A Convolutional Neural Network for Binary Classification of Monarch Butterflies

Ajay Goverdhan[1]

Received November 5, 2025
Accepted May 28, 2026
Electronic access July 15, 2026

Pollinators, especially monarch butterflies, are facing severe population declines that threaten biodiversity. This emphasizes the need for effective conservation strategies. However, these butterflies are visually similar to other Lepidoptera species, making field-based identification difficult. This study investigates an automated approach to identifying monarch butterflies, using deep learning. A custom convolutional neural network (CNN) consisting of two convolutional layers followed by a dense layer was implemented using Keras sequential API to perform binary image classification. The dataset was obtained from publicly available sources (Kaggle and images.cv) and comprised 5840 training images and 1461 test images. Data preprocessing was performed to optimize performance. The final model achieved an accuracy of 83.5%, with a precision of 82.5%, a recall of 82.5%, and an F1 score of 0.82. The results demonstrate that the CNN model can effectively identify monarch butterflies, highlighting its potential as a practical tool for supporting conservation efforts and as a cost-effective alternative to more complex architectures.

Keywords: Monarch butterflies, Image classification, deep learning, CNN, species identification, conservation

Introduction

Biodiversity is inalienably linked to our own survival. Pollinators such as monarch butterflies are considered keystone species—organisms that have a disproportionately large effect on their environment relative to their abundance[1]. Yet they are facing catastrophic population declines[2]. This underlines the very urgent need to ramp up conservation efforts. Accurate identification of monarch butterflies is essential for conservation monitoring. However, visual similarities between monarchs and other butterfly species make manual identification challenging, especially for non-experts[3].

Recent advances in artificial intelligence have enabled automated image classification systems. Initially, traditional machine learning models, such as K-nearest neighbors (k-NN) and support vector machines (SVMs), were used. These utilize manually engineered visual features. Kaya et al. have presented a model to identify various species of butterflies based on textural features extracted using the Gray Level Co-occurrence Matrix (GLCM) and Local Binary Pattern (LBP) and classified the images using Extreme Learning Machine (ELM), achieving an accuracy of 98.2%[4]. Kartika et al. extracted LBP features from the butterfly images and used the region props algorithm with SVM to classify the images, obtaining nearly 66% accuracy, which is low[5]. Whereas Xue et al. achieved 98% accuracy using GLCM features and a weighted k-NN classifier[6]. However, it must be noted that one disadvantage of traditional machine learning models is that model performance depends upon human-selected features, identification of which is often time-consuming and requires domain expertise[7].

Over the last few years, deep learning techniques have been used for butterfly classification. It is a technique that enables computer systems to improve with experience and data[7]. They use Artificial Neural Networks (ANN) that can automatically extract features and classify them into given categories based on input data[8]. Singh et al.[9] used a CNN-based model, which was designed to automatically extract and analyze intricate features within butterfly images, demonstrating a high accuracy of 99%. Rodrigues et al. developed a five-layer CNN to classify ten butterfly species using a dataset of 832 images and reported an accuracy of 90%[10]. Halit et al.[11] proposed a 16-layer CNN using data augmentation techniques such as rotation, zooming, and flipping, and reported a validation accuracy of 93.41%.

Large-scale datasets and transfer learning models have further advanced classification performance. Zhu et al.[12] proposed a cascade architecture using pretrained AlexNet for feature extraction and SVM for classification of Lepidoptera species from 1301 images and achieved an accuracy of 100%. Chang et al.[13] compared several CNN architectures for butterfly and moth classification and found that ResNet 18 performed best, achieving a top-5 accuracy of 92.6% on a dataset

