. As a result, classes with a larger number of samples contribute more significantly to the overall F1-score, making this metric well-suited for evaluating performance on imbalanced multi-class datasets.

Unified notation ensures consistency and clarity in evaluating imbalanced multi-class classification performance in weighted precision, weighted recall, and weighted F1-score. Here, C = total number of classes, n_i = number of samples in class i , N is the total number of samples, P_i , R_i , $F1_i$ = precision, recall, and F1-score for class i , respectively.

Table 2 presents a comparative evaluation of Classical CNN models on the Non-Segmented and Segmented butterfly datasets. The experimental analysis demonstrates that classical CNN architectures achieve significantly better performance on the segmented butterfly dataset than on the non-segmented dataset. In particular, ResNet-50 achieved the highest performance on the segmented butterfly dataset with a weighted F1 score 85.33%.

Table 2: Performance Analysis of Classical CNN Deep Learning Architectures on Non-Segmented and Segmented Butterfly Dataset.

Models	Family	Architecture	Dataset	Accuracy	Precision	Recall	F1-Score
Classical CNN models	Inception	Inception-V1	Non-SEG	66.09	64.3	62.09	60.73
			SEG	75.64	77.05	75.64	74.82
		Inception-V3	Non-SEG	54.77	57.72	54.77	52.09
			SEG	65.11	68.26	65.11	62.11
	VGG	VGG-16	Non-SEG	67.16	70.88	67.16	66.3
			SEG	75.61	76.86	75.61	74.57
		VGG-19	Non-SEG	66.16	69.7	62.57	63.99
			SEG	73.14	74.59	73.14	72.18
	Xception	Xception	Non-SEG	67.63	68.35	67.63	65.48
			SEG	70.23	72.57	70.23	68.72
	ResNet	ResNet-18	Non-SEG	65.53	66.47	65.53	64.38
			SEG	79.34	80.95	79.34	78.34
		ResNet-34	Non-SEG	63.38	64	63.38	61.89
			SEG	79.1	79.79	79.1	78.57
		ResNet-50	Non-SEG	73.8	75.11	73.8	73.51
			SEG	85.38	85.98	85.38	85.33
		ResNet-101	Non-SEG	68.81	69.39	68.81	68.08
			SEG	83.91	84.58	83.91	83.87
		ResNet-152	Non-SEG	69.65	70.98	69.65	69.25
			SEG	82.51	83.51	82.51	82.26
	DenseNet	DenseNet-121	Non-SEG	71.02	73.35	71.02	70.56
			SEG	82.49	83.38	82.49	82.08
		DenseNet-161	Non-SEG	75.93	76.91	75.93	75.02
			SEG	83.2	83.94	83.2	82.44
		DenseNet-169	Non-SEG	74.77	76.33	74.77	74.27
			SEG	84.82	85.31	84.82	84.42
		DenseNet-201	Non-SEG	71.99	72.19	71.99	68.81
			SEG	83.49	84.37	83.49	83.15

Table 3 presents a comprehensive performance comparison of modern convolutional neural network architectures. This includes ResNeXt, Advanced-ResNet variants, EfficientNet, ConvNeXt, and RegNet families, evaluated on both Non-Segmented and Segmented butterfly image datasets. The ConvNeXt family achieves the highest performance, with ConvNeXt-Tiny achieving the best results on the SEG dataset, with a weighted F1-score of 90.88%.

Table 3: Performance Analysis of Modern CNN Architectures on Non-Segmented and Segmented Butterfly Dataset.

