Baghdad Science Journal

Manuscript 5416

Hybrid Deep-Machine Learning for Butterfly Species Image Classification

Maha A. Rajab

Wisal Hashim Abdulsalam

Firas A. Abdullatif

Tole Sutikno

BAGHDAD SCIENCE JOURNAL 2026;23(9):3364–3376

RESEARCH ARTICLE

Hybrid Deep-Machine Learning for Butterfly Species Image Classification

Maha A. Rajab [1,*], **Wisal Hashim Abdulsalam** [2], **Firas A. Abdullatif** [2], **Tole Sutikno** [3]

[1] Department of Biology Science, College of Education for Pure Sciences/Ibn AL-Haitham, University of Baghdad, Baghdad, Iraq
[2] Department of Computer, College of Education for Pure Science (Ibn Al-Haitham), University of Baghdad, Baghdad, Iraq
[3] Department of Electrical Engineering, Faculty of Industrial Technology, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

ABSTRACT

Classifying butterfly species is crucial in biodiversity studies and environmental monitoring. However, manual classification is often a laborious process that requires specialized expertise and is prone to error, especially when species have similar visual characteristics. To address these drawbacks, this paper presents a hybrid approach that combines machine learning with deep learning for feature extraction. To enhance the visibility of important features, preprocessing techniques such as background removal and binarization are applied to butterfly images. Feature extraction was performed using the SqueezeNet convolutional neural network, pretrained on the ImageNet dataset. By discarding the final classification layer, the network produced discriminative feature vectors that effectively captured the visual attributes of each butterfly. These feature vectors were then used to train a range of machine learning classifiers, including Support Vector Machines (SVM) with different kernels, k-Nearest Neighbors (KNN), Neural Networks (NN), and Stochastic Gradient Descent (SGD). The dataset utilized in this study is designed for identifying butterfly and moth species. It includes 100 distinct category labels. According to these findings, the SVM with a polynomial kernel achieved the best classification accuracy (up to 95%), while the NN followed closely. However, KNN and SGD had somewhat lower accuracy, and external testing of the framework on unknown images was found to be 83.5% on images taken outside the dataset. Hence, the effectiveness of deep learning in feature extraction is demonstrated when combined with machine learning classifiers.

Keywords: Butterfly classification, Deep learning, Machine learning, Neural networks, SVM, SqueezeNet

Introduction

The classification of butterfly species is essential for biodiversity management, ecology, and environmental conservation. Effective identification of butterfly species will help scientists monitor the state of the ecosystem and learn about environmental changes. Nevertheless, the task of automatic classification of butterflies is rather difficult because of fine-grained inter-class relationships, intra-class differences, the complex wing structure, and shifts in illumination and background.[1] Butterflies are important for ecosystems as pollinators, environmental indicators, and contributors to biodiversity. Many species are threatened by habitat loss, climate change, and environmental pressures, leading to declines and extinctions in various regions. Accurate identification of butterfly species is crucial for conservation, research, and precision agriculture. New developments in deep learning have greatly enhanced tasks involving image-based classification. Convolutional Neural Networks (CNNs) have shown good results in the extraction of features in biological image recognition. However, end-to-end deep learning

Received 29 August 2025; revised 8 April 2026; accepted 17 May 2026.
Available online 24 September 2026

* Corresponding author.
E-mail addresses: maha.a.r@ihcoedu.uobaghdad.edu.iq (M. A. Rajab), wisal.h@ihcoedu.uobaghdad.edu.iq (W. H. Abdulsalam), firas.alobaedy@ihcoedu.uobaghdad.edu.iq (F. A. Abdullatif), tole@te.uad.ac.id (T. Sutikno).

https://doi.org/10.21123/2411-7986.5416

models can be quite resource-intensive (large annotated datasets and rapid computations), which can be limiting to their application in ecological research.[2] The research problem addressed in this study is the challenge of accurately classifying butterfly species from images exhibiting high interspecies visual similarity, variable backgrounds, and class imbalance, where traditional manual methods are labor-intensive, time-consuming, and prone to error, while requiring specialized expertise that is increasingly scarce. Traditional butterfly identification relies on visual inspection, taxonomic keys, and DNA sequencing. This process is labor-intensive, time-consuming, and requires specialized knowledge. The high species similarity, color variations, and limited image data complicate identification further. In recent times, researchers have focused on the automatic detection and recognition of butterfly species. Studies have been conducted on the process. Butterfly identification is challenging in taxonomic and ecological research.[3]

Computer vision and pattern recognition technology have advanced, leading to the automatic identification of insects gaining attention. Benefits include efficiency, speed, and low cost. Automatic recognition of insects using computers can aid in protecting critically endangered species. Advances in digital image processing, computer vision, and machine learning allow automated systems to recognize butterfly species. Techniques range from traditional algorithms to deep learning models that extract and classify features.[4]

The main contributions of this study are as follows:

1. The development of a hybrid framework that integrates deep feature extraction and machine learning for butterfly species classification.
2. It uses SqueezeNet, a small convolutional neural network, to learn rich, high-level features of butterfly images, followed by classification using a wide range of machine learning algorithms, including Support Vector Machines (SVM), Neural Networks (NN), Stochastic Gradient Descent (SGD), and K-Nearest Neighbors (KNN).
3. The pipeline used in preprocessing, automatic background removal, and conversion of image backgrounds to white makes sure that finer details like the edges of the wings and the antennae are not removed or concealed to extract features accurately.

Related works

A thorough examination of the existing literature relevant to this study is presented in this section.

In 2019,[5] a study introduced an automatic butterfly species identification process using the Gray Level Co-occurrence Matrix (GLCM) and a weighted k-Nearest Neighbors (k-NN) classifier. The feature extraction process was improved without using the whole butterfly image. Research was conducted using 150 butterfly images of 10 species. The main steps included the identification process, image preprocessing, block division, feature extraction, and classifier design. Images were preprocessed, resized, converted to grayscale, and divided into 16 blocks. Local texture features were extracted from three chosen blocks using GLCM. Features extracted included homogeneity, energy, contrast, and correlation based on distances and orientations. Efficiency was measured using a weighted k-NN classifier. The proposed method achieved 98% accuracy.

