

Article

Advancing Taxonomy with Machine Learning: A Hybrid Ensemble for Species and Genus Classification

Loris Nanni ^{*}, Matteo De Gobbi [†], Roger De Almeida Matos Junior [†] and Daniel Fusaro

Department of Information Engineering, University of Padova, Via Giovanni Gradenigo, 6b, 35131 Padova, Italy; matteo.degobbi@studenti.unipd.it (M.D.G.); roger.dealmeidamatosjunior@studenti.unipd.it (R.D.A.M.J.); fusarodani@dei.unipd.it (D.F.)

^{*} Correspondence: loris.nanni@unipd.it

[†] These authors contributed equally to this work.

Abstract: Traditionally, classifying species has required taxonomic experts to carefully examine unique physical characteristics, a time-intensive and complex process. Machine learning offers a promising alternative by utilizing computational power to detect subtle distinctions more quickly and accurately. This technology can classify both known (described) and unknown (undescribed) species, assigning known samples to specific species and grouping unknown ones at the genus level—an improvement over the common practice of labeling unknown species as outliers. In this paper, we propose a novel ensemble approach that integrates neural networks with support vector machines (SVM). Each animal is represented by an image and its DNA barcode. Our research investigates the transformation of one-dimensional vector data into two-dimensional three-channel matrices using discrete wavelet transform (DWT), enabling the application of convolutional neural networks (CNNs) that have been pre-trained on large image datasets. Our method significantly outperforms existing approaches, as demonstrated on several datasets containing animal images and DNA barcodes. By enabling the classification of both described and undescribed species, this research represents a major step forward in global biodiversity monitoring.



Academic Editor: Firstname Lastname

Received: 31 December 2024

Revised: 5 February 2025

Accepted: 9 February 2025

Published:

Citation: Nanni, L.; De Gobbi, M.; De Almeida Matos Junior, R.; Fusaro, D. Advancing Taxonomy with Machine Learning: A Hybrid Ensemble for Species and Genus Classification. *Algorithms* **2025**, *1*, 0. <https://doi.org/>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: ensemble; convolutional neural networks; support vector machine; discrete wavelet; DNA barcode.

1. Introduction

Exploring biodiversity involves a complex and demanding process. It begins with extensive fieldwork, where entomologists venture into diverse and often remote habitats to gather specimens. These are subjected to rigorous identification procedures, including morphological assessments, genetic analyses, and taxonomic evaluations. This meticulous work is necessary for deepening our understanding of animal diversity, their ecological roles, evolutionary links, and interactions with human endeavors. For instance, it is important to highlight that insects play a critical role in ecosystems, such as pollination, decomposition, and serving as a food source for other organisms. This underscores the urgency of cataloging and preserving their diversity.

In this paper, we mainly focus on insect datasets. Although an estimated 5.5 million insect species exist, only around 20% have been documented [1]. The challenge of documentation is intensified by the extinction of numerous species before they can be formally described [2].

In biology, taxonomy refers to the branch of science concerned with the conception, naming, and classification of groups of organisms. Modern classification divides any organism based on the Domain (Archaea, Bacteria, and Eucarya) and then on Kingdom, Phylum, Class, Order, Family, Genus, and Species. An example can be seen in Figure 1. Traditionally, taxonomists rely on morphological features to classify insects by their physical characteristics [3]. However, these keys are less effective for undescribed species with indistinct or missing diagnostic features. To address this limitation, DNA barcoding [4] serves as a complementary method, using genetic variation to identify species when phenotypic traits are insufficient [5].

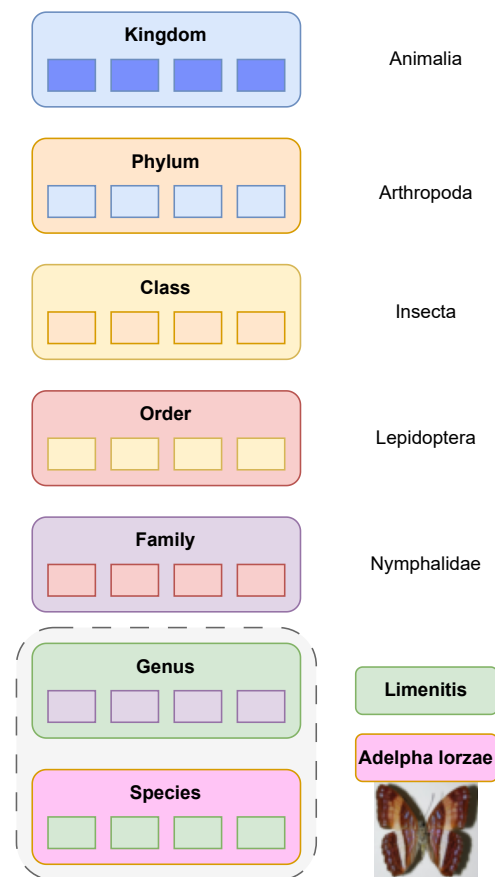


Figure 1. Example of modern taxonomic categorization for *Adelpha lorzae*.

Identifying insect species remains a major challenge. Although the DNA Barcode Database (BOLD) [6] holds a large amount of genetic data, most of it is not related to known species. This mismatch reveals the slow progress in identification, made worse by a shortage of taxonomists and a decline in traditional taxonomy practices, as explained in [7]. To address this, there is an urgent need for faster and more efficient methods to uncover and classify species.

In this study, we tackle critical challenges in machine learning for insect classification, focusing on the issue of incomplete species representation. We propose an ensemble model designed to classify both known and unknown species. This model integrates traditional support vector machines (SVMs) with deep learning by converting conventional feature patterns into two-dimensional representations through vector-to-matrix reshaping, forming a three-channel input. The proposed approach combines DNA barcoding data with image-derived features to train SVM and the neural nets.

The motivation for adopting this ensemble approach lies in leveraging the distinct strengths of both neural networks and SVM, making the system particularly effective for

tasks that require diverse decision boundaries and robust performance against overfitting. Neural networks and SVMs utilize fundamentally different learning techniques, which often lead to different errors on different sets of samples. Neural networks reduce error through backpropagation across multiple layers, whereas SVMs focus on maximizing the margin between classes. By combining these complementary prediction strategies in an ensemble, the model benefits from the diverse decision boundaries created by each algorithm, ultimately improving overall performance.

The main contributions of this paper are as follows:

- The development of an ensemble classifier that outperforms traditional SVMs and previous state-of-the-art (SOTA) methods;
- The introduction of a novel technique to represent an image as a feature vector and a method, based on discrete 1D wavelet transforms, for representing feature vectors as images;
- Two new datasets have been collected and made available to the community [8];
- The provision of all resources and source code as open-source tools for researchers.

We are aware that there are a lot of methods proposed in the literature to represent an image as a vector, from hand-crafted to learned approaches, but the goal of this paper is not to propose a method to describe an image as a vector and compare it with all current SOTA methods. Our goal is to propose a complete image+DNA barcoding system for a given problem. For this reason, we make use of three datasets already tested in the related literature that allow us to compare our method with the current SOTA. Our approach uses the same hyperparameters in all datasets and obtains a new SOTA. Therefore, we argue that the proposed method can be useful to the community.

We are aware that there are many other methods to classify a vector; for example, we could use a standard multi-layer perceptron network or a boosting method, and so on. The goal of this work is also to see if a very different method than the standard ones, obtained by using DWT to transform the vector into a matrix and then to perform the tuning of a pre-trained ResNet50 on ImageNet, allows extracting information from the data different from that obtained by an SVM. We emphasize that we are using libSVM, by far the most widely used SVM tool, and SVM is still the most widely used classifier, so the fact that the proposed ensemble performs better than SVM is an interesting result for the machine learning and deep learning community.

The paper is structured as follows. Section 2 outlines the literature review. Section 3 provides an overview of the dataset utilized in this study and presents the proposed approach. Section 4 outlines the experimental setup. Section 5 discusses the key findings and challenges encountered.

