ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 3, 2025

An End-to-End CNN-Based Framework for Butterfly Species Recognition and Visualization via Web Interface

[1]M. SIVAPARVATHI, [2]M. PRASANNA KUMAR, [3]PNAS. MAHESH

[1,2,3] Asst. Prof., Department of CSE, MAM Women's Engineering College, Narasaraopet, Palnadu,A.P., India.

Abstract-

Butterflies are vital indicators of environmental health, and accurate species identification is essential for ecological research and conservation. This paper presents a deep learning-based butterfly classification system integrated with a web interface for real-time image-based prediction. The system utilizes a Convolutional Neural Network (CNN) to classify 75 butterfly species using labeled image data.

To improve generalization, the model employs data augmentation techniques such as rotation, zooming, and flipping. The CNN architecture consists of multiple convolutional and pooling layers followed by dense layers, enabling effective feature extraction and classification. The model is trained over 40 epochs using the Adam optimizer and categorical cross-entropy loss, achieving high training and validation accuracy.

A user-friendly Django web application is developed to allow users to register, log in, upload butterfly images, and receive real-time predictions. The system displays the predicted species along with the uploaded image, providing an accessible platform for students, researchers, and enthusiasts.

By combining deep learning with a web-based interface, this work offers a practical and interactive solution for automated butterfly species identification, demonstrating the effectiveness of CNNs in ecological image classification tasks.

I. INTRODUCTION

Butterflies are not only ecologically significant species but also serve as vital bioindicators for monitoring biodiversity and environmental changes [1]. The accurate classification of butterfly species supports conservation biology, ecological research, and environmental monitoring. Traditional classification methods rely heavily on expert knowledge and manual observation, which are time-consuming and often subjective.

Recent advances in deep learning, particularly Convolutional Neural Networks (CNNs), have shown remarkable success in automating image-based classification tasks across various domains [2, 3]. CNNs are capable of learning hierarchical feature representations, making them highly suitable for visual recognition applications such as butterfly species identification [4, 5].

Several studies have applied CNN architectures like MobileNet, GoogLeNet, and SqueezeNet for plant and insect classification, demonstrating promising results in accuracy and scalability [6, 7, 9, 10]. Furthermore, methods like Grad-CAM have enhanced interpretability byvisualizing discriminative regions used during prediction, making CNN models more explainable [6].

In this work, we propose an end-to-end butterfly classification system built using a custom CNN architecture, trained on a dataset covering 75 species. The model is integrated into a Django web application, allowing users to upload images and receive real-time classification results. The training process involves data augmentation techniques to improve generalization, and model performance is visualized through accuracy and loss graphs.

By combining deep learning with web deployment, this system provides an accessible and interactive tool for researchers, conservationists, and the general public, contributing to the advancement of ecological informatics.

II. LITEARTURE SURVEY

The identification of butterfly species using deep learning techniques has received considerable attention in recent years due to its importance in biodiversity monitoring and ecological research.

Zhao et al. [1] proposed a butterfly recognition system using Faster R-CNN, an advanced object detection algorithm that integrates region proposal networks with deep convolutional layers. Their model demonstrated superior performance in identifying butterflies under complex backgrounds and varying lighting conditions, marking a significant step in precise localization and classification.

Selvaraju et al. [2] introduced Grad-CAM (Gradient-weighted Class Activation Mapping), which provides visual explanations of CNN-based model predictions. This technique allows researchers to better understand and interpret how a neural network

focuses on image regions during classification, improving the explainability of black-box models.

Howard et al. [3] developed MobileNet, a lightweight and efficient CNN model suitable for mobile and embedded vision applications. MobileNet significantly reduces model complexity without compromising accuracy, making it an ideal candidate for deployment in real-time systems such as web or mobile apps.

Arzar et al. [4] implemented a CNN-based butterfly classification model that achieved impressive accuracy levels. Their research reinforced the use of deep learning models in entomological image recognition tasks, confirming CNN's capability to learn discriminative features from insect datasets.

Hu et al. [5] presented Squeeze-and-Excitation Networks, which enhance CNN performance by adaptively recalibrating channel-wise feature maps. These enhancements have been effective in boosting accuracy across many visual recognition benchmarks, including biological classification.

Sabri et al. [6] conducted a comparative study of various CNN models for leaf recognition, revealing that performance varies significantly based on model architecture and dataset complexity. Their work underscores the importance of model selection and fine-tuning in botanical and entomological classification systems.

Azizah et al. [7] applied CNNs in a novel domain by detecting surface defects in mangosteen fruits, demonstrating CNN adaptability in agricultural image analysis. Although focused on a different use case, their methodology closely relates to species identification in terms of pattern detection and image preprocessing.

