Received: 2 March 2021 | Revised: 6 June 2021 | Accepted: 8 June 2021

DOI: 10.1002/ece3.7845

ORIGINAL RESEARCH

Ecology and Evolution Open Access WILEY

# Using comprehensive machine-learning models to classify complex morphological characters

Dequn Teng[1] | Fengyuan Li[1] | Wei Zhang[1,2]

[1]State Key Laboratory of Protein and Plant Gene Research, School of Life Sciences, Peking University, Beijing, China

[2]Peking-Tsinghua Center for Life Sciences, Academy for Advanced Interdisciplinary Studies, Peking University, Beijing, China

**Correspondence**
Wei Zhang, State Key Laboratory for Protein and Plant Gene Research, Peking-Tsinghua Center for Life Sciences, School of Life Sciences, Peking University, Beijing 100871, China.
Email: weizhangvv@pku.edu.cn

**Funding information**
National Natural Science Foundation of China, Grant/Award Number: 31871271; China Postdoctoral Science Foundation, Grant/Award Number: 2020M680214; Natural Science Foundation of Beijing Municipality, Grant/Award Number: JQ19021; Qidong-SLS Innovation Fund

## Abstract

1. Recognizing and classifying multiple morphological features, such as patterns, sizes, and textures, can provide a comprehensive understanding of their variability and phenotypic evolution. Yet, quantitatively measuring complex morphological characters remains challenging.
2. We provide a machine learning-based pipeline (SVMorph) to consider and classify complex morphological characters in multiple organisms that have either small or large datasets.
3. Our pipeline integrates two descriptors, histogram of oriented gradient and local binary pattern, to meet various classification needs. We also optimized feature extraction by adding image data augmentation to improve model generalizability.
4. We tested SVMorph on two real-world examples to demonstrate that it can be used with small training datasets and limited computational resources. Comparing with multiple CNN-based methods and traditional techniques, we show that SVMorph is reliable and fast in texture-based individual classification. Thus, SVMorph can be used to efficiently classify multiple morphological characters in distinct nonmodel organisms.

KEYWORDS

classification, data augmentation, feature extraction, machine learning, morphological characters, SVMorph

## 1 | INTRODUCTION

As intuitive reflections of biodiversity, morphological characteristics provide valuable taxonomic insights into phenotypic evolution and its underlying genetic mechanisms (Grant & Grant, 2006; Joron et al., 2006; Peichel et al., 2001). Therefore, comparative morphology is generally the indispensable, first step of organismal feature identification and investigation. Yet, quantitatively measuring morphological variation remains challenging.

Fortunately, imaging techniques have advanced so that morphological characteristics can be acquired easily. Commonly, morphological features of interest are measured manually and then classified based on measured data, and the usual geometric morphometrics describes individuals through relative landmark positions (Bookstein, 1996). For example, *Drosophila* wings are two-dimensional structures with clear venation patterns, so the veins and their crossings are usually considered landmarks whose distances and angles can be measured precisely (Cavicchi et al., 1981; Guerra et al., 1997).

However, using those manual methods with large-scale datasets is both difficult and inefficient. Researchers have developed several advanced methods, such as WINGMACHINE (Houle et al., 2003), DrosoWing (Loh et al., 2017), and FijiWings (Dobens & Dobens, 2013), which detect *Drosophila* landmark information automatically, and MORPHOJ (Klingenberg, 2011), which performs further morphological analyses based on landmark data. Similarly, some automated systems including machine learning-based methods use fundamental wing morphological features to classify specific insects. For example, using machine learning-based approaches Crnojević et al. (2014) detected vein junctions to discriminate hoverfly species. Yang et al. (2015) developed DAIIS, which identifies owlfly species according to wing outlines. Also, color pattern modeling characterizes *Heliconius* butterfly phenotypes by comparing color patterns (Le Poul et al., 2014); a machine learning-based approach classifies guenon face patterns (Allen & Higham, 2015); and Patternize, another machine learning-based method that works for multiple organisms, uses color pattern variations (Van Belleghem et al., 2018). While those approaches are useful, either for focal organisms or for processing a specific morphological feature, they are not easily used with organisms other than their original targets or for detecting and classifying more complex features that may harbor a wider range of morphological variability in a combination of patterns, shapes, and textures. A comprehensive and efficient solution that considers and handles such integrative features is still required.