[1] *Hopkinton High School, Massachusetts, USA*

of 14,270 images. Zhao et al.[14] and Almryad and Kutucu[15] reported lower accuracy for deep learning-based models due to their use of smaller datasets and not using preprocessed images, respectively. In 2022, Chen and his team[3] created MonarchNet, the first comprehensive dataset consisting of butterfly imagery for monarchs and five visually similar species. His team trained a baseline deep-learning classification model to serve as a tool for differentiating monarch butterflies. Sar et al.[16] created a parrot model that underwent fine-tuning, which adjusts the already-trained ResNet 18 model for the butterfly classification task. By taking advantage of pre-trained networks on the big dataset of butterfly pictures, they made use of the extracted features to reach an optimal performance level for the specific task. They found their model to be more accurate than any currently existing approach. Ruiaka[17] optimized the accuracy of ResNet50 using an augmentation approach and ensemble deep learning for butterfly image classification and achieved an accuracy of 95%. Theivaprakasham et al.[18] reported the highest classification accuracy using ResNet models, particularly ResNet-152, which achieved an accuracy of 94.44%. They used Squeezenet-1.1, which is a lightweight model, to develop a field-ready "Android Application" for real-time butterfly identification.

While these studies report higher accuracy, they use deeper architectures such as ResNet or transfer learning techniques, which require large amounts of input data for training. These require computational resources and fine-tuning expertise that may not be available for small-scale conservative initiatives[7]. Despite strong performance across studies, most work focuses on large multi-species datasets or complex architectures. Fewer studies evaluate simpler custom CNN models specifically designed to distinguish monarch butterflies from visually similar species. A systematic review by Yasmin et al.[8] highlights that it would be beneficial to create a CNN that learns with one species, and the model would not be trained with other malformations that are not of relevance. Simple architecture may also make it small, fast, and suitable for mobile devices with limited memory and processing power, thereby expanding its potential for use by citizen scientists and volunteers[18].

Therefore, this study aims to evaluate the effectiveness of a custom CNN for binary classification of monarchs and non-monarchs. The objectives of this study are:

- Develop a custom convolutional neural network for binary classification of monarch and non-monarch butterflies.
- Evaluate model performance using standard metrics.

To address this objective, a supervised learning approach was used to design and train a custom CNN on a curated dataset of butterfly images. Model performance was evaluated using standard metrics, including accuracy, precision, recall, and F1-score, to assess its effectiveness relative to findings reported in the literature.

Dataset/Methods

This project utilizes datasets from two primary sources to develop a binary classification model for monarch butterfly identification. The first source was the Butterfly Image classification dataset from Kaggle, consisting of approximately 6499 labeled images (224×224 pixels) spanning roughly 75 butterfly species. This dataset included 90 images of monarch butterflies. The second source was a monarch-labeled dataset from images.cv from which 802 monarch images were selected. The images were consolidated into a new project directory named [Author name] ButterflyProject. All images were standardized to a resolution of 224×224 pixels to ensure consistency across datasets before model training. After merging, the dataset contained 894 monarch images and 7,301 images overall. This consolidation enabled the creation of a larger monarch class while preserving species diversity within the non-monarch category. Despite these steps, the combined dataset exhibited class imbalance, with monarch images comprising a smaller proportion of the overall dataset compared to the aggregated non-monarch classes. This imbalance was addressed during model development through stratified data splitting and under-sampling of the majority class in the training dataset (described in the Forming the test and train dataset section). Potential dataset biases may arise from differences in image sources, including variations in lighting conditions and backgrounds. These factors could impact the model's generalizability when applied to images collected in different environments. Future work could address some of these limitations.

Pre-processing

Before model training, the dataset was preprocessed to convert the raw data into a structured format that can be effectively used by machine learning models. Before training, images were loaded from folders using Python. Only JPG, JPEG and PNG images were included. All images were resized to 224×224 pixels. Images were converted from BGR to RGB to maintain correct color representation. Pixel values were normalized to a range of 0–1 by dividing by 255 to help improve training speed. While CV models typically use grayscale for data reduction, we chose to retain color, as it is a significant factor in the butterfly phenotype that helps determine species.