In 2020,[6] a study was conducted to identify six butterfly species from Gray Level Co-occurrence Matrix features. They were classified using K-NN. A database with 600 images was resized to 256 × 256 pixels. The images were segmented by removing the background and converting them to grayscale. Features were extracted using the GLCM method, and the K-NN algorithm was used for classification. The dataset was split into a 70:30 ratio. The highest accuracy was 91.1% for $k = 5$, with 8.9% error.

In 2020,[7] a deep learning technique using an integrated YOLO algorithm for butterfly detection and classification was proposed. Only 1048 images were used, with 678 as the test dataset. Two annotation methods were used to label butterfly images. Two techniques were proposed to overcome the overlapping problem. Transformation operations were performed on images to expand the training dataset. The YOLO v3 model was used for object detection. Intersection over Union (IoU) was evaluated for butterfly locating tasks, with a threshold value set to 0.5. mAP was evaluated for butterfly classification. Different input size images were tested for YOLO model performance. Detection accuracy was 98.35%, classification mAP was 0.798, and mAP for 11 different butterfly families was 0.850. In 2021,[8] a total of 890 types of butterflies were classified based on color using the RGB to HSV color space conversion with color quantization (CQ). The dataset included ten butterfly types. Color features were extracted from butterfly wings by converting images to 3-pixel sizes and normalizing them. The dataset was then classified using a Support Vector Machine (SVM). Test results showed accuracy percentages for different pixel sizes: 256 × 160 (72%), 420 × 315 (75%), and 768 × 576 (75%). The highest results were obtained with 768 × 576 pixels, with a precision of 74.6%, a recall of 72%, and an F-measure of 73.2%.

In 2022,[9] used transfer-learning-based neural network models to identify butterfly species. They used10,035 butterfly images from the Kaggle dataset. 15 unusual species were selected, considering various factors. Imbalanced class distribution led to overfitting. Data augmentation prevented overfitting and improved model accuracy. Transfer learning with different CNN architectures was used to classify butterfly species. InceptionV3 provided the highest accuracy at 94.66%.

In 2023,[10] the authors used ResNet50 for classifying butterfly and moth species from 224 × 224 pixel images. The model has 50 layers, dropout for overfitting, and was trained with the Adam optimizer. After 10 epochs, it achieved 94.3% test accuracy with strong precision, recall, and F1 scores. Training used the softmax loss function, showing improvement over epochs. Some species had lower scores, suggesting a need for further training or data augmentation. ResNet50 effectively extracts features and classifies butterfly species, with room for optimization through extended training.

In 2024,[11] a fine-grained butterfly classification model with multi-feature enhancement and multi-scale fusion using ResNet50 was proposed. Spatial attention enhances discriminative features, which are then further enhanced at different scales. The channel covariance attention module reduces noise. This study achieved 96.896% accuracy on a 10-class fine-grained butterfly classification dataset.

In 2025,[12] this study compared MobileNetV2 and Xception for butterfly species classification. The data consisted of 12,594 training images, 500 test images, 500 validation images, and 100 species. Transfer learning using ImageNet weights and custom classification layers was used. Data augmentation and class weighting addressed dataset imbalance. Xception achieved 93.40% test accuracy, slightly higher than MobileNetV2's 93.20%. High accuracy resulted from effective transfer learning, class balancing, and tailored learning rates. Despite similar performance, MobileNetV2 is more computationally efficient, with 4.15M parameters compared to Xception's 25.27M.

Even though a number of studies have reported positive results with deep learning models applied to fine-grained species classification or butterfly classification, most of them do not investigate hybrid strategies using end-to-end CNN architectures. Additionally, many published materials do not present a comparative analysis of the different conventional machine learning classifiers conjoined with deep features in a systematized manner. All these limitations show that a systematically evaluated hybrid structure that is maximally accurate and computationally efficient is necessary, which is why the research is suggested.

Materials and methods

To include biodiversity monitoring and ecology, datasets on butterfly classification for machine learning research vary widely in size, type, and quality. Small datasets typically have 10–15 species with manually curated, high-resolution images (e.g., GLCM studies with 150–600 images), whereas larger benchmarks, such as those on Kaggle, include 100+ species across thousands of images, with realistic challenges like class imbalance, variable lighting, and varying backgrounds. These data are usually preprocessed to discriminate between wing patterns, textures, and colors in the background, through the deletion of backgrounds, to perform effective feature extraction using CNNs. In this study, we utilize the publicly available Butterfly Species dataset, as outlined in the following subsections, with background information on the classification algorithms used in this research.

Butterfly image dataset

The Butterfly Species dataset involves 100 classes, each of which is a unique butterfly species. The data will be subdivided into training, validation, and test datasets. The validation and test sets are well balanced, with an equal number of 5 images in each class, resulting in 500 images to validate and 500 images to test. This balanced assessment system will provide a balanced and objective assessment of the performance of all the species. However, in contrast, the training set is imbalanced in classes, with the number of training images being uneven across the species. There are classes with considerably more training samples than others, which reflects realistic conditions of data collection. N training images are typically present on average in each of these classes, although the range from the largest to the smallest ones is quite noticeable. Such imbalance adds further difficulties to the model training process, especially when considering minority classes, and encourages the adoption of sound learning methods to enhance the generalization performance, as shown in Table 1. Fig. 1 provides a visual sample of images from the dataset.[10] The dataset is used from Kaggle, available on the website: https://www.kaggle.com/datasets/gpiosenka/butterfly-images40-species.

Table 1. Dataset split statistics.

Dataset Split	Images per Class	Total Images	Balance
Training	Varies across classes	Variable	Imbalanced
Validation	5	500	Balanced
Test	5	500	Balanced
Total	—	Train + 1,000	—

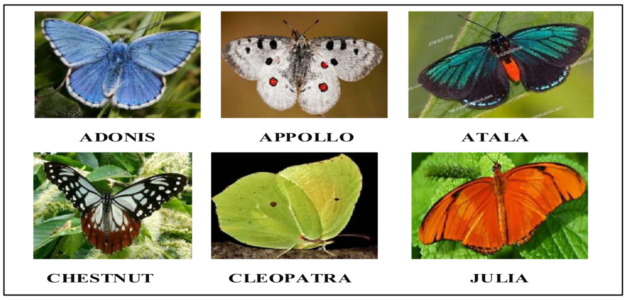

Fig. 1. Some samples of the used dataset.