As illustrated above, machine learning has recently provided remarkable image processing solutions that can both efficiently identify complex features and accomplish classification tasks (Lürig et al., 2021). Note that although deep-learning algorithms are widely applied to feature extraction and classification, they require relatively larger training sets and cannot appropriately handle classification problems with fewer samples. In contrast, support vector machine (SVM) universally handles generalized problems and performs well when confronted by a limited or imbalanced sample size, for the decision boundary can be determined by a few support vectors (Tang et al., 2009). Therefore, we developed an SVM-based pipeline, SVMorph, for complex morphological classification. To begin, SVMorph tackles many classification needs by combining two optional descriptors, histogram of oriented gradient (HOG) (Dalal & Triggs, 2005) and local binary pattern (LBP) (Baraldi & Panniggiani, 1995), to extract morphological features from images, and it then trains an SVM classifier to carry out classification tasks. We also employ data augmentation to improve model generalizability, which aims to increase the dataset size by appropriately modifying existing dataset. To illustrate SVMorph's usage and performance, we used it to classify two groups of organisms that possess distinct and polymorphic features. Ultimately, SVMorph demonstrated its considerable potential and universality by excellently and proficiently handling complex morphological classification.

## 2 | SVMORPH WORKFLOW

We developed SVMorph in the MATLAB environment (MathWorks Inc.). The overall workflow for a typical task generally includes data preprocessing, feature extraction, and classification (Figure 1). When conducting a new task, a classifier must be trained and established before new data may be processed.

## 3 | DATA ACQUISITION AND PREPROCESSING

While raw images may be captured using digital cameras that can provide better image resolution, we recommend two-dimensional structures, such as insect wings, be photoscanned for better imaging homogeneity. For example, using a regular scanner (HP LaserJet Pro M227fdw) with a resolution of 600 dpi, we obtained a single forewing image of about 1,000 × 1,000 pixels. While using a digital camera (NIKON D850) with a 1:1 macro lens, we could acquire a single forewing image of over 3,000 × 3,000 pixels with much higher resolution. However, SVMorph or other deep learning-based algorithms only require 256 × 256-pixel or 224 × 224-pixel images as input, and even lower resolutions work as well. Therefore, the need for image resolution is easily satisfied, so we are more concerned with image homogeneity. When the images come from different sources, it may be difficult to ensure the consistency of the photographing equipment or conditions. Relatively, scanners can provide stable photographing conditions and low operational complexity to obtain images with similar quality. Nevertheless, when digital cameras are properly set up, image homogeneity will not be a critical issue. In addition, data augmentation has the potential to reduce the effects of unstable photographing conditions. Therefore, ordinary photographing devices and normal conditions can satisfy the requirements of SVMorph. After data acquisition, SVMorph requires all input image data be preprocessed as follows. Given that both HOG and LBP feature descriptors extract feature information from eight-bit grayscale images, raw images should be transformed into eight-bit grayscale format (Figure 2a,b) and the appropriate brightness, contrast, and exposure are required (Figure 2a) to ensure optimum classification accuracy. For our sample datasets, we adjusted the butterfly images with the following parameters: brightness 10 and contrast −50, and we did not adjust the spider images. Note that the adjustments in this step are optional. For photoscanned images, the adjustment in this step is equivalent to setting these parameters correctly when photographing with a digital camera. We want the morphological features we care most about to be well expressed in the images. Therefore, adjustments such as lighting and exposure can depend on the visual effects of the images, that is, whether key morphological features can be clearly distinguished visually. In general, there are no particular restrictions on photographing, but the conditions need to be consistent for a classification task.