Forming the Train and Test Dataset

All butterfly pictures were combined into a unified project directory called [Author name]ButterflyProject. This curated

dataset included 894 monarch butterflies along with other non-monarch classes. The butterfly dataset was organized in a project folder titled [Author name]ButterflyProject, which contained images grouped into class-specific subfolders (e.g., MONARCH, VICEROY, etc.) along with a `labels.csv` file linking each image to its class. The code read this CSV, gathered all image file paths, and loaded the pictures into memory. Then we used a binary label—if "monarch" is found in the class name, the label is set to 1. Otherwise, the label is set to 0 (non-monarch). The dataset was randomly split into training and testing sets using the `train_test_split` function from scikit-learn, with 80% of the data used for training and 20% for testing. The stratify option ensures that both sets are balanced in terms of monarch and non-monarch butterflies. Under-sampling was used to balance the training dataset by keeping all monarch images but randomly selecting a smaller (and almost equal) number of non-monarchs. This prevents the model from being biased toward the majority class and helps it learn to recognize monarch butterflies more accurately.

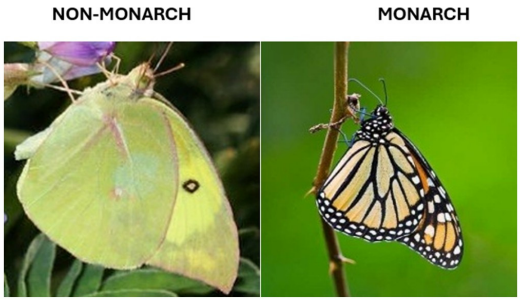

Fig. 1 Example images from the dataset. Left: Non-Monarch. Right: Monarch.

Methodology/Models

CNNs are good at handling image data because they can automatically learn patterns and features such as edges, shapes, and textures, by using specialized layers called convolutional layers.

In this study, a CNN-based butterfly classification model is proposed. The model consisted of a custom CNN implemented using TensorFlow and Keras in Python, comprising two convolutional layers for feature extraction followed by a dense layer for binary classification.

The image training process on butterfly classifications is an important step that helps to enhance the model's performance and make it better at classifying the type of butterfly (in this case, monarch).

The proposed model consists of multiple layers designed to extract image features and perform binary classification. In the first layer, the image is input in order to have three color channels and a size of 224×224 pixels. This layer is called the input layer. The first convolutional layer consists of 32 filters with a 3×3 kernel size and ReLU activation to detect basic visual features such as edges and patterns. The second convolutional layer consists of 64 filters and a 3×3 kernel size was applied using the ReLU activation function to extract more complex features.

Each convolutional layer is followed by a max pooling layer with a 2×2 window size to make the image smaller while keeping important features. The Flatten layer converted the features into a single list for the next layer. This was followed by a dense (fully connected) layer with 64 neurons and a ReLU activation function was added to learn higher-level feature combinations.

After this, a final output layer with one neuron and a sigmoid activation function was used to produce a probability of whether the image is a monarch or a non-monarch.

Training was performed using the RMSprop optimizer with a learning rate of 0.0001 and the binary cross-entropy loss function. This model was trained for 20 epochs with a batch size of 64. Model performance was monitored using validation data during training.

Data Augmentation (Attempted)

Initial experiments explored data augmentation techniques to address the limited number of monarch images ($n = 90$) available in the Kaggle dataset. Augmentation methods such as rotation, flipping and scaling were applied to increase the effective training sample size. However, these experiments resulted in severe overfitting, with the model achieving near-perfect training accuracy but failing to generalize to unseen data. The outcome is likely due to an extremely small number of monarchs in the training dataset. Augmentation proved to be insufficient to improve model performance. To address this limitation, an additional monarch image dataset ($n = 802$) was incorporated from images.cv, thus increasing class representation.

Results

To evaluate the performance of the CNN model, several standard classification metrics were used, including accuracy, precision, recall, F1 score, and binary cross-entropy loss [19].