Model	Family	Architecture	Dataset	Accuracy	Precision	Recall	F1-Score
Modern CNN	ResNeXt	ResNeXt-50	Non-SEG	67.5	69.3	67.5	67.03
			SEG	81.52	81.96	81.52	81.26
		ResNeXt-101	Non-SEG	70.23	71.42	70.23	69.79
			SEG	83.62	84.15	83.62	83.45
	Advanced-ResNet	Wide-ResNet- 50	Non-SEG	74.04	74.77	74.04	73.87
			SEG	85.35	86.28	85.25	85.31
		Res2Net-50	Non-SEG	50.23	53.76	50.23	47.91
			SEG	66.58	68.59	66.58	64.8
	EfficientNet	EfficientNet- B0	Non-SEG	69.97	71.35	69.97	69.64
			SEG	81.86	82.69	81.86	81.65
		EfficientNet- B1	Non-SEG	74.93	75.9	74.93	74.56
			SEG	85.93	86.33	85.93	85.74
		EfficientNet- B2	Non-SEG	69.29	70.91	69.29	68.82
			SEG	83.14	83.47	83.14	82.87
		EfficientNet- B3	Non-SEG	66.79	68.87	66.79	66.28
			SEG	83.35	83.97	83.35	83.23
		EfficientNet- B4	Non-SEG	65.77	68.02	65.77	65.23
			SEG	82.75	83.39	82.75	82.55
		EfficientNet- B5	Non-SEG	67.03	68.66	67.03	66.83
			SEG	79.84	80.48	79.84	79.55
		EfficientNet- B6	Non-SEG	66.48	67.76	66.48	66.14
			SEG	79.84	80.52	79.84	79.47
		EfficientNet- B7	Non-SEG	68.16	69.42	68.16	68.71
			SEG	82.07	82.86	82.07	81.88
	ConvNeXt	ConvNeXt- Tiny	Non-SEG	84.9	85.49	84.9	84.81
			SEG	90.78	91.4	90.78	90.88
		ConvNeXt- Small	Non-SEG	84.75	85.34	84.75	84.57
			SEG	90.28	90.92	90.28	90.29
		ConvNeXt- Base	Non-SEG	85.43	85.81	85.43	85.09
			SEG	90.94	91.55	90.94	90.67
	RegNet	RegNet-X- 400MF	Non-SEG	68.11	70.21	68.11	67.99
			SEG	81.57	82.77	81.57	81.63
		RegNet-X- 800MF	Non-SEG	68.11	70.21	68.11	67.99
			SEG	83.91	84.44	83.91	83.64
		RegNet-X- 1.6GF	Non-SEG	75.11	76.75	75.11	75.12
			SEG	85.3	86.25	85.3	85.31
		RegNet-Y- 400MF	Non-SEG	67.63	69.6	67.63	67.52
			SEG	81.94	82.86	81.94	81.86
		RegNet-Y- 800MF	Non-SEG	73.41	75.48	73.41	73.6
			SEG	84.4	84.81	84.4	84.27
		RegNet-Y- 1.6GF	Non-SEG	73.09	74.57	73.09	73.24
			SEG	84.43	85.37	84.43	84.67

Table 4 presents the performance comparison of lightweight CNN deep architectures, including MobileNet variants, ShuffleNet, SqueezeNet, and MNASNet, evaluated on both segmented and non-segmented butterfly datasets. Across the evaluated architectures, results on the segmented dataset improved compared to those on the non-segmented dataset. The experimental results for MobileNet-V3 Large showed a significant improvement on the segmented dataset with a 13.89% weighted F1-score compared to the non-segmented dataset.

Table 4: Performance Analysis of Lightweight Deep Learning Architectures on Non-Segmented and Segmented Butterfly Dataset.

Model	Family	Architecture	Dataset	Accuracy	Precision	Recall	F1-Score
Lightweight CNN models	MobileNet	MobileNet	Non-SEG	72.07	72.95	72.07	71.71
			SEG	79.6	81.38	76.93	79.07
		MobileNet-V1	Non-SEG	74.9	76.11	74.9	74.54
			SEG	85.03	86.11	85.03	84.9
		MobileNet-V2	Non-SEG	75.09	77.12	75.09	75.2
			SEG	84.25	85.25	84.25	84.04
		MobileNet-V3 Small	Non-SEG	67.66	68.54	67.66	67.45
			SEG	81.52	81.87	81.52	81.3
		MobileNet-V3 Large	Non-SEG	71.25	71.78	71.25	71.01
			SEG	85.03	85.55	85.03	84.9
	ShuffleNet	ShuffleNetV2	Non-SEG	70.83	71.8	70.83	70.22
			SEG	83.01	83.74	83.01	82.62
	SqueezeNet	SqueezeNet	Non-SEG	74.43	76.15	74.43	73.8
			SEG	84.56	86.27	84.56	84.43
	MNASNet	MNASNet-0.5	Non-SEG	71.75	72.95	71.95	71.56
			SEG	83.38	83.88	83.38	83.23
		MNASNet-1.0	Non-SEG	74.59	76.41	74.59	74.79
			SEG	84.77	85.67	84.77	84.8
		MNASNet-1.3	Non-SEG	74.01	75.12	74.01	73.87
			SEG	82.36	83.25	82.36	82.37

Table 5 presents the performance evaluation of attention-enhanced convolutional neural network architectures based on the Squeeze-and-Excitation (SE) mechanism, including SE- ResNet50, SE-ResNet101, SE-ResNet152, and SENet154, assessed on both Non-Segmented and Segmented datasets. The models are evaluated to analyse the impact of channel-wise attention on classification performance. A clear performance improvement is observed in all models when segmentation is applied. The SE-ResNet50 architecture consistently outperforms deeper SE-Net variants, achieving the highest accuracy, precision, recall, and F1 score on both the Non-Segmented and Segmented datasets. In contrast, deeper SE-based models such as SE-ResNet101, SE-ResNet152, and SENet154 exhibit notably lower performance on the dataset.