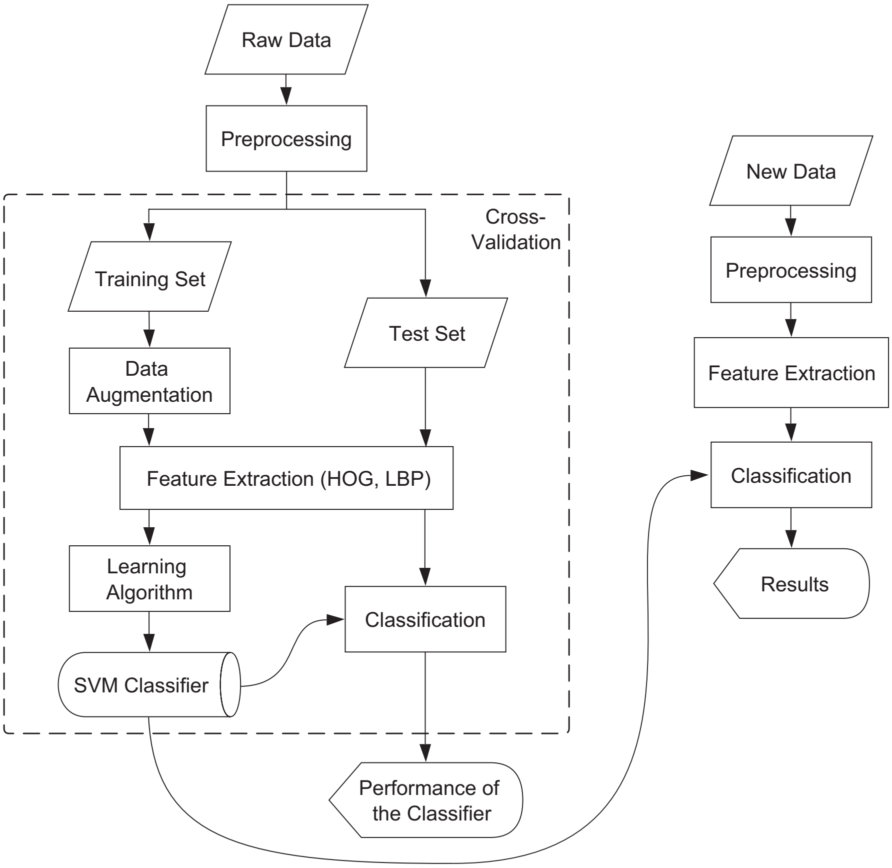

**FIGURE 1** SVMorph framework and workflow. HOG, histogram of oriented gradient (Dalal & Triggs, 2005); LBP, local binary pattern (Baraldi & Panniggiani, 1995); and SVM, support vector machine

Regarding the sample size, as mentioned above, SVM generally performs well compared to other algorithms when faced with a limited or imbalanced sample size, because the decision boundary of SVM is determined only by a few samples (support vectors). Besides, the data augmentation step can improve the classification to some extent as well. Therefore, SVMorph is less affected by the limited sample size. In practice, the criterion of sufficient sample size is difficult to predict and needs to be determined based on the actual model performance. Nevertheless, the model performance generally improves with the increasing sample size, so larger sample size is always preferred. On the other hand, when adding samples, highly imbalanced class distributions of samples should be avoided as well. In practice, we recommend at least 10 samples for each class and the ratio of sample sizes between different classes should be <5. For example, if class A has a sample size of 10 which is the least among all classes, the sample size of other classes should not exceed 50 for each.

## 4 | DATA AUGMENTATION

For pattern recognition tasks, a set of preprocessed, classified, and labeled image data functions as a training dataset, and many training samples usually produce the most robust model. However, the practical circumstances of morphological studies underlie the critical need for a data augmentation step in SVMorph, where the rarity of specific specimens often makes sample collection very challenging. Data augmentation artificially creates synthetic training data with label-preserving transformations by adding slight modifications such as cropping, flipping, rotation, and color jittering, which effectively deal with limited sample size datasets.

Additionally, data augmentation may potentially involve both prior knowledge about the data and task-specific invariances, each of which can regularize the model (Dao et al., 2019). Therefore, to promote task-specific invariances and generalization of classification models, different augmentation methods are needed for different tasks. Based on data-specific invariances and features, the preprocessed training datasets may undergo one or more augmentation methods (Figure 2c,d) by implementing the *augmentation.m* script in SVMorph. In our sample datasets, the butterfly wing patterns are less sensitive to parameters such as resolution, brightness, and contrast, so we adopted extensive augmentation methods for butterfly images. As for spider images with texture features more susceptible to the abovementioned image factors, we applied fewer methods to avoid distorting the texture features during the augmentation process. Despite the ubiquity of data augmentation in image classification, its underlying theoretical principles are not well understood. Therefore, in practice, the choice of data augmentation methods often depends on the actual performance. Basically, cropping and flipping methods are more reliable and commonly used, which may be considered as the primary choices. We used random parameters for data augmentation, which refer to the specific input arguments of data augmentation methods. For example, in the noise addition step, we introduced Gaussian white noise with random mean values and random variance values. Thus, different images were augmented