2. Related Works

Machine learning (ML) techniques present innovative solutions by analyzing complex data patterns to classify species and detect anomalies. Traditional ML approaches have shown promise in recognizing subtle morphological traits in images, including those of undescribed species, as shown in [9]. Although not yet as accurate as DNA-based methods, recent advancements indicate that image-based ML is approaching expert-level performance in entomological studies [10–15]. However, these models are constrained by incomplete training datasets, which are particularly problematic when dealing with rare or undescribed species and the morphological variations that occur in different stages of insect life [16,17].

In recent years, many machine/deep learning approaches have been proposed for DNA barcoding classification [18–20]. In [21], the study examines two primary approaches to taxonomic classification: database-based methods and machine learning techniques.

Database methods generally achieve higher accuracy when extensive reference data are available, whereas machine learning approaches perform better with limited datasets but tend to be less precise overall. The study also demonstrates that integrating multiple database-based methods enhances classification accuracy, offering valuable insights for computational biology. In [22], the authors introduce BayesANT, a Bayesian nonparametric taxonomic classifier designed to determine the taxonomic affiliation of DNA sequences, even for organisms without reference sequences or previously unidentified taxa. BayesANT utilizes species sampling model priors to detect unobserved taxa across different taxonomic levels, enabling flexible and probabilistic predictions. The algorithm was evaluated using Finnish arthropod data and exhibited high accuracy, particularly in identifying taxa absent from the training dataset. In [23], a novel deep learning approach is proposed, integrating Elastic Net-Stacked Autoencoder (EN-SAE) with Kernel Density Estimation (KDE), referred to as the ESK model. An experimental validation on three datasets confirms the effectiveness and superiority of ESK, demonstrating its capability to accurately classify fish from different families based on DNA barcode sequences. For a recent comparison of standard machine learning algorithms applied to species family classification using DNA barcodes, see [24].

Deep learning, a subset of machine learning, has been utilized in various entomological fields, including pest detection, analyzing plant–insect interactions [25,26,34], environmental DNA (eDNA) [27] or invertebrates image classification [28–30]. However, these applications are often tailored to specific insect groups, limiting their broader applicability, as in [31]. A critical challenge for ML-based insect identification lies in addressing both described and undescribed species. Many current models operate under the assumption of a fully represented dataset, which is rarely the case. Additionally, these methods face significant difficulties in managing the vast number of insect species and distinguishing outliers within the highly diverse Insecta class (see [16]). Other recent approaches are as follows: [32], where the application of DNA barcoding data is proposed in image-based out-of-distribution detection in fine-grained taxa identification; and [33], where a novel framework is proposed to combine computer vision and bulk DNA metabarcoding specimen processing pipelines to improve the accuracy and taxonomic granularity of individual specimen classifications.

3. Materials and Methods

Classifying species is a critical task in biodiversity monitoring; this process is often resource-intensive and prone to human error, particularly when dealing with unknown or undescribed species. The challenge lies in developing an automated system capable of accurately classifying both described and undescribed species while addressing the limitations of existing approaches, which often fail to effectively utilize multimodal data such as images and DNA barcodes. The primary goal of this research is to design an efficient and accurate machine learning-based system for classifying species and grouping undescribed species at the genus level. This system aims to support global biodiversity monitoring by overcoming the limitations of manual taxonomy and existing computational approaches. The specific goals of this work are as follows:

- **Data Representation:** Transform one-dimensional DNA barcode data into a format compatible with state-of-the-art image-based neural networks to leverage pre-trained models effectively.
- **Integration of Modalities:** Develop a hybrid machine learning approach that integrates image and DNA barcode data to improve classification accuracy.

- **Performance Validation:** Benchmark the proposed method against existing approaches on publicly available datasets to demonstrate its superiority in terms of accuracy and generalizability.

The following methods, techniques, and tools are used:

- **Data Preprocessing:** Images and DNA are described as feature vectors and then the conversion of feature vectors into two-dimensional three-channel matrices, using the Discrete Wavelet Transform, step is performed.
- **Model Architecture:** The application of pre-trained CNN architectures and SVM to process data by leveraging CNN-extracted features.
- **Ensemble Approach:** Combining predictions from CNN and SVM to improve classification accuracy through score fusion strategies.
- **Implementation Tools:** MATLAB has been used for data transformation, preprocessing, and classification; PyTorch has been employed for implementing, training, and fine-tuning CNN models used in the feature extraction step.
- **Datasets:** Multiple publicly available datasets containing animal images and DNA barcodes were used to validate the approach.

In this section, we will first explain the datasets used and proposed in this paper, then the methods to extract features from an image and the related DNA barcoding; finally, the methods to classify a given vector will be explained. As previously explained, we will use SVM and neural networks as classifiers, where neural networks are trained by rearranging the vector as a three-channel matrix. We assess the performance using five datasets:

- The one proposed in [16], named the Badirli dataset in the rest of the paper (<https://dataworks.indianapolis.iu.edu/handle/11243/41>, accessed March 28, 2025);
- Two new datasets that were proposed here (<https://zenodo.org/records/14277812>, accessed March 28, 2025), detailed in Sections 3.1 and 3.2;
- The Beetle and the Fish datasets (<https://zenodo.org/records/14728702>, accessed March 28, 2025), proposed in [34].

3.1. Dataset with Simulated Undescribed Species

In accordance with the methodology outlined in the original paper, the data utilized in our experiments were obtained from the Barcode of Life Data System (BOLD) [6], which is a cloud-based data storage and analysis platform developed at the Centre for Biodiversity Genomics in Canada. The data consist of 32,424 image samples, e.g., see Figure 2, of insect species from four Insecta orders, *Diptera*, *Coleoptera*, *Lepidoptera*, and *Hymenoptera*, each associated with a DNA barcode COI mitochondrial sequence of that species.

Due to the fact that we did not have access to the original images used in [16], we resorted to downloading the data from the BOLD Systems platform to recreate a dataset that closely matches [16]. However, we encountered some discrepancies: some species names have been updated, and certain species have been split into two distinct categories. As a result, our dataset exhibits some differences compared to [16]. Table 1 highlights these differences.

Table 1. Differences in the collected dataset compared to [16], the columns Genera and Species report the number of classes of the samples.

	Genera	Species	Samples
[16]	368	1040	32,848
Here	371	1050	32,424

Next, we split the data following the methodology described in [16]. We considered all genera containing three or more species and we randomly selected 30% of the species within each of these genera as “undescribed” (only the genus is known while the species is unknown) and added all the samples from these species in the test set. We then split the remaining described data into 80% for the training+validation and 20% for the test set. Hence, the final test set consists of this 20% together with the previously designated undescribed species. This process is described by Algorithm 1. After applying Algorithm 1 to obtain a training+validation set and a test set, we apply it again on the training+validation set to obtain a training set with only described species and a validation set with both described and undescribed species. It is necessary that the undescribed species in the validation set are different species from the undescribed species in the test set; this is important to avoid class leaking from the test set to the validation set, ensuring that the undescribed species in the test set remain unseen during validation.