Table 5: Performance Analysis of Attention-enhanced CNNs (SE-Net Variants) on Non-Segmented and Segmented Butterfly Dataset.

Model	Family	Architecture	Dataset	Accuracy	Precision	Recall	F1-Score
Attention-enhanced CNNs	SE-Net	SE-ResNet50	Non-SEG	75.32	77.19	75.32	75.4
			SEG	85.93	86.56	85.93	85.81
		SE-ResNet101	Non-SEG	18.34	25.78	18.34	16.92
			SEG	34.17	42.82	34.17	31.67
		SE-ResNet152	Non-SEG	19.44	25.41	19.44	18
			SEG	36.19	44.38	36.19	34.63
		SENet154	Non-SEG	24.72	36.95	24.72	23.36
			SEG	44.19	50.35	44.19	41.78

Table 6 presents the comparative performance of transformer-based architectures, namely Vision Transformer (ViT-B/16), DeiT-Small, and Swin-Tiny, evaluated on butterfly species classification under both segmented and non-segmented datasets. The results demonstrate consistent and significant improvement on the segmented dataset. Swin-Tiny achieves the highest overall performance, with an F1 Score of 90.47%.

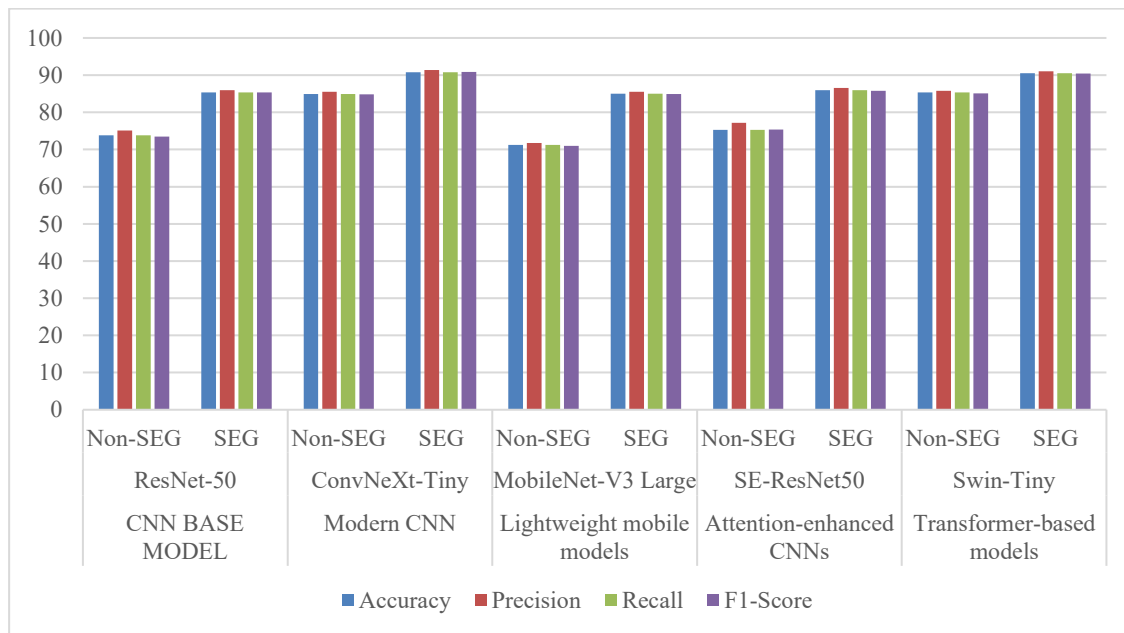
Table 6: Performance Analysis of Transformer-based models on Non- Segmented and Segmented Butterfly Dataset.