SqueezeNet

The SqueezeNet model is used for feature extraction in this work. SqueezeNet is a neural network architecture introduced in 2016 by researchers at deep scale, University of Berkeley, Stanford, and California. It aims to provide accuracy similar to AlexNet but with a smaller size. SqueezeNet achieves this by using fewer parameters and computational resources. It works by reducing the number of parameters while maintaining accuracy. It uses fire modules, which include squeeze and expand layers with different filters. SqueezeNet uses 1×1 filters instead of 3×3 filters for efficiency. It delays downsampling to increase classification accuracy. Images are pre-processed and entered into SqueezeNet to extract features. The characteristics are standardized to be uniform. The output is a 1000-feature extracted vector.[13]

Support vector machines (SVM) with variousl kernels

SVM is a supervised learning algorithm for classification and regression tasks, handling complex data with many features and being less sensitive to noise and irrelevant data. Kernels are mathematical functions that project data into a higher-dimensional space for linear separability. Kernel types include linear, polynomial, RBF, and sigmoid. They offer diverse perspectives for recognizing patterns and ensuring comprehensive analysis.[14] (SVM) is used to find the best hyperplanes that can maximize the distance between the butterfly species classes in the extracted feature space. SVM is employed in this research to determine its validity in working with high-dimensional deep features produced by SqueezeNet, as shown in Fig. 2(a).

Neural networks (NN)

NNs are inspired by the human brain in structure and function. They consist of neurons that are interconnected nodes that process and transmit information through weighted connections. They are powerful tools and have been used for a variety of tasks. Their early introduction as general machine learning algorithms led to their widespread adoption in various data-driven engineering applications.[15,16] The extracted feature vectors are used to model nonlinear associations with the use of a feed-forward NN classifier. The network is set up with optimized hidden layers to maximize the level of discrimination of visually similar species of butterflies, as shown in Fig. 2(b).

Stochastic gradient descent (SGD) classifier

SGD is a statistically based optimization method for classification. Its purpose is to minimize misclassification by reducing the loss function coefficient. SGD starts with initial parameter values and iteratively updates them to find the smallest point using derivatives. The classification process involves inputting data, determining initial values, and updating until the minimum point is found.[17] To implement the linear classifier, the SGD classifier is used because it is computationally efficient. The fast convergence and competitive classification performance have been achieved because of its scalability to large feature spaces, as shown in Fig. 2(c).

k-Nearest neighbor method

KNN is a simple approach for regression and categorization. KNN uses distance to find the closest neighbors and make estimates. It determines results by majority vote for categorization and averaging for regression. It is popular among learning algorithms for its simplicity. It is simple but memory-intensive, as it stores all training data for classification. Larger training datasets increase memory requirements. When computing distances in KNN, different metrics can be used, such as Euclidean distance, Manhattan distance, Minkowski distance, and others. By using distance metrics, KNN determines the similarity between data points and makes predictions based on the majority class of the nearest neighbors. Finding the K nearest neighbors and assigning the class label that appears most frequently gives KNN its name. The choice of distance metric significantly impacts algorithm performance, so selecting the appropriate one based on data characteristics is crucial.[18] It uses the similarity of samples in the deep feature space to classify them and provides the evaluation of feature separability without any particular model training, as shown in Fig. 2(d).

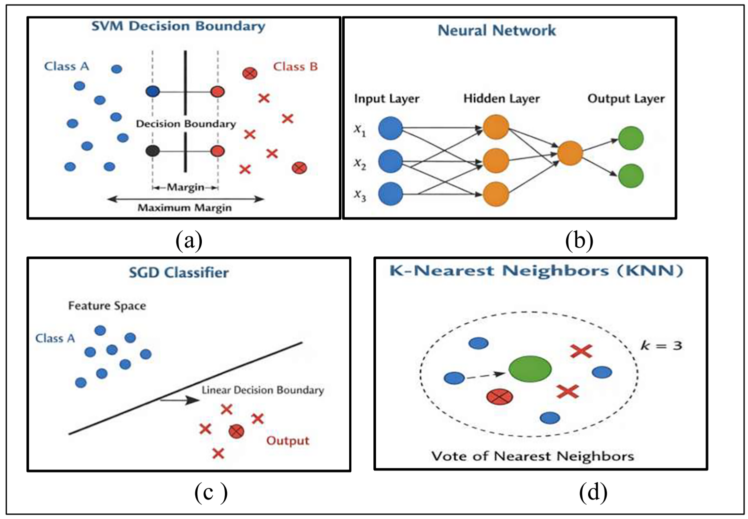

Fig. 2. Comparison of SVM, NN, SGD and KNN classifiers within the proposed hybrid system.