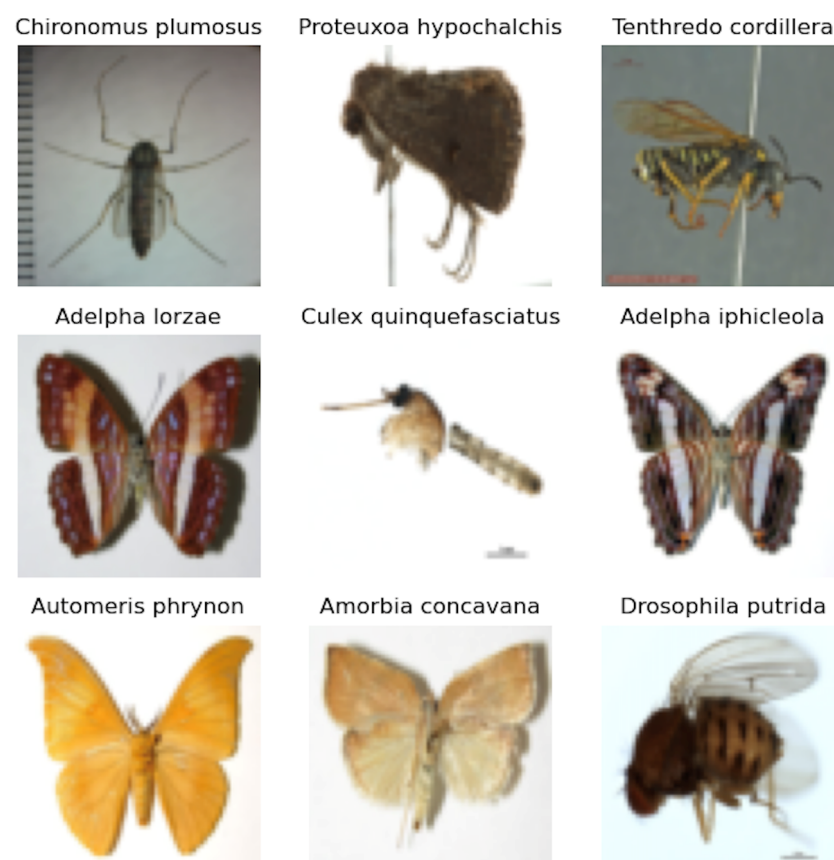


Figure 2. Dataset example.

We did not use the validation set for the training or the hyperparameter tuning of our models described in Section 3.4, but we provide this validation set to be used in future works that use the same dataset in order to be able to compare our results.

Algorithm 1 Split dataset to simulate undescribed species

Require: D ▷ Dataset to split, containing species and their genus
Ensure: $training, test$ ▷ Split dataset with undescribed species

- 1: Initialize $test \leftarrow \emptyset$
- 2: Group species in D by genus
- 3: **for all** genus g in D where $|g| \geq 3$ **do**
- 4: Randomly select 30% of species in g as *undescribed*
- 5: Add all samples in D with *undescribed* species to $test_unseen$
- 6: Remove samples with *undescribed* species from D
- 7: **end for**
- 8: Split the remaining D into $training$ (80%) and $test_seen$ (20%)
- 9: $test = test_seen \cup test_unseen$
- 10: **return** $training, test$

3.2. Dataset with Undescribed Species

To create a dataset of real-life undescribed species, we began by listing all genera present in the BOLD dataset containing simulated undescribed species. Using this list, we queried the BOLD Systems database to download samples belonging to these genera but with no species name, indicating that they were undescribed at the time of data retrieval. From these downloaded samples, we randomly selected 5% to ensure the dataset was comparable in size to the one containing simulated undescribed species. As a result, the final dataset contains 40,050 samples (image and COI mitochondrial DNA sequence) representing real-life undescribed species. This dataset is used entirely as a test set, while training is carried out using the dataset described in Section 3.1.

3.3. Beetle and Fish Datasets

Two other datasets with global coverage for specific groups have been used. Both of them, namely the Beetle and Fish datasets, have been proposed in [34], requiring a minimum of five images per species. The Beetle dataset comprises 615 mitochondrial COI fragments from 123 beetle species across three families: Coccinellidae, Cantharidae, and Anthribidae. The Fish dataset includes 1070 mitochondrial COI fragments representing 214 fish species from 15 families. These groups served as excellent test cases to evaluate the algorithm power and robustness; just to show the generalizability of the method, we also used a dataset of fishes that is not related to insects.

3.4. Feature Extraction

All the models presented in this section have been trained only on the training set of the dataset with simulated undescribed species presented in Section 3.1, without any hyperparameter tuning using the validation set. After the training, the weights of the models have been saved and used to extract the features from the other datasets.

We reproduced the methodology detailed in [16] using our data. To reproduce their DNA feature extraction technique, we used the same Convolutional Neural Network (CNN) architecture but we changed the activation function of the fully-connected layer from Tanh to LeakyReLU because we experienced vanishing gradient. For the image features, we used a pre-trained Resnet101 that gave us a vector of 2048 features. The Resnet was not fine-tuned, as described in [16].

Our method also used a CNN to extract the DNA features. Our CNN architecture consists of 2 convolutional layers, both of which use a one-dimensional (5×1) kernel (in order to avoid reducing too much the output dimension), batch normalization, and LeakyReLU as the activation function. A dropout layer is used (70% dropout rate) after the second convolutional layer. The vectors are then flattened and projected on a linear layer of size 1500, followed by another dropout (70% dropout rate) and a LeakyReLU as

mentioned before. The output is finally projected on a linear layer of size equal to the number of classes.

The main idea behind using the 5×1 convolution instead of 3×3 as in the original paper was to maintain the shape of the second dimension of the tensor constant without using padding. This allows the CNN to focus on finding local patterns in neighboring nucleotides (Figure 3). Using 3×3 kernels would mean also considering convolutions across the one-hot encoding of a single nucleotide, possibly losing positional information and finding irrelevant relations.

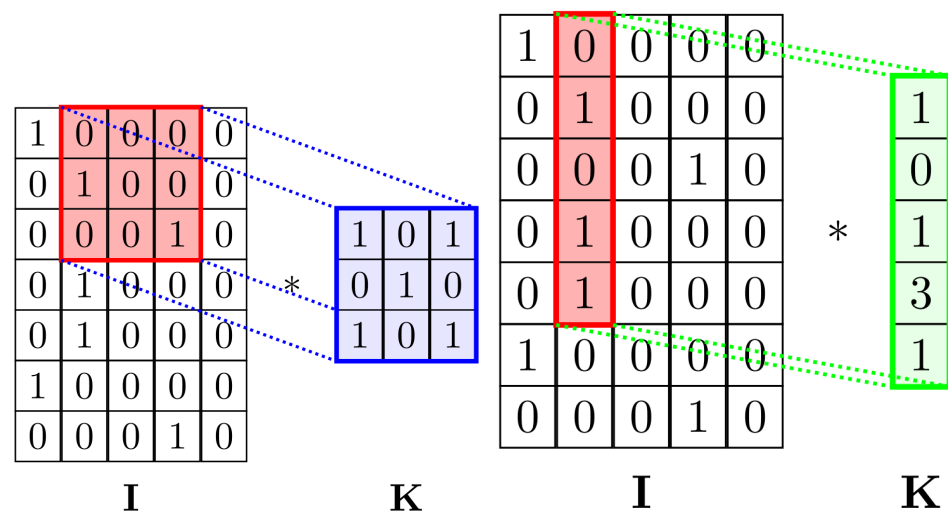


Figure 3. Comparison of 3×3 convolution with 5×1 convolution. I represents the input of the convolution, while K is the filter. The symbol $*$ denotes the convolution operator. In each I matrix, the red region highlights the current area where the filter K (shown in blue or green) is applied.



Figure 4. Images generated by the ReACGAN.

Furthermore, for extracting the image features, we used the intermediate layer of the discriminator of a conditional Generative Adversarial Network (GAN) model, named Rebooted Auxiliary Classifier Generative Adversarial Network (ReACGAN). ReACGAN [35] is a newer version of the ACGAN model (a model of conditional GAN) with the purpose of improving the stability of the training by using residual connections (similar to ResNet), spectral normalization, embedding normalization, conditional batch normalization, and a different loss. ReACGAN aims to solve the exploding gradient and mode collapse problems that occur in ACGAN when the dataset contains a high number of classes. Since we were experiencing mode collapse with regular ACGANs due to having a large number of classes, we decided to use this improved version. The residual connections are the same as in ResNet. Some results of image generation are shown in Figure 4.