The model achieved 88.5% training accuracy and 83.5% test accuracy (Table 1). Accuracy is a performance metric that measures how often a model's predictions match the true labels. The small difference between training and test accuracy

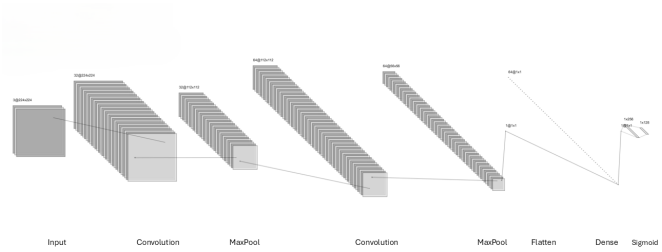

Fig. 2 Illustration of the proposed Convolutional Neural Network. The Flatten layer converts the $56 \times 56 \times 64$ feature maps into a one-dimensional vector of length 200,704. For visualization purposes, the schematic displays a reduced placeholder size.

suggests only mild overfitting and indicates that the model generalizes well to unseen butterfly images. These results suggest that the model captures most features of monarchs vs. non-monarchs.

Precision indicates the degree of confidence that a butterfly classified as a monarch by the model is, in fact, a true monarch. High precision is very important in conservation tasks. For example, if non-monarchs are mislabeled as monarchs, estimates of monarch population could be inflated. This may reduce the urgency of conservation interventions, resources allocated, and public interest. Hence, high precision is of utmost importance. The model achieved 90% precision on the training dataset and 82.5% precision in the test dataset.

Recall (Sensitivity) measures the proportion of actual butterflies correctly identified by the model. The recall score was 85.5% on the training dataset and 82.5% on the test dataset, indicating that the model successfully detects most monarch butterflies but may miss some.

The F1 score, which balances precision and recall, reached 0.82 on the test dataset, suggesting a well-balanced model that performs well both at avoiding false positives and detecting true monarchs.

Model performance was further evaluated using the binary cross-entropy loss function, which measures the difference between the model's predictions and the correct labels. A perfect model would have a loss approaching zero. A random or guessing model would have a loss of 0.69. Our model achieved a loss value of 0.3, indicating that the predictions are reasonably close to the true labels and the model is learning meaningful visual features such as wing patterns, coloration and shape.

This model achieved higher accuracy than some classical machine learning model studies[5]. Classical machine learning models rely on manually engineered features and can be resource-intensive. The model achieved a test accuracy of 83.5%, which is lower than prior studies[3,16,17] that report accuracies above 90%. However, these studies often utilize deeper architectures and larger datasets, which may not be feasible in resource-constrained conservation settings.

Overall, these evaluation metrics demonstrate that the proposed CNN model performs reliably in distinguishing monarch butterflies from visually similar species.

Table 1 Model performance metrics for monarch classification.

Metric	Training Dataset	Testing Dataset
Accuracy	88%	83.5%
Precision	90%	82.5%
Recall	85.5%	82.5%
F1-Score	87%	82%
Loss	0.30	0.38

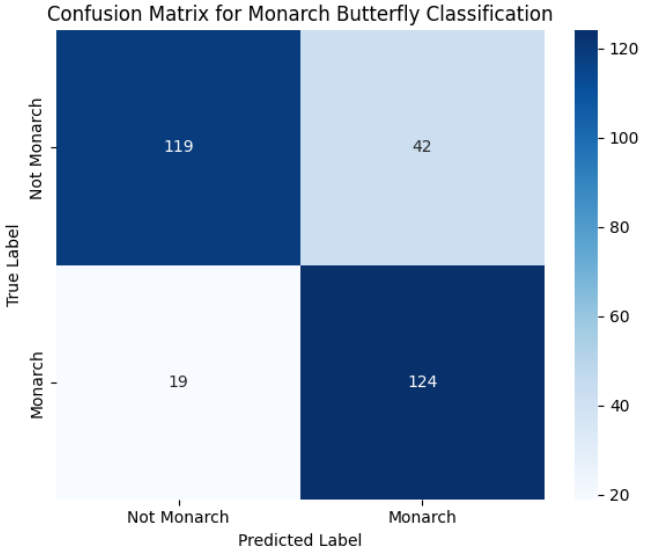

Fig. 3 Confusion Matrix of Monarch Butterfly Model.

The confusion matrix entries are interpreted as follows:

- **True Negatives (TN = 136):** 136 non-monarchs correctly identified as non-monarchs.
- **False Positives (FP = 25):** 25 non-monarchs were incorrectly predicted as monarchs.
- **False Negatives (FN = 25):** 25 monarchs were incorrectly predicted as non-monarchs.
- **True Positives (TP = 118):** 118 monarchs were correctly identified.