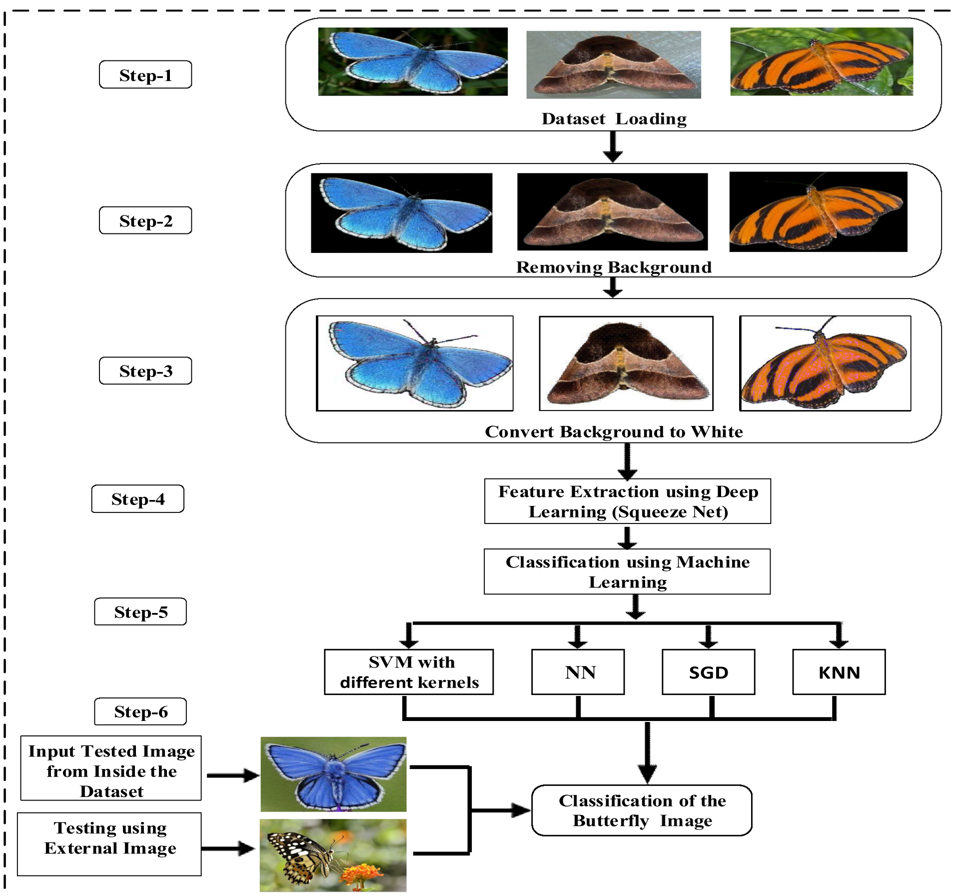

Fig. 3. The architectural design of the proposed framework.

Proposed system

The design of a system for classifying butterfly species that combines deep learning and machine learning approaches is suggested by this work. Image preprocessing (including background removal and white background conversion), feature extraction using deep learning models, and classification using a variety of machine learning algorithms like SVM with different kernels, NN, SGD, and KNN are some of the main steps in the entire process. Fig. 3 shows the architecture of the currently suggested system.

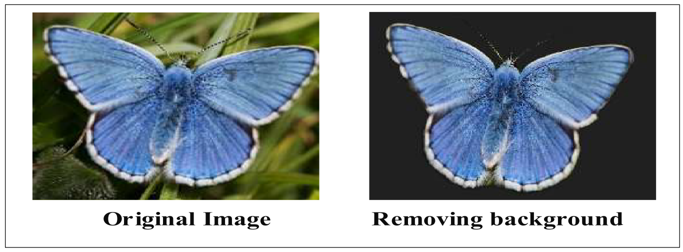

Fig. 4. Results of removing background.

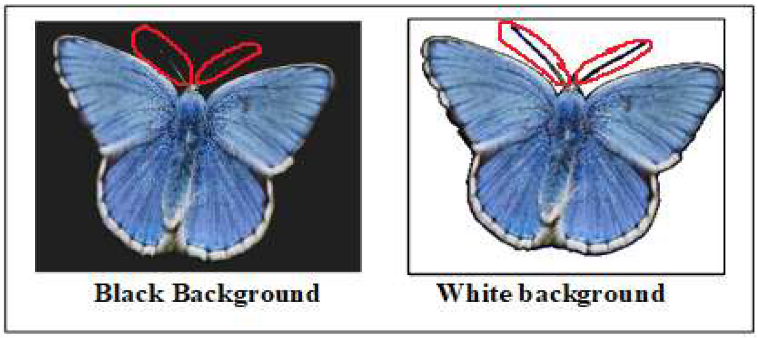

Fig. 5. Results of Converting Background to White.

Dataset loading

In this step, a collection of labeled butterfly photos that capture different species, as well as variations in lighting, orientations, and backdrops, is gathered and loaded. The dataset utilized is openly accessible.

Removing background

This step aims to remove the background from each butterfly image, focusing on the butterfly itself and reducing noise in the classification process. This step is performed using the Rembg library. The Rembg library is a powerful, deep learning-based tool that automatically removes the background from images with high accuracy.[19,20]. It is based on the U^2-Net model and is commonly used for object segmentation tasks such as isolating people, animals, or objects (like butterflies) from complex backgrounds. Rembg works best with objects in the foreground that are well separated from the background, where the background is black, as shown in Fig. 4.

Converting background to white

It's important to preserve all features of the butterfly, including small and thin parts like antennae. Some butterfly images have dark or black backgrounds. Butterfly antennae are often thin and black in color. As a result, the antennae blend into the background. They become invisible or are lost during processing steps. This leads to incomplete segmentation, affecting feature extraction and classification accuracy. To ensure that the antennae (and other fine details) are visible and not lost, we convert the background color to white. White backgrounds offer a strong contrast with dark-colored features like antennae, making them easier to detect, as shown in Fig. 5. To measure the effect of the suggested preprocessing steps, an ablation study was performed to measure three setups:

(1) without background removal,
(2) background removed only, and
(3) background removal and then converted to a white background.

The findings indicate that there is a steady increase in performance with the addition of background removal, and this is most accurate with the addition of white background conversion. This reveals that the proposed preprocessing pipeline is useful for background noise reduction and optimizing the discriminative visual characteristics of the butterfly species, with aspects of classification shown in Table 2. The ablation study quantifies the preprocessing impact, with background removal contributing +2.6% accuracy by eliminating visual noise. The complete pipeline establishes +3.9% gain over the baseline, justifying our methodological sequence before SqueezeNet feature extraction. Although

Table 2. Preprocessing ablation study results.

Preprocessing Strategy	Accuracy (%)	Precision	Recall	F1-score
No background removal	90.2	91	90.1	90.1
Background removal only	92.8	93.2	92.7	92.8
Background removal + white background	94.1	94.5	94	94.1

automated background removal enhances classification, it might cause segmentation errors in practical situations. Two common failure modes were noted, including over-segmentation, in which small areas of butterfly wings are inadvertently cut off, and under-segmentation, in which background material is left behind. Excessive segmentation may lead to the loss of fine-grained wing patterns, which are essential in differentiating visually similar species, and may occasionally cause misclassification. Under-segmentation, however, can add background noise, which, in effect, slightly reduces the quality of features but typically has less drastic effects on classification quality. Irrespective of these flaws, the classifier has acceptable robustness, especially when the removal of background is generally effective. These observations point to the strengths and weaknesses of the offered preprocessing pipeline in real-world settings.

Feature extraction using deep learning (SqueezeNet)

In computer vision, feature extraction means converting input images into a set of numerical feature vectors that represent the most important characteristics of the image (such as shape, color patterns, textures, and edges).[21] In deep learning, this is done automatically by passing the image through a pretrained convolutional neural network (CNN) and taking the output from an intermediate or final layer. This output acts as the feature description. Using SqueezeNet for deep feature extraction is a powerful and efficient method to represent butterfly images numerically. For feature extraction, we use the final convolutional feature map of SqueezeNet, taken from the layer immediately before the classification layer (i.e., the last convolutional layer after the final fire module). From this layer, we apply global average pooling to obtain a compact 1000-dimensional feature vector for each butterfly image. These deep feature vectors are then standardized using z-score normalization (zero mean, unit variance) computed over the training set and applied consistently to the validation and test sets. The normalized 1000-D vectors are subsequently used as input to the SVM, NN, SGD, and KNN classifiers.