Spectral normalization, introduced in [36] to stabilize the training of the discriminator, is applied to both the layers of the generator and the discriminator in the ReACGAN. It works on the weight matrix W applying (exp. 1), where h is a randomly initialized vector. This is equivalent to dividing the matrix by its maximum singular value.

$$\max_{h:h \neq 0} \frac{\|Wh\|_2}{\|h\|_2} \quad (1)$$

Conditional batch normalization (exp. 3) differs from regular batch normalization (exp. 2) by determining the values of the parameters β and γ using a linear layer from the input features instead of learning one value for them. In conditional GANs it allows the model to learn different scaling factors for different classes of samples (e.g., different species or genera).

$$\gamma\left(\frac{x - \mu}{\sigma}\right) + \beta \quad (2)$$

$$\text{MLP}_\gamma(x)\left(\frac{x - \mu}{\sigma}\right) + \text{MLP}_\beta(x) \quad (3)$$

where μ and σ are the mean and the standard deviation of the values of elements of the batch. It has been proved, by [35], that in ACGAN discriminators the gradients scale with the norm of the sample embedding (the feature extracted by the discriminator).

Finally, the D2D-CE (data-to-data cross-entropy) loss is used. Normally, ACGANs compute the cross-entropy between the feature extracted by the discriminator (called sample embedding) and the embedding of the class label which is called proxy (in ACGANs, a one-hot encoding of the label can be used instead of an embedding of the class label).

In [35] it is showed how normalizing the sample embedding and the label proxy avoids the exploding gradient problem that appears at early training and is one of the causes of early mode collapse.

Equations (4) and (5) describe the common cross-entropy and the D2D-CE loss functions, respectively. Both are expressed by considering $F(\mathbf{x})$, the feature embedding vector extracted from image x by the penultimate layer of the discriminator, and $W = [\mathbf{w}_1, \dots, \mathbf{w}_c]$, which is the weight matrix of the last layer of the discriminator.

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N} \sum_{i=1}^N \log \left(\frac{\exp(F(\mathbf{x}_i)^\top \mathbf{w}_{y_i})}{\sum_{j=1}^c \exp(F(\mathbf{x}_i)^\top \mathbf{w}_j)} \right). \quad (4)$$

$$\mathcal{L}_{\text{D2D-CE}} = -\frac{1}{N} \sum_{i=1}^N \log \left(\frac{\exp([\mathbf{f}_i^\top \mathbf{v}_{y_i} - m_p]_- / \tau)}{\exp([\mathbf{f}_i^\top \mathbf{v}_{y_i} - m_p]_- / \tau) + \sum_{j \in \mathcal{N}(i)} \exp([\mathbf{f}_i^\top \mathbf{f}_j - m_n]_+ / \tau)} \right), \quad (5)$$

The D2D-CE (Equation (5)) is a modified version of CE (Equation (4)), where $\mathbf{v}_{y_i} = \frac{\mathbf{w}_{y_i}}{\|\mathbf{w}_{y_i}\|}$ and $\mathbf{f}_i = \frac{P(F(\mathbf{x}_i))}{\|P(F(\mathbf{x}_i))\|}$, P is a projection carried out by a linear layer, $[\cdot]_- = \min(\cdot, 0)$ $[\cdot]_+ = \max(\cdot, 0)$.

$= \max(\cdot, 0)$, $\mathcal{N}(i)$ is the set of indices of the samples in the minibatch for which the label is different from y_i (it is the real label); the margins m_p and m_n and the temperature τ are hyperparameters, and we use the implementation and the values of the hyperparameters suggested in [37].

Also in D2D-CE, the denominator (Equation (5)) of the softmax still computes the similarity between the sample embedding and the proxy (either one-hot or embedding) in order to consider data-to-class similarities, but in the denominator, we split the summation in two: a term equal to the numerator plus a term that computes the similarities between the sample embeddings for images of the batch belonging to different classes ($j \in \mathcal{N}(i)$). This second term does not consider data-to-class relationships because it does not involve the weights of the last layer \mathbf{v}_{y_i} . Conversely, it considers relationships between the sample embeddings of different classes. For this reason, the loss function is called Data-to-Data CE.

This makes it so that by minimizing the loss, we make the sample embeddings more similar to the corresponding class proxies, but at the same time, we make the sample embeddings of images belonging to different classes different between each other. This idea is similar to the contrastive loss used in siamese networks.

The intuition behind this loss is that we make the discriminator use visual features from the images to distinguish images of different classes instead of just making it only guess the class directly. If this intuition is correct, it would be useful for our purpose since our objective is not just to generate realistic images but to also obtain useful features that encode the class of the insect.

The last step to obtain the final version of D2DCE (Equation (5)) is to consider the 3 hyperparameters: the margins m_p and m_n and the temperature τ . Since the model is too big for our dataset, we pre-trained it on a dataset of arbitrary animals taken from various internet datasets for 25 epochs. Then, we fine-tuned it on our dataset for 12 epochs.

The dataset for the pre-training of the ReACGAN contains [38] and some other datasets from kaggle with pictures of insects and other animals. The whole pre-training dataset can be found at [39].

3.5. Discrete Wavelet Transform

In numerical analysis and functional analysis, a discrete wavelet transform (DWT) is a type of wavelet transform where the wavelets are discretely sampled. A significant advantage of DWT compared to Fourier transforms is its ability to provide temporal resolution, capturing both frequency and time location information [40].

Given a 1D discrete signal $x[n]$, the DWT is calculated by passing the signal through two filters:

- A low-pass filter $h[n]$ (scaling function);
- A high-pass filter $g[n]$ (wavelet function).

The outputs of these filters are downsampled by a factor of 2 to reduce the number of coefficients, effectively halving the time resolution. The signal decomposition can be expressed as follows:

$$cA[n] = \sum_k x[k]h[n-k]$$

$$cD[n] = \sum_k x[k]g[n-k]$$

where $cA[n]$ comprises the approximation coefficients (low-frequency components) and $cD[n]$ comprises the detail coefficients (high-frequency components).

The filtering and downsampling process is recursive. After each decomposition, the approximation coefficients are further decomposed into new approximation and detail coefficients at the next level.

This decomposition process is visualized as a binary tree, see Figure 5, where each node represents a sub-space with distinct time-frequency localization. This structure is commonly referred to as a filter bank.

The proposed approach utilizes the following mother wavelets:

- Haar;
- Daubechies;
- Symlets;
- Coiflets;
- Biorthogonal;
- Reverse Biorthogonal;
- Discrete Meyer;
- Fejér–Korovkin Orthogonal;
- Beylkin Orthogonal;
- Vaidyanathan Orthogonal.

We did not perform a study to overfit which wavelets to use, instead using the ones available in MATLAB and using the default parameters; the same set is used for all the datasets, so we assume that there is no risk of overfitting. Below is the pseudocode for the described approach; see pseudocode Algorithm 2. This process is applied three times to create the 3 channels of the matrix used for feeding ResNet50.

Algorithm 2 DWT approach for reshaping data

Ensure: Define *vector* the feature vector that describes a given pattern

Initialize a square matrix *Mat* of size $\lceil \frac{\text{length}(\text{vector})}{8} \rceil$ filled with zeros.

Define $\text{num_levels} = \log_2(\text{length}(\text{vector}))$

$\text{row} \leftarrow 1$

$\text{OrigVector} = \text{vector};$

▷ Original feature vector

for $\text{diffWavelet} = 1:\text{inf}$ **do**

▷ Iterate over wavelet types;

$\text{vector} = \text{OrigVector};$

▷ Fill the matrix *Mat* with wavelet coefficients

for $\text{filter} = 1:(\text{num_levels} - 4)$ **do** ▷ Discard the last 4 levels due to low dimensionality

$[\text{approximation}, \text{detail}] = \text{apply wavelet to 'vector'};$ ▷ Randomly choose mother wavelet for this iteration, extract approximation and detail coefficients. Such choice is random for each network, obviously for each pattern in a given network the same set of mother wavelets is used.