Discussion

Pollinators, such as monarch butterflies, are essential for maintaining ecological balance and serve as keystone species for our ecosystem. However, conservation projects today operate under financial restraints and face several resource limitations. A review of prior research shows that deep learning models could be used to classify monarchs—some focused on monarchs and several similar species[3], while others used pre-trained models[4] and complex fine-tuning to achieve this. This study investigated whether a simpler custom CNN could achieve effective performance for targeted monarch butterfly classification.

The results of this study demonstrate that relatively simple CNNs can be effectively applied to butterfly image classification, specifically for distinguishing monarch butterflies from visually similar species. The trained model achieved strong performance across multiple evaluation metrics, indicating that it successfully learned to distinguish visual features. Overall, the model achieved an accuracy of 83.5% with high precision and recall, meeting the study's primary objectives.

Prior work[16–18] has reported high accuracies (often exceeding 90%) using deep pre-trained models such as ResNet. These results are not directly comparable due to the differences in model architecture, dataset size, and evaluation approaches. These higher accuracies are likely attributable to the use of deeper architectures, larger datasets, and transfer learning approaches. However, these approaches often involve greater computational costs and require fine-tuning expertise. In contrast, this work demonstrates that simple custom CNN models can serve as a practical and cost-effective alternative to more complex architectures. This is especially important for smaller research groups or conservation organizations that may not have such resources. Furthermore, similar methods could be adapted to other species facing ecological threats. This study contributes to the growing body of research that supports the use of artificial intelligence for biodiversity monitoring and conservation.

Despite promising results, several limitations should be acknowledged. The model achieved 83.5% accuracy, but certain classification errors still occurred (25 false positives). Visual evaluation of images in Figure 4 provides insight into potential sources of misclassification by the model. The presence of highly saturated floral elements, stems, and other objects in the background may have interfered with feature extraction. This indicates opportunities for further improvement. Another limitation is that most images show a single butterfly in an up-close pose, which may differ significantly from field conditions. Also, interpretability techniques such as Gradient-weighted Class Activation Mapping (Grad-CAM) were not implemented in this study due to time limitations. This would provide deeper insight into misclassified samples. Additionally, the initial dataset exhibited class imbalance between monarch and non-monarch images, which required sampling adjustments that may have influenced model performance. From an ethical standpoint, it is important to consider data equity (AI remains accessible only to a certain demographic), avoid overreliance on automated predictions, and always include human oversight.

Future work could improve model performance by incorporating larger training datasets, particularly high-quality, high-resolution images. Conducting cross-validation would improve the robustness of performance evaluation. Additional improvements could include incorporating Grad-CAM to visualize the regions of images the model relies on during prediction. This would help improve the understanding of the model's decision-making. Comparing the custom CNN with lightweight pretrained architectures such as MobileNet would provide more insight into performance. The model could also be evaluated on independent, field-like datasets that include images captured under realistic conditions, such as those taken by citizen scientists with smartphones. Prior research has demonstrated the feasibility of deploying butterfly classification models in real-world applications. For example, Theivaprakasham et al.[18] developed a field-ready Android application using a lightweight SqueezeNet-1.1 model. Given the growing interest in such tools and widespread adoption of mobile applications, the integration of the model presented in the study into a mobile-based platform may represent an area of future exploration, particularly as larger and more diverse datasets become available to support robust classification. AI-based classification systems could, in this way, support citizen science initiatives, enabling rapid preliminary identification of butterfly species with subsequent verification by human oversight.

As datasets and model architecture continue to improve, artificial intelligence may play an increasingly important role in ecological monitoring and conservation efforts. Simple custom models could be explored first and may be sufficient for targeted classification tasks when trained on carefully curated data, offering a practical approach for conservation applications.