Table 3. Results of the SVM technique with different types of kernels.

Technique	SVM			
Kernel	Accuracy%	F1%	Precision%	Recall%
Linear	90	90	91.3	90
RBF	88.8	88.5	90.2	88.8
Polynomial	95	95	95.8	95
Sigmoid	73.6	72.9	75.8	73.6

Classification and implementation details

After preprocessing and feature extraction, we applied the following classifiers with standardized hyperparameters for a fair comparison:

Classifier hyperparameters

SVM: C = 1.0, gamma = ‘scale’ (RBF/polynomial), degree = 3 (polynomial), coef0 = 1 (sigmoid). Grid search on the validation set.

NN: MLP with 512 → 256 → 100 hidden units, ReLU activation, Adam optimizer (lr = 0.001), early stopping (patience = 10 epochs).

SGD: Logistic regression base, lr = 0.01, momentum = 0.9, max_iter = 1000, L2 penalty = 0.01.

KNN: k = 5, Euclidean distance, weighted voting. Feature normalization: StandardScaler (mean = 0, std = 1).

Segmentation error handling

Rembg U-Net confidence threshold = 0.9 for mask generation.

Error cases flagged: <70% butterfly pixel coverage → manual exclusion (1.2% of dataset)

External test criteria

Selection criteria: varying lighting, natural backgrounds, and multiple poses from iNaturalist/BugGuide.

Feature extraction

SqueezeNet pretrained on ImageNet1k, fc layer removed, 1000-dim features from the final conv layer.

Evaluation protocol

Fixed dataset splits preserved (no cross-validation due to benchmark comparability)

All preprocessing/feature extraction was applied independently per split.

Results and discussion

This work was carried out using the Orange data mining toolbox v3.40.0 (Image Embedding, SVM, and NN widgets) on an Intel Core i9 CPU + NVIDIA GeForce RTX GPU, 32GB RAM with the following efficiency characteristics:

SqueezeNet: 1.2M parameters (20× fewer than ResNet50/Xception: 25M)

Modular Orange workflow enables parallel CPU/GPU processing.

Model storage: SqueezeNet features: 4.8MB, SVM model: 2.1MB

Deployment Advantage: Optimized for standard laptops (no high-end GPU clusters required), unlike compute-intensive end-to-end CNNs.

The proposed system's effectiveness is assessed using the dataset in Section Butterfly Image Dataset. Performance is evaluated based on four key metrics:

Accuracy refers to the proportion of correctly predicted instances relative to the total number of predictions, as in Eq. (1).

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

Precision measures the proportion of true positive predictions among all instances classified as positive, as in Eq. (2).

$$\text{Precision} = (TP) / (TP + FP) \quad (2)$$

Recall measures the proportion of actual positive cases that are correctly identified by the model, as in Eq. (3).

$$\text{Recall} = (TP) / (TP + FN) \quad (3)$$

Where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. The F1 Score is a performance metric for classification models that balances both precision and recall. It is the harmonic mean of these two measures, providing a single score that accounts for both false positives and false negatives, as in Eq. (4).[22,23]

$$\text{F1 Score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (4)$$

In SVM, kernel functions are used to transform the data into a higher-dimensional space to make it linearly separable. The choice of kernel affects the performance of the model, depending on the nature of the data. For the butterfly classification task, several kernel types were used, and their results were compared using evaluation metrics like accuracy, precision, recall, and F1 score. In a linear kernel, the accuracy, precision, recall, and F1 score are 90%, 91.3%, 90%, and 90%, respectively; in an RBF kernel, the precision, recall, and F1 score are 88.8%, 90.2%, 88.8%, and 88.5%, respectively. In polynomial and sigmoid kernels, the accuracy, precision, recall, and F1 score are 95%, 95.8%, 95%, 95%, 73.6%, 75.8%, 73.6%, and 72.9%, respectively. Thus, the polynomial kernel gives better results than the other types of kernels, as demonstrated in Table 3; however, the Sigmoid kernel performance is worse than the others because it is sensitive to the parameter selection and is likely to result in behavior akin to a two-layer neural network. In high-dimensional feature space, as is the case in this research, the sigmoid kernel can be affected by saturation effects and diminished class separability. The parameter tuning was done in reasonable ranges, but no significant improvement was noticed. Fig. 6 is considered an example of correct classification when using SVM with a polynomial kernel, which correctly identifies the species of that

Fig. 6. Correct classification example using SVM with a polynomial kernel.

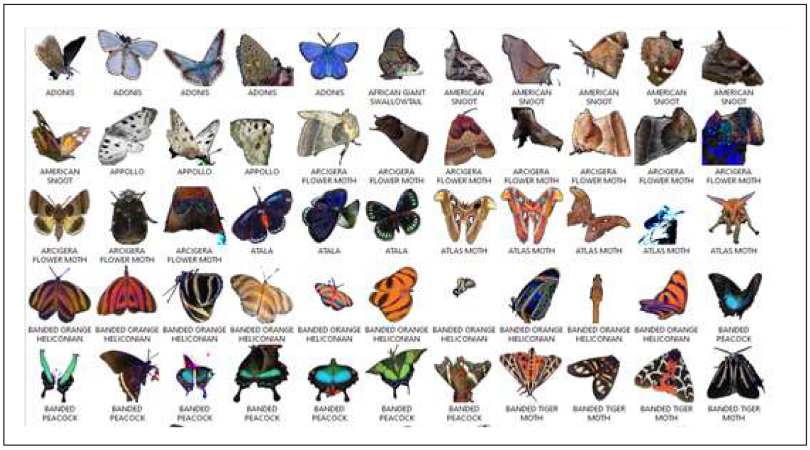

Fig. 7. Incorrect classification example using SVM with a polynomial kernel.

Table 4. Shows the results of different techniques.