$\text{vector} = \text{approximation};$

▷ Use approximation vector for next iteration

- Check($\text{filter}==1, \text{approximation}, \text{detail}$) At the first filter bank level, approximation and detail coefficients are resized to 25% of their size.

- Check($\text{filter}==2, \text{approximation}, \text{detail}$) At the second filter bank level, coefficients are resized to 50%. ▷ the rationale is to avoid reducing the dimensionality of the other levels, important to underline that the output matrix will be resized to the size required by ResNet50 (i.e., square matrix of size 224 with 3 channels)

$\text{Mat}(\text{row}, :) = \text{approximation};$

$\text{Mat}(\text{row}+1, :) = \text{detail};$

$\text{row} = \text{row} + 2;$

end for

if $\text{row} > \text{size}(\text{Mat}, 2)$ **then** ▷ Exit condition: if row is higher than the number of rows of *Mat*

break;

end if

end for

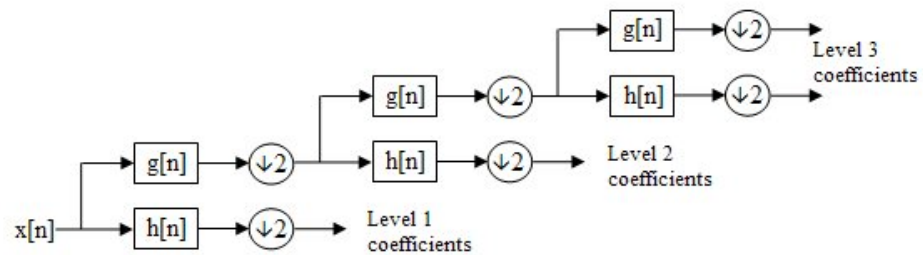


Figure 5. A multi-level decomposition of a signal $x[n]$ using the Discrete Wavelet Transform (DWT). The signal is iteratively passed through a low-pass filter $h[n]$ and a high-pass filter $g[n]$, followed by downsampling ($\downarrow 2$) at each level. The resulting coefficients at different levels represent progressively lower-resolution approximations and detailed components of the original signal.

3.6. Classification Approaches: Support Vector Machine and ResNet50

One of the most influential approaches to supervised learning is the support vector machine [41]. This method is parameterized by a set of N weights $\mathbf{w} \in \mathbb{R}^N$ and a bias term $b \in \mathbb{R}$. In a binary classification task, the SVM predicts a class $y \in \{-1, 1\}$ for a sample vector $\mathbf{x} \in \mathbb{R}^N$ using the following decision function:

$$y = \text{sgn}(\mathbf{w}^\top \mathbf{x} + b), \quad (6)$$

which defines a hyperplane in \mathbb{R}^N referred to as margin. The margin, or the distance between the hyperplane and the closest points from each class, is maximized during training to achieve optimal separation. For multi-class classification, the problem becomes more complex and can be approached in various ways. A common method is the one-vs-one strategy, which divides the task into multiple binary classification problems, one for each pair of classes. The final prediction is computed by majority voting, often incorporating distance from the margin as a tiebreaker. However, this approach requires training an SVM for every class pair, which can significantly increase computational costs. Another strategy is the one-vs-all, which requires to train a model for each unique class in order to distinguish it from all the other classes. The final prediction is computed by selecting the class for which the model predicts the highest margin. Compared to the one-vs-one strategy, the one-vs-all is more robust to an imbalanced dataset and is particularly fast to train, especially when the number of classes is large. Here, we used the one-vs-all approach and the LibSVM toolbox (<https://www.csie.ntu.edu.tw/~cjlin/libsvm/>, accessed March 28, 2025).

ResNet (Residual Network), introduced by He [42], is a deep learning architecture designed to address the vanishing gradient problem in training deep neural networks. It introduces residual connections, or skip connections, that allow gradients to flow directly through the network, bypassing one or more layers. This is achieved by reformulating the layers to learn a residual mapping $F(\mathbf{x}) = H(\mathbf{x}) - \mathbf{x}$, where $H(\mathbf{x})$ is the original mapping, and the output is $F(\mathbf{x}) + \mathbf{x}$. ResNet is highly effective for multi-class classification tasks. The network consists of stacked residual blocks, each comprising convolutional layers, batch normalization, and ReLU activations, with a skip connection that adds the input of the block to its output. The architecture scales to hundreds or thousands of layers while maintaining high performance. ResNet models are often initialized with weights pre-trained on the dataset ImageNet [43], leveraging features learned from over a million diverse images, a technique also known as transfer learning. This approach accelerates convergence, improves performance on downstream tasks, and is computationally efficient compared to training a deep network from scratch. Each net is trained for 10 epochs, with

a batch size equal to 30, a learning rate 0.001, and stochastic gradient descent (SGD) for optimization.

4. Results

In this section, the different approaches are compared using the five datasets described in Section 3. The classification performance for the Badirli dataset (performance is reported in Table 2) and the two datasets proposed here, see Section 3.1 (performance is reported in Table 3) and Section 3.2 (performance is reported in Table 4), was assessed by the weighted species accuracy and the weighted genus accuracy:

$$\text{Weighted Accuracy} = \frac{1}{n} \sum_{j=1}^n \frac{y_j}{n_j}, \quad (7)$$

where for class j , y_j is the number of correctly classified patterns of class j , n_j is the total number of patterns for that class, and n is the number of species or the number of genera. Notice that this is the same formula for the weighted species accuracy and the weighted genus accuracy.

Instead, for the Beetle and Fish datasets, the standard accuracy is used as a performance indicator, as in the related literature; the performance is reported in Tables 5 and 6.

The following methods are reported in this section:

- Bad, the method detailed in [16];
- Bad_D , the method, based only on DNA barcoding, detailed in [16];
- SVM_D , SVM trained using the features proposed in [16] to represent the DNA sequence;
- SVM_O , SVM trained using the features proposed in [16] to represent both the DNA sequence and image;
- SVM_N , SVM trained using the features proposed in this paper to represent both DNA sequence and image;
- SVM_f , sum rule between SVM_O and SVM_N ;
- DWT_O , an ensemble of 15 ResNet50 trained using the DWT approach coupled with the features proposed in [16] to represent both the DNA sequence and image;
- DWT_N , an ensemble of 15 ResNet50 trained using the DWT approach coupled with the features proposed in this paper to represent both the DNA sequence and image;
- DWT_f , sum rule between DWT_O and DWT_N ;
- $\text{DWT}_O + \text{SVM}_O$, sum rule between DWT_O and SVM_O ;
- $\text{DWT}_f + \text{SVM}_f$, sum rule between DWT_f and SVM_f ;
- eDNA, the method proposed in [44] for DNA barcoding classification;
- Proposed, the weighted sum rule among $0.5 \times \text{DWT}_f + 1 \times \text{SVM}_f + 1 \times \text{eDNA}$.

Not all methods are reported for all datasets; for example, in the case of the Badirli dataset, since we do not have any images available, we cannot calculate the new features detailed in Section 3.4. In both the Fish and Beetle datasets, the performance of SVM with original features is low; to reduce computation time, we did not compute DWT_O and, therefore, the method named “Proposed” for the Beetle and Fish dataset is given by the following: $0.5 \times \text{DWT}_N + 1 \times \text{SVM}_N + 1 \times \text{eDNA}$.

Notice that, before the sum rule, the scores of each approach are normalized to mean 0 and standard deviation 1.

In the tests reported in the following Tables 2 and 3, we suppose that an oracle divides the animals between those with known species and those only with known genera. In the final test, reported in Figure 6, we adopt a robust protocol in which all animals are classified at the species level, using a rejection threshold; those not classified in any species

are classified at the genus level: this test is similar to a real application of this kind of problem.