Acknowledgments

I want to thank my mentor, Mr. Henry Cerbone, PhD Candidate at the University of Oxford, MS in Computer Science, Harvard University, for his guidance in this project. I would like to extend my heartfelt appreciation to Mx. Victoria Lloyd, PhD Candidate at the CN Yang Institute for Theoretical Physics at Stony Brook University, for their unwavering guidance, patience, and support, without which this project would not have been possible. I would also like to thank the entire

Fig. 4 Example images of Misclassified Monarchs.

Inspirit AI team for their support and guidance throughout this project.

References

1 California Department of Fish and Wildlife, *Science: Pollinators*, 2020, https://wildlife.ca.gov/Science-Institute/Pollinators, Accessed on 4/22/2025.

2 C. B. Edwards, E. F. Zipkin, E. H. Henry, L. Ries, A. M. Shapiro, A. J. Swengel, S. R. Swengel, D. J. Taron, B. Van Deynze, J. Wiedmann, W. E. Thogmartin, C. B. Schultz *et al.*, *Rapid declines in butterflies across the United States during the 21st century*, 2025, 10.1126/science.adp4671.

3 T. Y. Chen, *MonarchNet: Differentiating Monarch Butterflies from Butterflies Species with Similar Phenotypes*, 2022, 10.48550/ARXIV.2201.10526.

4 Y. Kaya, L. Kayci, R. Tekin and Ö. F. Ertuğrul, *Evaluation of texture features for automatic detecting butterfly species using extreme learning machine*, 2014, 10.1080/0952813X.2013.861875.

5 D. S. Y. Kartika, D. Herumurti and A. Yuniarti, *Local binary pattern method and feature shape extraction for detecting butterfly image*, 2018, 10.21660/2018.50.IJCST21.

6 A. Xue, F. Li and Y. Xiong, *Automatic identification of butterfly species based on gray-level co-occurrence matrix features of image block*, 2019, 10.1007/s12204-018-2013-y.

7 I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, 2016, https://www.deeplearningbook.org/.

8 R. Yasmin, A. Das, L. J. Rozario and M. E. Islam, *Butterfly detection and classification techniques: A review*, 2023, 10.1016/j.iswa.2023.200214.

9 A. K. Pandey, S. Singh, A. S. Singh, A. K. Mishra and B. Yadav, *Butterfly species recognition using convolutional neural network*, 2023.

10 R. Rodrigues, R. Manjesh, P. Sindhura, S. N. Hegde and A. Sheethal, *Butterfly species identification using convolutional neural network (CNN)*, 2019.

11 H. Çetiner, *A Novel Method for Classification of Butterfly Species Using CNN*, 2023.

12 L. Q. Zhu, M. Y. Ma, Z. Zhang, P. Y. Zhang, W. Wu, D. D. Wang, D. X. Zhang, X. Wang and H. Y. Wang, *Hybrid deep learning for automated lepidopteran insect image classification*, 2017, 10.1080/00305316.2016.1252805.

13 Q. Chang, H. Qu, P. Wu and J. Yi, *Fine-grained butterfly and moth classification using deep convolutional neural networks*, Preprint, 2017, 10.13140/RG.2.2.22642.84161.

14 R. Zhao, C. Li, S. Ye and X. Fang, *Butterfly recognition based on Faster R-CNN*, 2019, 10.1088/1742-6596/1176/3/032048.

15 A. S. Almryad and H. Kutucu, *Automatic identification for field butterflies by convolutional neural networks*, 2020, 10.1016/j.jestch.2020.01.006.

16 A. Sar, T. Choudhury, S. Aich, P. Joshi, B. Pant and B. K. Dewaghan, *Butterfly Image Classification using Modification and Fine-Tuning of ResNet18*, 2024, 10.1109/OTCON60325.2024.10688302.

17 D. Ruaika and S. Uyun, *Optimization of Residual Network Using Data Augmentation and Ensemble Deep Learning for Butterfly Image Classification*, 2023, 10.1109/OTCON60325.2024.10688302.

18 H. Theivaprakasham, *Identification of Indian butterflies using deep convolutional neural network*, 2021, 10.1016/j.aspen.2020.11.015.

19 M. Vakili, M. Ghamsari and M. Rezaei, *Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification*, arXiv:2001.09636, 2020, 10.48550/arXiv.2001.09636.