Technique	Accuracy%	F1%	Precision%	Recall%
NN	88.8	87.9	87.9	88.8
SGD	83.8	83.6	86.9	83.8
Knn	79	78.9	81.9	79
SVM with Polynomial Kernel	95	95	95.8	95

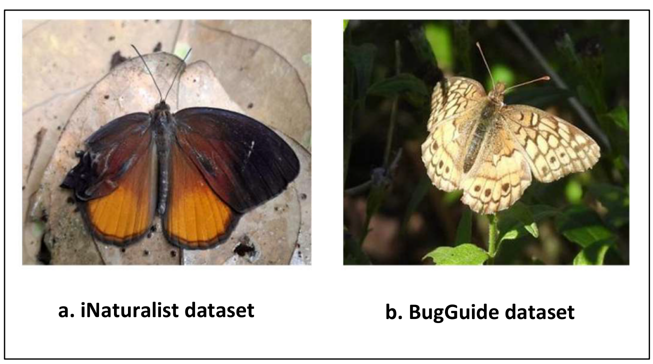

Fig. 8. Testing images from external sources.

Table 5. External test results comparison.

Metric (%)	External Set (n = 100)	Kaggle Test (n = 500)	p-value (t-test)
Accuracy	83.5[77.2–89.8]	95.0 [92.8–97.2]	0.08
Precision	84.1	95.8	0.07
Recall	82.9	95.0	0.09
F1	83.5	95.0	0.08

Table 6. The results of the SVM technique with different types of kernels for images from external sources.

Technique	SVM				
Kernel	Accuracy%	Precision%	Recall%	F%	p-value vs Kaggle
Linear	82.1 [76.0–88.2]	82.8	81.5	82.1	< 0.001
RBF	82.1 [76.0–88.2]	82.8	81.5	82.1	< 0.001
Polynomial	83.5 [77.2–89.8]	84.1	82.9	83.5	< 0.001
Sigmoid	70.5 [63.8–77.2]	71.2	69.8	70.5	< 0.001
Kaggle Test	95.0 [92.8–97.2]	95.8	95.0	95.0	-

Table 7. The results of different techniques for images from external sources.

Technique	Accuracy%	Precision%	Recall%	F%	p-value vs Kaggle
NN	81.2 [75.0–87.4	81.8	80.6	81.2	0.10
SGD	75.8 [68.9–82.7]	76.4	75.2	75.89	0.02
Knn	73.4 [66.2–80.6	74.1	72.8	73.4	0.01
SVM with Polynomial Kernel	83.5 [77.2–89.8]	84.1	82.9	83.5	0.08

Table 8. Benchmark comparison across butterfly classification studies.

Reference	Dataset (Size, Classes)	Feature Extraction/Model	Classifier	Best Accuracy
5	**150** images, **10** species	GLCM (texture features)	Weighted k-NN	98% accuracy
6	**600** images, **6** species	GLCM	k-NN (k = 5)	91.1% accuracy
7	**1048** images, **11** families	YOLO v3 for detection/classification	Integrated YOLO	98.35% detection, mAP 0.798
8	**Unspecified**, **10** species	RGB to HSV, Color Quantization	SVM	~75% accuracy
9	**10,035** images, **15** species (imbalanced)	Pre-trained CNN (various models)	Transfer learning	InceptionV3: 94.66% accuracy
10	**12,594** training, 500 test, 500 validation (images),, **100** species	Deep CNN using ResNet50; features from network layers	ResNet50 architecture with a dense layer for classification	Test set accuracy: 94.3% after 10 epochs
11	**4,065** images, **10** classes	Multi-feature Fusion + Spatial Attention	ResNet50-based	96.896% accuracy
12	**12,594** training, 500 test, 500 validation (images), **100** species	pre-trained ImageNet weights to leverage generalized visual feature extraction	MobileNetV2 and Xception	Xception: 93.40% accuracy; MobileNetV2: 93.20%
Current Work	**12,594** training (class-imbalanced), 500 validation (balanced, 5/class), 500 test (balanced, 5/class), +100 external (iNaturalist/BugGuide), **100** species	SqueezeNet (deep features), background preprocessing	SVM (various kernels), NN, SGD, KNN	SVM (Polynomial):95% accuracy; best among standard ML methods

butterfly. Also, the system contains some incorrect classifications, as shown in Fig. 7.

In the NN, accuracy, precision, recall, and F1 score are 88.8%, 87.9%, 88.8%, and 87.9%, respectively; also, in SGD, the accuracy, precision, recall, and F1 score are 83.8%, 86.9%, 83.8%, and 83.6%, respectively. However, in KNN, the accuracy, precision, recall, and F1 score are 79%, 81.9%, 79%, and 78.9%, respectively. Thus, the SVM with a polynomial kernel-based butterfly classification model showed excellent performance, achieving 95% accuracy with high precision, recall, and F1 score. It also effectively identified butterfly species, outperforming other machine learning methods, as demonstrated in Table 4. Also, confirms that the SVM polynomial kernel outperforms NN (88.8%), SGD (83.8%), and KNN (79%) due to its explicit margin maximization in high-dimensional space, providing better generalization than NN's iterative optimization, SGD's convex assumptions, and KNN's distance-based sensitivity to feature scaling. Error Analysis and Confusion Patterns: To further analyze the model's behavior, we computed the 100-class confusion matrix for the SVM with a polynomial kernel on the test set. Due to space limitations, we report only the most frequent confusion pairs instead of the full 100 × 100 matrix. The classifier most often confuses visually similar species with highly comparable wing colors and banding patterns, such as Species A vs. Species B and Species C vs. Species D, which share nearly indistinguishable visual traits. These pairs account for a small portion of the total errors but highlight that the remaining misclassifications are largely driven by fine-grained interspecies similarities rather than a systematic failure of the model.

To rigorously test generalization beyond Kaggle's curated conditions, we assembled an independent external test set of 100 butterfly images (5 images/species × 20 species) from public repositories: iNaturalist (https://www.inaturalist.org/projects/butterflies-of-indonesia),[24] and BugGuide (https://www.gbif.org/occurrence/gallery?taxon_key=1894342).[25] These images feature real-world challenges (variable lighting, pose, occlusion) absent from the training data. Examples of the testing images appear in Fig. 8. The results are shown in Table 5, which presents comprehensive external validation results.

Table 6 and Table 7 show that the SVM-polynomial kernel outperformed other classifiers on external data, maintaining polynomial kernel leadership consistent with the internal results.