While accuracy provides an overall measure of the model's performance, it does not fully capture the nuances of classification performance, especially for imbalanced datasets. Therefore, we will extend the evaluation metrics to include Precision and Recall for comparing our proposed ensemble with the baseline SVM. Precision (PR) and Recall (RE) are defined as follows:

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

where TP = true positives; FP = false positives; and FN = false negatives. In Table 7, we report PR and RE for comparing our proposed ensemble with SVM_f , when it is available, or SMV_O . This test provides further confirmation of the effectiveness of the proposed ensemble compared to the SVM ensemble. This is significant, as SVM remains one of the most widely used classifiers in both the academic literature and practical applications.

Table 2. Accuracy of different methods on the Badirli Dataset.

Badirli	Species	Genus
Bad	98.21	81.95
Bad _D	98.65	71.85
[45]	99.47	83.22
SVM _D	99.20	80.23
SVM _O	99.22	81.26
DWT _O	99.17	76.57
DWT _O + SVM _O	99.40	81.92
eDNA	98.74	87.20
Proposed	99.47	87.74

Table 3. Accuracy of different methods on the New Insects Dataset, Section 3.1.

New Insects	Species	Genus
[45]	99.05	84.02
SVM _D	99.16	79.07
SVM _O	98.21	82.01
SVM _N	98.05	65.05
SVM _f	99.00	83.67
DWT _O	98.83	75.08
DWT _N	98.54	82.98
DWT _f	99.12	83.85
DWT _f + SVM _f	99.15	85.51
eDNA	98.48	88.49
Proposed	98.99	91.35

Table 4. Accuracy of different methods on the Unseen Dataset, Section 3.2.

Unseen	Genus
SVM _D	23.85
SVM _O	32.75
SVM _N	46.82
SVM _f	47.70
DWT _O	30.38
DWT _N	28.25
DWT _f	42.73
DWT _f + SVM _f	48.15
eDNA	33.36
Proposed	48.56

Table 5. Accuracy of different methods on the Beetle Dataset.

Beetle	Species
[34]	98.10
[45]	98.20
SVM _D	95.13
SVM _O	90.48
SVM _N	97.69
SVM _f	97.69
DWT _O	—
DWT _N	86.66
DWT _N + SVM _f	98.03
eDNA	98.20
Proposed	98.51

Table 6. Accuracy of different methods on the Fish Dataset.

Fish	Species
[34]	96.30
SVM _D	94.75
SVM _O	91.73
SVM _N	95.70
SVM _f	95.61
DWT _O	—
DWT _N	92.82
DWT _N + SVM _f	96.83
eDNA	96.75
Proposed	97.03

The following conclusions can be obtained considering the tables reported in this section:

- In each dataset, one of the methods tested in this paper gets the new SOTA, and ensemble is proposed as the best method among the tested approaches. In general, the conclusions are different whether we use the large datasets or the two small ones (i.e., Beetle and Fish); this is, logically, because of the size of the training set, a more important factor for neural networks than SVM.
- Interestingly, between SVM and DWT, there is no winner; in some cases, SVM does better, in others DWT. However, their fusion allows both methods to improve. Performance is similar in the three large datasets, while in Beetle and Fish, which are much smaller in size, SVM performs much better than DWT; it is assumed that this is due to the fact that DWT is based on neural networks, which require a larger training set size than SVM.
- Similarly, also between the features proposed in [16] and those proposed in this paper, there is no clear winner; however, the fusion allows to improve the results of the individual methods. Again, there is a difference in the results between the three large datasets and the two smaller ones (Beetle and Fish), in which the features proposed in [16] perform badly.

Table 7. Precision–Recall performance indicator on different datasets.

Dataset	Approach	Precision	Recall
Beetle	SVM _f	99.73	100
	Proposed	99.81	100
Fish	SVM _f	99.48	100
	Proposed	99.57	100
Badirli—Species	SVM _O	99.82	98.71
	Proposed	99.42	99.14
Badirli—Genus	SVM _O	90.16	85.74
	Proposed	93.17	92.02
New—Species	SVM _f	99.46	99.00
	Proposed	99.38	98.99
New—Genus	SVM _f	90.91	89.69
	Proposed	95.90	95.63
Unseen	SVM _f	57.29	53.91
	Proposed	58.11	54.50

In the final test, reported in Figure 6, we adopt a realistic protocol for the proposed dataset, i.e., the one detailed in Section 3.1, where all the insects are first classified at the species level (the species are the classes); notice that we have two trained nets, one for species and one for genera classification. Let us suppose the following:

- $\theta_1(\mathbf{x})$ is the highest score (obtained using the species classification net) among the different species (i.e., classes) given a pattern \mathbf{x} ;
- $\theta_2(\mathbf{x})$ is the second highest score (obtained using the species classification net) of that pattern;
- $\theta(\mathbf{x}) = \theta_1(\mathbf{x}) - \theta_2(\mathbf{x})$.

Our rejection criteria are as follows:

- If $\theta(\mathbf{x}) > \tau$, the insect is assigned to a species class; otherwise, it is assigned to a genus class (i.e., it is classified by the network trained using the genus as classes).
- If a pattern belongs to a known species but is classified at the genus level, it is considered a classification error; clearly, a pattern with an unknown species is regarded as an error if classified at the species level.

In Figure 6, we report the plot of the species accuracy (x-axis) versus genus accuracy (y-axis) obtained by varying the rejection threshold τ . The green line is obtained by SVM_f , and the black line by our ensemble 'Proposed'. This test clearly shows the usefulness of the proposed ensemble versus SVM.

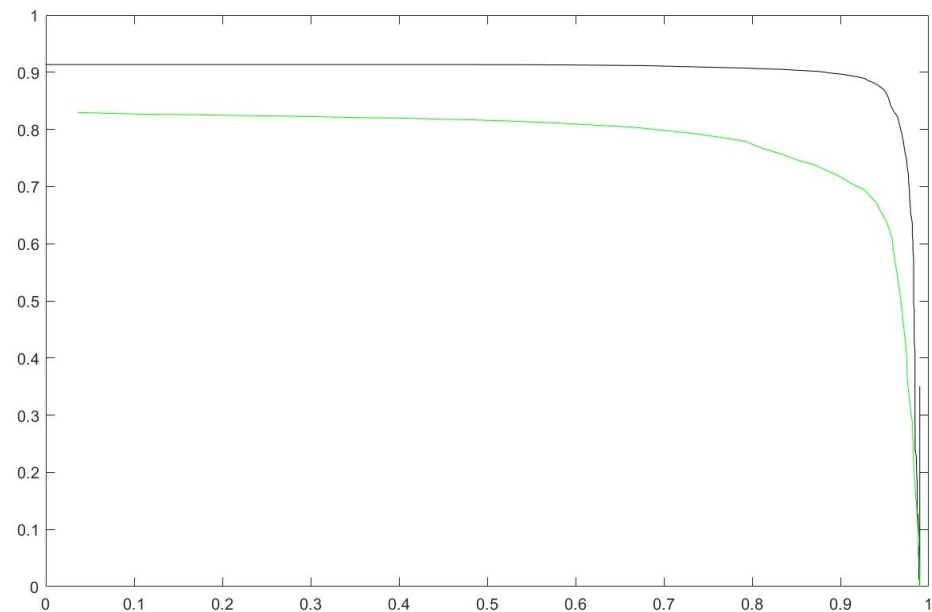


Figure 6. SVM_f (green line) vs. proposed ensemble (black line), with species accuracy (x-axis) versus genus accuracy (y-axis) obtained by varying the rejection threshold τ .

5. Discussion

Several interesting insights can be drawn from the results presented in the previous section. Let us examine some of them.