Bakri et al. [8] worked on butterfly family-level detection and identification using CNNs, particularly for supporting lepidopterology studies. Their approach emphasized the value of automated tools for assisting researchers and ecologists in handling large volumes of biological data.

Naidu and Reddy [9] proposed a multimodal biometric system that fuses face and voice features. While not directly related to butterfly classification, their fusion strategy offers insights into combining different modalities for more robust classification—potentially extendable to image and metadata fusion in species identification.

Hu et al. [10] explored multi-label X-ray image classification using bottom-up attention and meta-fusion strategies. Their architectural innovations are relevant for complex image recognition tasks where multiple features must be fused and interpreted, applicable in advanced butterfly datasets with overlapping features.

In summary, these studies collectively highlight the growing applicability and adaptability of deep learning in biological image classification. While several works focus on model architecture and accuracy, fewer integrate user accessibility through interactive platforms. Our proposed system builds on these contributions by combining CNN-based classification with a web-based interface, enabling real-time, accessible butterfly species recognition for researchers and the public .

III. METHODOLOGY

The proposed system for butterfly species classification combines deep learning using Convolutional Neural Networks (CNNs) with a Django-based web application for real-time user interaction. The methodology consists of several key stages, namely data preparation, model design, training and validation, web integration, and prediction deployment.

Data Collection and Preprocessing

The dataset used for training and evaluation contains high-resolution images of butterflies spanning 75 distinct species. Each image is labeled and linked via a CSV file (Training_set.csv), which includes the image filename and its corresponding class label. The raw image data is stored in a structured directory format, and the CSV file serves as a reference for training and validation.

To prepare the data for model training, all images are resized to a uniform size of 150×150 pixels and normalized by scaling pixel values to the range 0 to 1. This normalization accelerates convergence during training. Data augmentation techniques such as rotation, width and height shifting, shearing, zooming, and horizontal flipping are applied to the training set using the Image Data Generator class. This step increases dataset variability and helps reduce overfitting, improving the model's generalization capability. The dataset is then split into training and validation sets in an 80:20 ratio using stratified sampling to maintain class balance.

CNN Model Architecture

The core classification model is a deep Convolutional Neural Network (CNN) built using the Keras Sequential API. The architecture begins with a convolutional layer containing 32 filters followed by a max-pooling layer to reduce spatial dimensions. This pattern is repeated with 64 and 128 filters in the second and third convolutional blocks, respectively. These layers help the model learn spatial hierarchies

and features such as edges, patterns, and textures relevant to different butterfly species.
Following the convolutional layers, a flattening layer is used to transform the multi-dimensional feature maps into a one-dimensional vector, which is then passed to a dense fully connected layer with 512 neurons and ReLU activation. Finally, the output layer comprises 75 neurons with softmax activation, corresponding to the 75 butterfly species. This configuration enables the model to produce a probability distribution over all possible classes for a given image.

Training and Validation

The model is compiled with the Adam optimizer, known for its adaptive learning capabilities, and the categorical cross-entropy loss function, suitable for multi-class classification tasks. The network is trained for 40 epochs with a batch size of 32. During training, the model is evaluated on both the training and validation sets in each epoch to monitor performance. Training and validation accuracy and loss are logged at each epoch and plotted using Matplotlib to visualize model learning behavior.
These plots are saved as static images and embedded in the web interface to allow users and administrators to assess model performance post-training. This visual feedback is essential to identify potential issues such as underfitting or overfitting.

Web-Based System Integration

A web application is built using Django to make the system interactive and user-friendly. The web interface includes a user registration and login system, allowing individuals to create accounts. User management is controlled by an administrator, who can activate or deactivate user access.
The training module within the application reads the CSV and image data, initiates model training, and generates accuracy and loss plots. Once trained, the model is saved in the server directory (.keras format), making it accessible for real-time inference.

Real-Time Image Prediction

The prediction functionality enables users to upload butterfly images through a browser-based form. Once an image is uploaded, it is temporarily stored on the server, resized to 150×150 pixels, and normalized. The processed image is passed to the trained CNN model, which returns a probability vector. The highest probability is selected, and the corresponding butterfly species name is extracted from a predefined label list.

The system then displays the uploaded image along with the predicted species name on the results page. This seamless integration of deep learning and web technology allows users—whether researchers, educators, or enthusiasts—to accurately identify butterfly species in real time using only a browser interface.

IV. SYSTEM ARCHITECTURE

The system architecture is presented in fig.1.