Model	Family	Architecture	Dataset	Accuracy	Precision	Recall	F1-Score
Transformer-based models	Transformer	ViT-B/16	Non-SEG	82.33	82.69	82.33	82.18
			SEG	90	90.2	90	89.92
		DeiT-Small	Non-SEG	78.66	79.13	78.66	78.53
			SEG	87.87	88.07	87.87	87.57
		Swin-Tiny	Non-SEG	85.32	85.8	85.32	85.1
			SEG	90.49	91.05	90.49	90.47

Table 7 demonstrates a comparative evaluation of representative models from five major categories: CNN-based models, modern CNNs, lightweight CNN models, attention-enhanced CNNs, and transformer-based models on both Segmented and Non-segment butterfly datasets. It has been clearly observed that background separation is important, as it enhances classification results across all deep learning models. Based on the observations, it has been confirmed that ConvNext-Tiny and Swin-Tiny, from Modern CNN and Transformer-based models, respectively, outperform on all evaluation metrics in both segmented and non-segmented datasets. Fig 4 shows the graphical visualization

Table 7: Comparison of the best-performing deep learning models on the Segmented and Segmented Butterfly Dataset.

Model	Architecture	Dataset	Accuracy	Precision	Recall	F1-Score
CNN BASE MODEL	ResNet-50	Non-SEG	73.8	75.11	73.8	73.51
		SEG	85.38	85.98	85.38	85.33
Modern CNN	ConvNeXt-Tiny	Non-SEG	84.9	85.49	84.9	84.81
		SEG	90.78	91.4	90.78	90.88
Lightweight CNN Model	MobileNet-V3 Large	Non-SEG	71.25	71.78	71.25	71.01
		SEG	85.03	85.55	85.03	84.9
Attention-enhanced CNNs	SE-ResNet50	Non-SEG	75.32	77.19	75.32	75.4
		SEG	85.93	86.56	85.93	85.81
Transformer-based models	Swin-Tiny	Non-SEG	85.32	85.8	85.32	85.1
		SEG	90.49	91.05	90.49	90.47

**Fig.4 Graphical visualisation of best-performing deep learning models on the non-segmented and segmented butterfly dataset.**

The first and third columns are the input for the analysis of segmented and non-segmented images for classification, respectively. The second and fourth column shows the classification output. This analysis clearly shows that a non-segmented input image with unwanted background information leads to misclassification, because of similar colour, texture, background and size. After applying segmentation, background noise is completely removed, which helps the deep architecture to extract more detailed information from the RoI. Which leads to consistent improvement in the classification result.

Input non-segmented Image	Classification Result	Input segmented Image	Classification Result
	Tamil_Yemon		Angled Costor
	Angled Costor		Angled Costor
	Common Pierrot		Banded Bluepierrot
	Banded Bluepierrot		Banded Bluepierrot
	Gaudy Baron		Black Raja
	Black Raja		Black Raja
	Dark Brand Bush Brown		Common Bush Brown
	Common Bush Brown		Common Bush Brown
	Gram blue		Bright Babul Blue
	Bright Babul Blue		Bright Babul Blue

Fig.5 Some examples of classification improvement analysis after the segmentation on the ConvNeXt-Tiny architecture.

Fig. 5 and 6 show examples of actual images and randomly selected predicted images from the non-segmented and segmented datasets using ConvNeXt-Tiny. These misclassifications primarily occur in non-segmented data due to illumination, pose,

background, visual similarity, wing colouration, and spotting patterns. Even though there is an improvement in the F1 score on the segmented butterfly classification, it has also been observed that the architecture exhibits misclassification in segmented data.