- As is often the case in machine learning and deep learning, no single method consistently outperforms all others across the datasets tested in this study. Instead, the best results are achieved through an ensemble approach. This suggests that different methods capture distinct aspects of the data, and their combination allows for a more comprehensive extraction of information, ultimately leading to superior performance compared to individual methods.
- It is important to highlight that methods relying solely on image-based features perform worse than those based on DNA barcoding. However, integrating both types of features—those extracted from images and those derived from DNA information—enhances the performance of a DNA-only classifier.
- Even when compared to state-of-the-art methods across various datasets, our ensemble approach emerges as the top performer. This reinforces our confidence that, despite the complexity of combining different techniques, the proposed ensemble method is robust across diverse datasets. As a result, it serves as a strong baseline for other researchers working with combined image and DNA barcoding data.
- Another notable observation comes from the plot in Figure 6. The proposed method demonstrates a substantial improvement over SVM_f , which is currently the most widely

used classification technique; this highlights the potential usefulness of our system for the research community. Our method achieves impressive performance reaching 90% species-level accuracy and 90% genus-level accuracy; this is significantly higher than the results obtained with SVM. Moreover, these improvements have practical implications for expert naturalists, as they facilitate the identification of previously unknown species based on DNA barcoding and imaging with high accuracy.

- Finally, an interesting result reported in Table 4 pertains to the Unseen dataset, where species identities are unknown. Notably, DNA barcoding performs the worst in this dataset compared to the others tested. Nevertheless, even in this case, the ensemble method yields the best results. The fact that DNA barcoding does not significantly outperform image-based methods suggests that this dataset differs from the others. This further underscores the generalizability of the proposed approach.

As previously depicted, the results clearly show the usefulness of the ensemble proposed in this work; obviously, the disadvantage of the ensemble is the higher computational power required to perform inference and training. So, these ensemble-based methods are certainly not suitable for real-time analysis on edge computing devices, and they require the ability to access a server where modern GPUs are available. This is not a problem in many applications, e.g., thanks to low-cost satellite connections such as Starlink, it is relatively easy to have access to the network, so the number of projects in which an ensemble can be used has been increasing in recent years. In contrast, in all those applications where classification must be carried out on an unconnected device, simply because of the need to reduce power consumption, these approaches are not the ideal choice, because of the computing power required.

The experiments were conducted on the following hardware setup:

- Processor: Intel Core i5-12400 CPU (2.5 GHz, six cores);
- GPU: NVIDIA RTX 4070 (12 GB GDDR6X) for accelerated deep learning tasks;
- RAM: 32 GB DDR5;
- Programming Languages: Python 3.12.2 (for feature extraction tasks) and MATLAB R2024b (for wavelet transforms and classification). Deep Learning Framework: PyTorch 2.0, utilized for implementing and training the convolutional neural networks used for extracting features. LibSVM was used for support vector machines. MATLAB's wavelet toolbox and deep learning toolbox were used to transform the feature vector into two-dimensional matrices and then to classify these using CNN.

The computation times achieved in our experiments are notably low, demonstrating the efficiency of the processes involved. Using a batch of 1000 patterns, the mean inference time for DNA barcoding feature extraction was only 0.256 s, reflecting swift performance even with complex biological data. Furthermore, the image feature extraction process exhibited exceptional speed, with a mean inference time of just 0.006 s. These results underline the computational efficiency of both methods, ensuring rapid data processing and scalability for larger datasets.

6. Conclusions

In this study, we investigated the performance benefits of combining neural networks with support vector machines (SVM). Our research contributes to the field by evaluating the effectiveness of integrating multiple classifiers to construct heterogeneous ensembles.

The key innovations introduced in this work are as follows. We proposed a novel method for building CNN ensembles by leveraging different mother wavelets for vector-to-matrix transformations and new methods for representing DNA sequences and images as feature vectors. We demonstrated the superior performance of the proposed method by comparing the ensembles with SVM-based models. We developed an ensemble that

surpasses previous state-of-the-art approaches and standalone SVM models. For future work, we plan to extend our analysis by incorporating additional datasets to improve the generalization of our method. We also aim to explore alternative techniques from the literature for generating matrices suitable for CNN training. Further directions include devising new methods to describe DNA barcodes, incorporating images of individual insects, and developing approaches to reject patterns lacking species labels. Finally, we intend to focus on distillation techniques and continuous learning strategies to enable edge computing and the inclusion of numerous new classes, addressing the challenge of catastrophic forgetting.

Author Contributions: Conceptualization, all authors; methodology, all authors; software, all authors; investigation, all authors; data curation, all authors; writing—original draft preparation, all authors; writing—review and editing, all authors; supervision, L.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: Data and code will be made available at: <https://github.com/LorisNanni/Advancing-Taxonomy-with-Machine-Learning-A-Hybrid-Ensemble-for-Species-and-Genus-Classification>, accessed March 28, 2025.

Acknowledgments: We would like to acknowledge the support that NVIDIA provided us through the GPU Grant Program. We used a donated GPU to train the CNNs used in this work.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Stork, N.E. How Many Species of Insects and Other Terrestrial Arthropods Are There on Earth? *Annu. Rev. Entomol.* **2018**, *63*, 31–45.
2. Costello, M.J.; May, R.M.; Stork, N.E. Can We Name Earth’s Species Before They Go Extinct? *Science* **2013**, *339*, 413–416.
3. Buck, M.; Woodley, N.; Borkent, A.; Wood, D.; Pape, T.; Vockeroth, J.; Michelsen, V.; Marshall, S. Key to Diptera Families-Adults; In *Manual of Central American Diptera*; CRC Press: Boca Raton, FL, USA, 2009; pp. 95–144.
4. Hebert, P.D.N.; Cywinska, A.; Ball, S.L.; deWaard, J.R. Biological identifications through DNA barcodes. *Proc. R. Soc. B Biol. Sci.* **2003**, *270*, 313 – 321.
5. Burns, J.M.; Janzen, D.H.; Hajibabaei, M.; Hallwachs, W.; Hebert, P.D.N. DNA barcodes and cryptic species of skipper butterflies in the genus *Perichares* in Area de Conservación Guanacaste, Costa Rica. *Proc. Natl. Acad. Sci. USA* **2008**, *105*, 6350–6355.
6. Ratnasingham, S.; Hebert, P. bold: The Barcode of Life Data System. *Mol. Ecol. Notes* **2007**, *7*, 355–364.
7. Or, M.C.; Ascher, J.S.; Bai, M.; Chesters, D.; Zhu, C.D. Three questions: How can taxonomists survive and thrive worldwide? *Megataxa* **2020**, *1*, 19–27.
8. De Gobbi, M.; De Almeida Matos Junior, R.; Lavezzi, L. *Insect DNA Barcode and Image Dataset*. 2024. <https://doi.org/10.5281/zenodo.14277812>, accessed March 28, 2025, Zenodo DOI dataset.
9. Haarika, R.; Babu, T.; Nair, R.R. Insect Classification Framework based on a Novel Fusion of High-level and Shallow Features. *Procedia Comput. Sci.* **2023**, *218*, 338–347.
10. Milošević, D.; Milosavljević, A.; Predić, B.; Medeiros, A.S.; Savić-Zdravković, D.; Stojković Piperac, M.; Kostić, T.; Spasić, F.; Leese, F. Application of deep learning in aquatic bioassessment: Towards automated identification of non-biting midges. *Sci. Total Environ.* **2020**, *711*, 135160.
11. Raitoharju, J.; Meissner, K. On Confidences and Their Use in (Semi-)Automatic Multi-Image Taxa Identification. In Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China, 6–9 December 2019; pp. 1338–1343.
12. Valan, M.; Makonyi, K.; Maki, A.; Vondráček, D.; Ronquist, F. Automated Taxonomic Identification of Insects with Expert-Level Accuracy Using Effective Feature Transfer from Convolutional Networks. *Syst. Biol.* **2019**, *68*, 876–895.
13. Fujisawa, T.; Noguerales, V.; Meramveliotakis, E.; Papadopoulou, A.; Vogler, A.P. Image-based taxonomic classification of bulk insect biodiversity samples using deep learning and domain adaptation. *Syst. Entomol.* **2023**, *48*, 387–401.
14. Buschbacher, K.; Ahrens, D.; Espeland, M.; Steinhage, V. Image-based species identification of wild bees using convolutional neural networks. *Ecol. Inform.* **2020**, *55*, 101017.
15. Hansen, O.L.P.; Svenning, J.C.; Olsen, K.; Dupont, S.; Garner, B.H.; Iosifidis, A.; Price, B.W.; Høye, T.T. Species-level image classification with convolutional neural network enables insect identification from habitus images. *Ecol. Evol.* **2020**, *10*, 737–747.