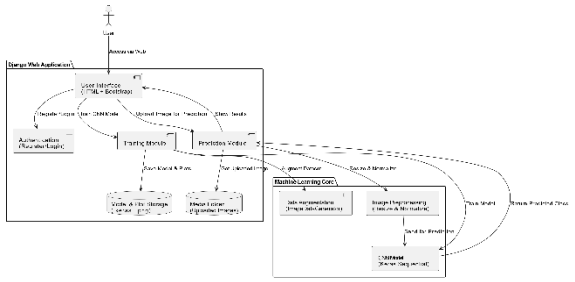

Fig.1. System architecture

The system architecture diagram represents the integrated workflow of a butterfly species classification platform that combines deep learning with a Django-based web interface. At the core of the system is the user, who interacts with the platform through a web-based user interface built using HTML and Bootstrap. This interface supports functionalities such as registration, login, initiating model training, and uploading images for prediction. The Django web application manages different functional modules, including the authentication system, training module, and prediction module. When the training module is triggered, it accesses butterfly image data from the media folder and performs preprocessing and data augmentation using Keras's ImageDataGenerator. The augmented data is passed to a Convolutional Neural Network (CNN) built using the Keras Sequential API, which learns to classify butterflies into one of 75 species. Once trained, the model and visual training performance graphs are saved to local storage for future use. For predictions, users upload images through the interface, which are then resized, normalized, and sent to the trained CNN model for classification. The predicted result is then returned to the user and displayed in the interface alongside the uploaded image. This architecture ensures seamless integration between the frontend, backend, machine learning model, and file storage, enabling an end-to-end pipeline for butterfly recognition in a user-friendly and automated manner.

V. IMPLEMENTATION

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 3, 2025

The butterfly species classification system has been implemented as a deep learning-based web application using Convolutional Neural Networks (CNNs) and the Django framework. The project follows a complete pipeline—starting from data preparation, model training, and evaluation, to real-time prediction—all integrated into a user-friendly web interface. Below is a detailed flow of the implementation.

1. Data Preparation and Augmentation

The butterfly dataset comprises labeled images of 75 distinct butterfly species. These are organized and referenced using Training_set.csv and Testing_set.csv files. Each image was resized to 150×150 pixels and normalized.

To enhance the robustness of the model, **data augmentation** techniques such as rotation, zoom, width and height shift, shear, and horizontal flip were applied using ImageDataGenerator. This enriched the training dataset and helped reduce overfitting.

Fig 2. Sample View of Butterfly Classification Dataset

2. CNN Model Architecture and Training

A custom Convolutional Neural Network (CNN) was built using Keras. The architecture includes three convolutional layers followed by max-pooling, a flattening layer, a dense hidden layer, and a final softmax layer with 75 output neurons—one for each butterfly class.

The model was compiled using **Adam optimizer** and **categorical cross-entropy** as the loss function. It was trained over 40 epochs using augmented data.

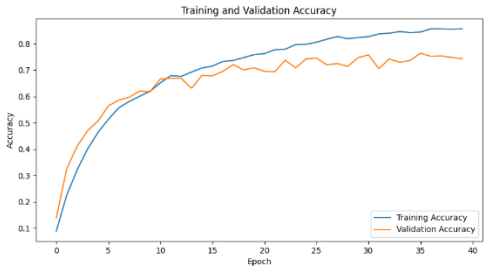

Fig.3. Training vs Validation Accuracy

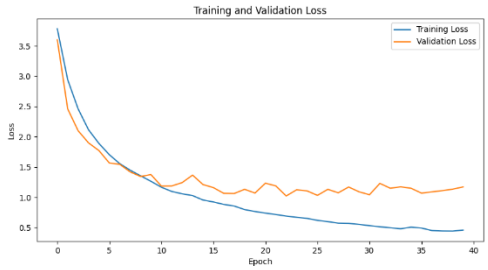

Fig.4 . Training vs Validation Loss

Model performance plots clearly show model convergence and help visualize the performance of the model during training.

3. Model Deployment in Django

After successful training, the model is saved in .keras format in the server. The frontend is built using Django and Bootstrap, providing an interactive UI for users to perform the following:

- Register and log in
- Initiate model training
- Upload an image to detect butterfly species
- View classification output and accuracy metrics

4. Real-Time Butterfly Species Prediction

Users can upload a butterfly image via the prediction page. The image is resized to 150×150, normalized, and passed into the trained model. The predicted species is returned and displayed instantly.