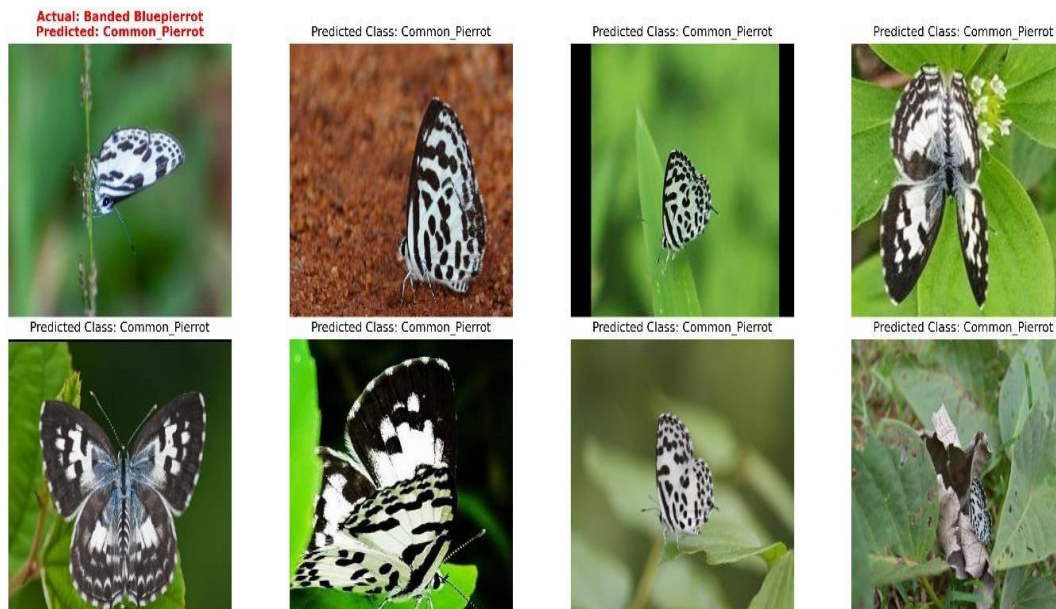


Fig.5 Comparison of a misclassified sample image and a randomly selected correctly classified image from the non-segmented dataset.

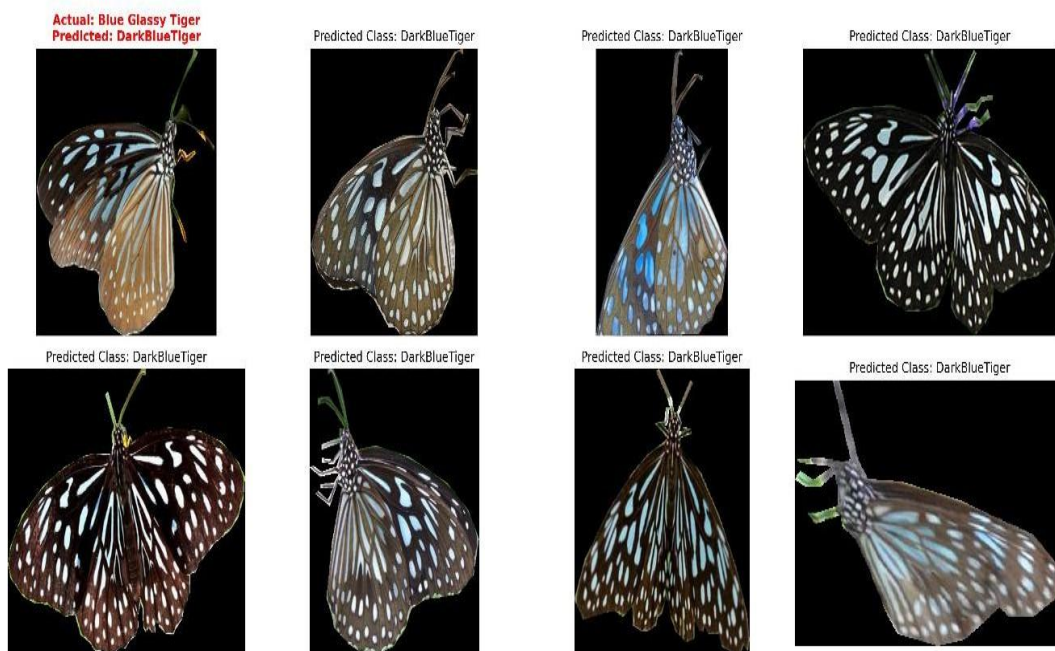


Fig.6 Comparison of a misclassified sample image and a randomly selected correctly classified image from the segmented dataset.

4. DISCUSSION AND CONCLUSION

The proposed research investigated the effectiveness of deep learning models for butterfly species classification. The experiments with deep architectures clearly indicate that segmentation significantly improves classification performance compared to non-segmented images. Overall, the results validate that segmentation is a vital preprocessing step for improving butterfly classification performance, particularly in natural, unconstrained environments.

In this paper, based on an extensive experimental study of deep learning models, ConvNeXt-Tiny and Swin-Tiny architectures outperformed all other architectures on the segmented and non-segmented butterfly image dataset, confirming that background removal is an effective preprocessing step for enhancing the classification accuracy. Despite this improvement, deep architectures exhibit misclassifications, especially among the closely related species. This misclassification is due to high inter-class visual similarity. These findings highlight the characteristics of fine-grained butterfly classification and the limitations of lightweight architectures in capturing highly discriminative features.