16. Badirli, S.; Picard, C.J.; Mohler, G.; Richert, F.; Akata, Z.; Dundar, M. Classifying the unknown: Insect identification with deep hierarchical Bayesian learning. *Methods Ecol. Evol.* **2023**, *14*, 1515–1530.
17. Badirli, S.; Akata, Z.; Mohler, G.; Picard, C.; Dundar, M.M. Fine-Grained Zero-Shot Learning with DNA as Side Information. In *Proceedings of the Advances in Neural Information Processing Systems*; Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2021; Volume 34, pp. 19352–19362.
18. Yang, C.H.; Wu, K.C.; Chuang, L.Y.; Chang, H.W. DeepBarcoding: Deep Learning for Species Classification Using DNA Barcoding. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2022**, *19*, 2158–2165.
19. Bertolazzi, P.; Felici, G.; Weitschek, E. Learning to classify species with barcodes. *BMC Bioinform.* **2009**, *10*, S7 – S7.
20. Sohsah, G.N.; Ibrahimzada, A.R.; Ayaz, H.; Cakmak, A. Scalable classification of organisms into a taxonomy using hierarchical supervised learners. *J. Bioinform. Comput. Biol.* **2020**, *18*, 2050026.
21. Tian, Q.; Zhang, P.; Zhai, Y.; Wang, Y.; Zou, Q. Application and Comparison of Machine Learning and Database-Based Methods in Taxonomic Classification of High-Throughput Sequencing Data. *Genome Biol. Evol.* **2024**, *16*, evae102.
22. Zito, A.; Rigon, T.; Dunson, D.B. Inferring taxonomic placement from DNA barcoding aiding in discovery of new taxa. *Methods Ecol. Evol.* **2023**, *14*, 529–542.
23. Jin, L.; Yu, J.; Yuan, X.; Du, X. Fish Classification Using DNA Barcode Sequences through Deep Learning Method. *Symmetry* **2021**, *13*.
24. Riza, L.S.; Ammar, M.; Rahman, F.; Prasetyo, Y.; Zain, M.I.; Siregar, H.; Hidayat, T.; Fariza, K.A.; Samah, A.; Rosyda, M. Comparison of Machine Learning Algorithms for Species Family Classification using DNA Barcode. *Knowl. Eng. Data Sci.* **2023**, *6*, 231.
25. Doan, T.N. Large-scale insect pest image classification. *J. Adv. Inf. Technol.* **2023**, *14*, 328–341.
26. Hedrick, B.P.; Heberling, J.M.; Meineke, E.K.; Turner, K.G.; Grassa, C.J.; Park, D.S.; Kennedy, J.; Clarke, J.A.; Cook, J.A.; Blackburn, D.C.; et al. Digitization and the Future of Natural History Collections. *BioScience* **2020**, *70*, 243–251.
27. Flück, B.; Mathon, L.; Manel, S.; Valentini, A.; Dejean, T.; Albouy, C.; Mouillot, D.; Thuiller, W.; Murienné, J.; Brosse, S.; et al. Applying convolutional neural networks to speed up environmental DNA annotation in a highly diverse ecosystem. *Sci. Rep.* **2022**, *12*, 10247.
28. Wührl, L.; Pylatiuk, C.; Giersch, M.; Lapp, F.; von Rintelen, T.; Balke, M.; Schmidt, S.; Cerretti, P.; Meier, R. DiversityScanner: Robotic handling of small invertebrates with machine learning methods. *Mol. Ecol. Resour.* **2022**, *22*, 1626–1638.
29. Klasen, M.; Ahrens, D.; Eberle, J.; Steinhage, V. Image-Based Automated Species Identification: Can Virtual Data Augmentation Overcome Problems of Insufficient Sampling? *Syst. Biol.* **2021**, *71*, 320–333.
30. Ärje, J.; Melvad, C.; Jeppesen, M.R.; Madsen, S.A.; Raitoharju, J.; Rasmussen, M.S.; Iosifidis, A.; Tirronen, V.; Gabbouj, M.; Meissner, K.; et al. Automatic image-based identification and biomass estimation of invertebrates. *Methods Ecol. Evol.* **2020**, *11*, 922–931.
31. MacLeod, N.; Canty, R.J.; Polaszek, A. Morphology-Based Identification of Bemisia tabaci Cryptic Species Puparia via Embedded Group-Contrast Convolution Neural Network Analysis. *Syst. Biol.* **2021**, *71*, 1095–1109.
32. Impiö, M.; Raitoharju, J. Improving Taxonomic Image-based Out-of-distribution Detection with DNA Barcodes. *arXiv* **2024**, arXiv:2406.18999. Available online: <http://arxiv.org/abs/2406.18999> (accessed on).
33. Blair, J.D.; Weiser, M.D.; Siler, C.; Kaspari, M.; Smith, S.N.; McLaughlin, J.F.; Marshall, K.E. A hybrid approach to invertebrate biomonitoring using computer vision and DNA metabarcoding. *bioRxiv* **2024**. <https://doi.org/10.1101/2024.09.02.610558>.
34. Yang, B.; Zhang, Z.; Yang, C.Q.; Wang, Y.; Orr, M.C.; Wang, H.; Zhang, A.B. Identification of Species by Combining Molecular and Morphological Data Using Convolutional Neural Networks. *Syst. Biol.* **2021**, *71*, 690–705.
35. Kang, M.; Shim, W.; Cho, M.; Park, J. Rebooting acgan: Auxiliary classifier gans with stable training. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 23505–23518.
36. Miyato, T.; Kataoka, T.; Koyama, M.; Yoshida, Y. Spectral Normalization for Generative Adversarial Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada, 30 April–3 May 2018.
37. Kang, M.; Shin, J.; Park, J. StudioGAN: A Taxonomy and Benchmark of GANs for Image Synthesis. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2023**, *45*, 15725–15742.
38. Wu, X.; Zhan, C.; Lai, Y.; Cheng, M.M.; Yang, J. IP102: A Large-Scale Benchmark Dataset for Insect Pest Recognition. In *Proceedings of the IEEE CVPR*, Long Beach, CA, USA, 15–20 June 2019; pp. 8787–8796.
39. De Gobbi, M.; De Almeida Matos Junior, R.; Lavezzi, L. Animal Image dataset for GAN pretraining. **2024**. <https://doi.org/10.5281/zenodo.14577906>, accessed March 28, 2025, Zenodo DOI dataset.
40. Mallat, S. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*, 3rd ed.; Academic Press, Inc.: Cambridge, MA, USA, 2008.
41. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; Section 5.7.2.
42. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 27–30 June 2015.

43. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), **Miami, FL, USA, 20–25 June** 2009; pp. 248–255.
44. Nanni, L.; Cuza, D.; Brahnam, S. AI-Powered Biodiversity Assessment: Species Classification via DNA Barcoding and Deep Learning. *Technologies* **2024**, *12*, 240. <https://doi.org/10.3390/technologies12120240>.
45. Nanni, L.; Maritan, N.; Fusaro, D.; Brahnam, S.; Meneguolo, F.B.; Sgaravatto, M. Insect identification by combining different neural networks. *Expert Syst. Appl.* **2024**, *in review*.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.