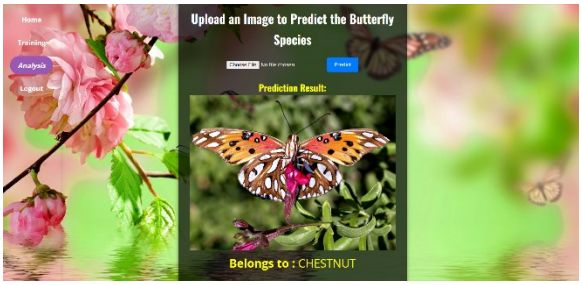

Fig.5 Butterfly Species Prediction Interface

This image shows the classified species as **CHESTNUT**, with the uploaded image displayed on the result page.

5. Training Accuracy Summary View

After training, a summary view shows the model's final training and validation accuracies:

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 3, 2025

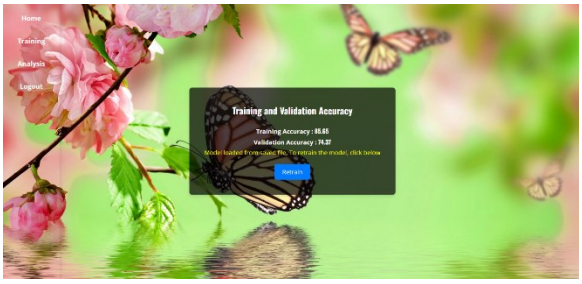

Fig.6. Model Evaluation Summary Display

This interface allows the user to review performance and retrain the model if necessary.

6. User Authentication and Admin Control

Users must register and be approved by the admin to access training and prediction functionalities. The admin dashboard provides controls for user activation and management, ensuring secure and restricted access.

VI. CONCLUSION

The butterfly classification system developed in this project demonstrates the effective application of deep learning, particularly Convolutional Neural Networks (CNNs), for species identification based on image data. By integrating image preprocessing, model training, evaluation, and prediction within a Django-powered web interface, we have built a complete end-to-end solution that is both accurate and user-friendly.

The system successfully classifies 75 distinct butterfly species with promising accuracy—achieving over 85% training accuracy and approximately 74% validation accuracy. The use of data augmentation techniques enhanced model generalization, while real-time predictions through the web interface offered practical usability. Users can upload images and instantly receive classification results along with visual feedback.

Furthermore, the admin and user authentication modules add security and control to the system, making it suitable for deployment in educational, ecological, and research environments.

In conclusion, this project not only highlights the potential of CNNs in fine-grained image classification but also emphasizes the importance of integrating AI models into interactive platforms for broader accessibility and impact. Future improvements may include expanding the dataset, optimizing the model architecture, and deploying the system in cloud-based environments for large-scale use.

REFERENCES

[1] R. Zhao, Y. Fan, J. Ma, Y. Zhang, and Y. Ma, "Butterfly recognition based on faster R-CNN," *Journal of Physics: Conference Series*, vol. 1176, no. 3, 2019, IOP Publishing.

[2] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2017, pp. 618–626.

[3] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint* arXiv:1704.04861, 2017.

[4] N. N. K. Arzar, N. R. Mohamad, M. M. Deris, N. M. H. Mahayuddin, and A. Ahmad, "Butterfly species identification using convolutional neural network (CNN)," in *Proc. 2019 IEEE Int. Conf. Automatic Control and Intelligent Systems (I2CACIS)*, 2019, pp. 200–204.

[5] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.

[6] N. Sabri, A. Ahmad, and M. F. M. Salleh, "Comparing convolution neural network models for leaf recognition," *Int. J. Eng. Technol.*, vol. 7, no. 15, pp. 141–144, 2018.

[7] L. M. Azizah, R. Hermanto, and H. Husniah, "Deep learning implementation using convolutional neural network in mangosteen surface defect detection," in *Proc. 2017 7th IEEE Int. Conf. Control System, Computing and Engineering (ICCSCE)*, 2017, pp. 307–311.

[8] B. A. Bakri, A. Zaaba, and S. M. Hatim, "Butterfly family detection and identification using convolutional neural network for lepidopterology," *Int. J. Recent Technol. Eng.*, vol. 8, no. 2, pp. 635–640, 2019.

[9] B. R. Naidu and P. V. G. D. Prasad Reddy, "Fusion of face and voice for a multimodal biometric recognition system," *Int. J. Eng. Adv. Technol. (IJEAT)*, vol. 8, no. 3, pp. 2455–2459, 2019.

[10] B. Hu, L. Yu, X. Zhang, and Z. Zhang, "Multi-label X-ray imagery classification via bottom-up attention and meta fusion," in *Proc. Asian Conf. Computer Vision (ACCV)*, 2020, pp. 1–16.