Future work will focus on integrating handcrafted features with deep features to improve classification performance, with a particular emphasis on visually similar colouration, pose variations and mimicking species.

References

1. Josiah Wang and Katja Markert and Mark Everingham, Learning Models for Object Recognition from Natural Language processing Description, "Proceedings of the British Machine Vision Conference" year ="2009".
2. Seung-Ho Kang, Su-Hee Song, Sang-Hee Lee, Identification of butterfly species with a single neural network system, *Journal of Asia-Pacific Entomology* 15 (2012) 431–435.
3. Nur Nabila Kamaron Arzar, Nurbaity Sabri, Nur Farahin Mohd Johari, Anis Amilah Shari, MohdRahmat Mohd Noordin, Shafaf Ibrahim, Butterfly Species Identification Using Convolutional Neural Network (CNN), 2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS 2019), 29 June 2019, Selangor, Malaysia.
4. Ruoyan Zhao, Cuixia Li¹, Shuai Ye and Xinru Fang, Butterfly Recognition Based on Faster R-CNN, *IOP Conf. Series: Journal of Physics: Conf. Series* 1176 (2019) 032048, doi:10.1088/1742-6596/1176/3/032048.
5. H.Theivaprakasham, Identification of Indian butterflies using Deep Convolutional Neural Network, *Journal of Asia-Pacific Entomology* (2020),doi: <https://doi.org/10.1016/j.aspen.2020.11.015>.
6. Aijiao tan, guoxiong zhou, and mingfang he, Rapid Fine-Grained Classification of Butterflies Based on FCM-KM and Mask R-CNN Fusion, 20, 2020.Digital Object Identifier 10.1109/ACCESS.2020.3007745.
7. Ayad Saad Almryad, Hakan Kutucu, Automatic identification for field butterflies by convolutional neural networks, *Engineering Science and Technology, an International Journal* 23 (2020) 189–195.
8. Fathimathul Rajeena P. P., Rasha Orban, Kogilavani Shanmuga Vadivel, Malliga Subramanian, Suresh Muthusamy, Diaa Salam Abd Elminaam, Ayman Nabil, Laith Abulaigh, Mohsen Ahmadi and Mona A. S. Ali. A Novel Method for the Classification of Butterfly Species Using Pre-Trained CNN Models, *Electronics* 2022, 11, 2016. <https://doi.org/10.3390/electronics11132016>.
9. Halit Çetiner, A Novel Method for Classification of Butterfly Species Using CNN, 2nd International Conference on Scientific and Academic Research, 2023, Published by All Sciences Proceedings.

10. Rumana Yasmin, Aparna Das, Liton Jude Rozario, Md. Ezharul Islam, Butterfly detection and classification techniques: A review, *Intelligent Systems with Applications* 18 (2023) 200214.
11. Diniati Ruaika, Shofwatul 'Uyun, Optimisation of Residual Network Using Data Augmentation and Ensemble Deep Learning for Butterfly Image Classification, *IJID (International Journal on Informatics for Development)*, e-ISSN: 2549-7448 Vol. 12, No. 2, December 2023, Pp. 350-361.
12. Abdul Fadlil, Ainin Maftukhah, Sunardi and Tole Sutikno, Butterfly Image Identification Using Multilevel Thresholding, Segmentasi and Convolution Neural Network Classification with Alexnet Architecture, *International Journal of Computing and Digital Systems* ISSN (2210-142X), [dx.doi.org/10.12785/ijcds/1501125](https://doi.org/10.12785/ijcds/1501125).
13. Idir Filali, Mohamed Ramdani, Brahim Achour, Butterfly segmentation by multi scaled quantum cuts in agro-ecological environment, *An International Journal, A publication of the European Association for Signal Processing (EURASIP)*, doi.org/10.1016/j.sigpro.2024.109420.
14. Anwar, H. Butterflies and benchmarks: a systematic evaluation of deep learning backbones to identify the best model for image-based butterflies species recognition. *Pattern Anal Applic* 29, 13 (2026). <https://doi.org/10.1007/s10044-025-01587-7>
15. Kunte, K. 2026. ABOUT US. In Kunte, K., S. Sondhi, and P. Roy (Chief Editors). *Butterflies of India*, v. 4.12. Indian Foundation for Butterflies.
16. ButterflyCircle.(n.d.). Butterfly images. <https://www.butterflycircle.com/iNaturalist>. (n.d.) Butterflies of Singapore. iNaturalist.