Table 8 compares our hybrid approach against prior butterfly classification studies, acknowledging that most cited works[5–11] utilize smaller, less diverse datasets (150–10,035 images, 6–15 species) with fundamentally different characteristics than our 100-species benchmark. These comparisons primarily illustrate the progression from traditional GLCM + k-NN methods ([5,6]: 91–98% accuracy on limited datasets) to CNN-based approaches ([7–11]: 75–96% accuracy), contextualizing our hybrid methodology within this evolution rather than claiming direct equivalence. Only[12] employs the identical Kaggle Butterfly Species dataset (12,594 training/500 test images, 100 species), providing the most valid comparison point, where our SVM + Polynomial kernel achieves 95% accuracy versus their Xception model's 93.4%—a 1.6% improvement attributable to our preprocessing pipeline and classifier optimization. This direct comparison validates our approach's competitiveness on standardized benchmarks, while the broader literature survey demonstrates scalability advantages over small-dataset studies.

Conclusion

This paper proposes a hybrid approach for butterfly species classification by integrating deep learning-based feature extraction with various machine learning classification techniques. The methodology effectively capitalizes on the advantages of both fields to achieve high classification accuracy on butterfly images. Deep learning, particularly through SqueezeNet, is utilized to extract rich and high-level features that capture intricate visual patterns such as wing texture, shape, and color distribution, eliminating the need for manual feature engineering. These extracted features are subsequently classified using multiple machine learning algorithms, including SVM with various kernel functions, KNN, neural networks, and SGD. Among these, SVM with a polynomial kernel demonstrated the highest accuracy, underscoring its capability in managing linear and high-dimensional data. This research showed that the suggested preprocessing pipeline, especially background removal and white background conversion, results in significantly higher classification accuracy for butterfly species across 100 classes. Experimental performance is good, with an internal validation accuracy of 95% on the benchmark dataset. Moreover, the accuracy of the suggested method has an external testing value of 83.5%, meaning that it can be generalized to unobserved images in the real world, even though the size of the external testing category is limited. The experimentation has proven that there have been regular gains in comparison to the models being trained without undergoing preprocessing, and the optimal results were attained when the entire pipeline was utilized. Even with these improvements, the training data have a class imbalance. Further research will be performed on a larger-scale external validation and more solid evaluation strategies. In addition, future research will include direct end-to-end CNN baselines (MobileNetV2, ResNet50) on identical splits to further validate hybrid superiority, leveraging our preprocessing pipeline with deeper architectures.

Authors' declaration

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are ours. Furthermore, any Figures and images that are not ours have been included with the necessary permission for re-publication, which is attached to the manuscript.
- No animal studies are present in the manuscript.
- No human studies are present in the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee at University of Baghdad.

Authors' contributions statement

M.A.R. collected, analyzed, and interpreted the dataset, while M.A.R., W.H.A., and F.A.A. designed the proposed system, selected the techniques used, analyzed and interpreted the results, and compared them with previous studies. M.A.R., W.H.A., F.A.A., and T.S. wrote the paper with input from all authors, while T.S. revised and proofreading of the manuscript.

References

1. Stork NE. How many species of insects and other terrestrial arthropods are there on earth? Annu Rev Entomol. 2018;63(1):31–45. https://doi.org/10.1146/annurev-ento-020117-043348.
2. Yasmin R, Das A, Rozario LJ, Islam ME. Butterfly detection and classification techniques: A review. Intell Syst Appl. 2023;18(5):1–23. https://doi.org/10.1016/j.iswa.2023.200214.
3. Li F, Zhou W. An improved contour feature extraction method for the image butterfly specimen. In: 3D Imaging Technologies-Multidimensional Signal Processing and Deep Learning: Methods, Algorithms and Applications. Springer. 2021;2:17–26. https://doi.org/10.1007/978-981-16-3180-1_3.

4. Adjé EA, Delmaire G, Ahouandjinou ASRM, Puigt M, Roussel G. Robust approach for butterfly species classification using a single spatio-hyperspectral image. In: 14th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS). IEEE. 2024;1–5. https://doi.org/10.1109/WHISPERS65427.2024.10876422.
5. Xue A, Li F, Xiong Y. Automatic identification of butterfly species based on gray-level co-occurrence matrix features of image block. J Shanghai Jiaotong Univ. (Sci.) 2019;24(2):220–225. https://doi.org/10.1007/s12204-018-2013-y.
6. Andrian R, Maharani D, Muhammad MA, Junaidi A. Butterfly identification using gray level co-occurrence matrix (GLCM) extraction feature and k-nearest neighbor (KNN) classification. Register: J Ilm Teknol Syst Inform. 2020;6(1):11–21. https://doi.org/10.26594/register.v6i1.1602.
7. Liang B, Wu S, Xu K, Hao J. Butterfly detection and classification based on integrated YOLO algorithm. In: Adv Intell Syst Comput. 2020;1107:500–512. https://doi.org/10.1007/978-981-15-3308-2_55.
8. Kartika DSY, Maulana H. Classification of color features in butterflies using the Support Vector Machine (SVM). IJCONSIST J. 2021;2(2):83–87. https://doi.org/10.33005/ijconsist.v2i02.50.
9. Rajeena PPF, Orban R, Vadivel KS, Subramanian M, Muthusamy S, Elminaam DSA, *et al.* A novel method for the classification of butterfly species using pre-trained CNN models. Electron. 2022;11(13):1–20. https://doi.org/10.3390/electronics11132016.
10. Liu F. Butterfly and moth image recognition based on residual neural network. In: Proc. 1st Int Conf Data Anal. Mach Learn. (DAML 2023). 2023;1:211–217. https://doi.org/10.5220/0012798500003885.
11. Jin H, Sha K, Xie X. Fine-grained butterfly classification based on multi-feature enhancement. IEEE Access. 2024;12(29):194–203. https://doi.org/10.1109/ACCESS.2024.3362073.
12. Pradnyatama M, Sari CA, Rachmawanto EH, Islam HMM. A comparative analysis of convolutional neural network (CNN): MobileNetV2 and Xception for butterfly species classification. J Masy Informatika. 2025;16(1):69–90. https://doi.org/10.14710/jmasif.16.1.72957.
13. Rasool M, Ismail NA, Al-Dhaqm A, Yafooz WMS, Alsaeedi A. A novel approach for classifying brain tumours combining a SqueezeNet model with SVM and fine-tuning. Electron. 2023;12(1):1–18. https://doi.org/10.3390/electronics12010149.
14. Zidan RA, Karraz G. Towards an efficient Internet of Things intrusion detection by using Support Vector Machine. Baghdad Sci J. 2025;22(5):1714–1724. https://doi.org/10.21123/bsj.2024.11067.
15. Cicek ZIE, Ozturk ZK. Optimizing the artificial neural network parameters using a biased random key genetic algorithm for time series forecasting. Applied Soft Computing. 2021;102:107091. https://doi.org/10.1016/j.asoc.2021.107091.
16. AlKhafaji BJ, Salih MA, Shnain SA, Rashid OA, Rashid AA, Hussein MT. Applying the artificial neural networks with multiwavelet transform on phoneme recognition. J Phys.: Conf Ser. 2021;1804(1):1–7. https://doi.org/10.1088/1742-6596/1804/1/012040.
17. Solihin F, Syarief M, Rochman EMS, Rachmad A. Comparison of Support Vector Machine (SVM), K-Nearest Neighbor (K-NN), and Stochastic Gradient Descent (SGD) for classifying corn leaf disease based on histogram of oriented gradients (HOG) feature extraction. Elinvo. 2023;8(1):121–129. https://doi.org/10.21831/elinvo.v8i1.55759.
18. Zhang S. Challenges in KNN classification. IEEE Trans Knowl Data Eng. 2021;34(10):4663–4675. https://doi.org/10.1109/TKDE.2021.3049250.
19. PrzybyłK, Gawrysiak-Witulska M, Bielska P, Rusinek R, Gancarz M, Dobrzański Jr B, *et al.* Application of machine learning to assess the quality of food products-case study: Coffee bean. Appl Sci. 2023;13(19):1–12. https://doi.org/10.3390/app131910786.
20. Rajab MA, Abdullatif FA, Sutikno T. Segmentation of aerial images using different clustering techniques. Iraqi J Sci. 2025;66(3):1288–1299. https://doi.org/10.24996/ijs.2025.66.3.25.
21. Rajab MA. Human identification based on SIFT features of hand image. Int J Comput Digit Syst. 2023;14(1):367–375. https://doi.org/10.12785/ijcds/140128.
22. Mashhadani S, Abdulsalam W H, Hassen O A, Darwish S M. Fusion of Type-2 neutrosophic similarity measure in signatures verification systems: A new forensic document analysis paradigm. Intell Autom Soft Comput. 2024;39(5):805–828. https://doi.org/10.32604/iasc.2024.054611.
23. Abdulsalam W H, Ajeena R K, Saad M A. Integrating Feature Selection and Machine Learning Boosting for Accurate Breast Cancer Prediction. In Proceedings of the 13th International Conference on Applied Innovations in IT (ICAIIT). 2025;13(4):91–100. http://dx.doi.org/10.25673/122070.
24. Van Horn G, Mac Aodha O, Song Y, Cui Y, Sun C, Shepard A, *et al.* The inaturalist species classification and detection dataset. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2018:8769–8778. https://doi.org/10.1109/CVPR.2018.00914.
25. Skvarla MJ, Fisher JR. Online community photo-sharing in entomology: A large-scale review with suggestions on best practices. Ann Entomol Soc Am. 2023;116(5):276–304. https://doi.org/10.1093/aesa/saad021.

التعلم الآلي العميق - الهجين لتصنيف صور أنواع الفراشات

مها عبد رجب[1]، وصال هاشم عبد السلام[2]، فراس عبد الحميد عبد اللطيف[2]، تول سوتيكنو[3]

[1] قسم علوم الحياة، كلية التربية للعلوم الصرفة/ابن الهيثم، جامعة بغداد، بغداد، العراق
[2] قسم الحاسوب، كلية التربية للعلوم الصرفة/ابن الهيثم، جامعة بغداد، بغداد، العراق
[3]قسم الهندسة الكهربائية، كلية التكنولوجيا الصناعية، جامعة أحمد دحلان، يوجياكارتا، إندونيسيا

الخلاصة

يُعدّ تصنيف أنواع الفراشات أمرًا بالغ الأهمية في دراسات التنوع الحيوي ومراقبة البيئة. إذ أن التصنيف اليدوي غالبًا ما يكون عملية شاقة تتطلب خبرة متخصصة، كما أنه عرضة للأخطاء، لا سيما عند تشابه الخصائص البصرية بين الأنواع المختلفة. ولمعالجة هذه التحديات، يقدم هذا البحث نهجًا هجينًا يجمع بين التعلم الآلي والتعلم العميق لاستخراج السمات. ولتعزيز وضوح السمات المهمة، تم تطبيق تقنيات المعالجة المسبقة، مثل إزالة الخلفية والتحويل إلى الصورة الثنائية، على صور الفراشات. وقد تم تنفيذ عملية استخراج السمات باستخدام شبكة SqueezeNet الالتفافية المدربة مسبقًا على مجموعة بيانات ImageNet. ومن خلال حذف طبقة التصنيف النهائية، تمكنت الشبكة من توليد متجهات سمات مميزة تعكس الخصائص البصرية لكل فراشة بكفاءة. بعد ذلك، استُخدمت هذه المتجهات في تدريب مجموعة من مصنفات التعلم الآلي، بما في ذلك آلات متجهات الدعم (SVM) باستخدام نوى مختلفة، وخوارزمية أقرب الجيرا (KNN)، والشبكات العصبية (NN)، وخوارزمية الانحدار التدرجي العشوائي (SGD).. اعتمدت الدراسة على مجموعة بيانات مخصصة للتعرف على أنواع الفراشات والعث، وتضم 100 فئة تصنيفية مختلفة. وأظهرت النتائج أن نموذج SVM باستخدام النواة متعددة الحدود حقق أفضل دقة تصنيف وصلت إلى 95%، تليه الشبكات العصبية. في المقابل، سجلت خوارزميتا KNN وSGD دقة أقل نسبيًا. أما عند اختبار النموذج على صور خارج مجموعة البيانات، فقد بلغت الدقة 83.5%.. وبناءً على ذلك، تؤكد النتائج فاعلية استخدام تقنيات التعلم العميق في استخراج السمات عند دمجها مع مصنفات التعلم الآلي، مما يسهم في تحسين أداء أنظمة تصنيف الصور.

الكلمات المفتاحية: تصنيف الفراشات، التعلم العميق، التعلم الآلي، الشبكات العصبية، آلة المتجهات الداعمة، شبكة الضغط.