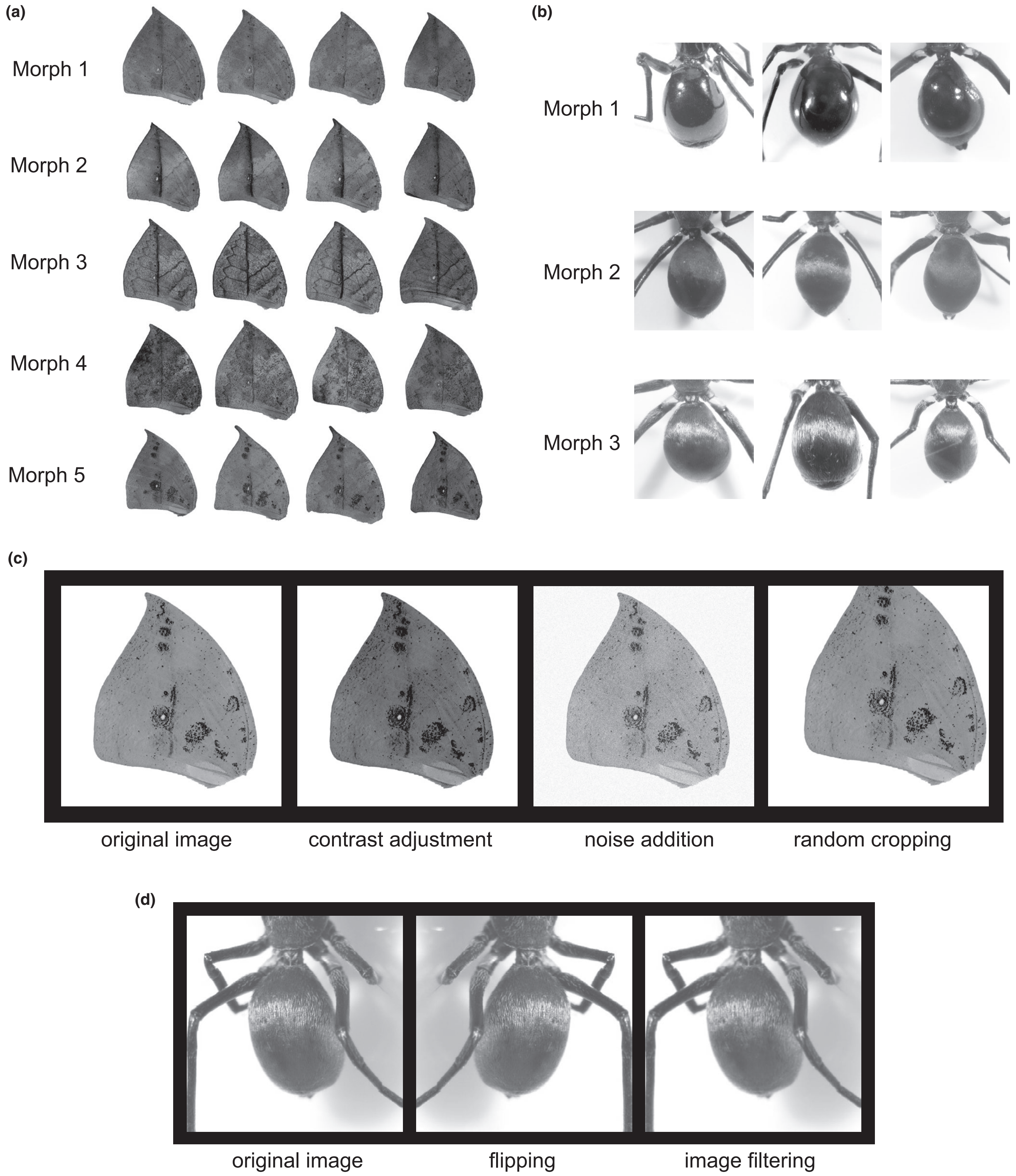

**FIGURE 2** Image data preprocessing and augmentation. (a) Examples of preprocessed grayscale images of polymorphic wing patterns of the dead-leaf butterfly, *Kallima inachus*. Morph 1: light-colored main vein; Morph 2: dark-colored main vein; Morph 3: dark-colored main vein with lateral veins; Morph 4: light-colored main vein with dense black dots; and Morph 5: light-colored main vein with black moldy-looking spots. (b) Examples of original images showing the polymorphic body textures of the jumping spider species complex in the *Toxeus* genus. Morph 1: mostly hairless abdomen; Morph 2: abdomen with sparse fine black hair; and Morph 3: abdomen densely covered with white hair. (c) Data augmentation examples of contrast adjustment, noise addition, and random cropping of dead-leaf butterfly wing images. (d) Data augmentation examples showing flipping and image filtering of jumping spider images

differently, which could increase the diversity of the image datasets and avoid introducing bias during the augmentation process. We also tested the effects of data augmentation on the two sample datasets, subsequently demonstrating that data augmentation significantly improved model accuracies in both cases (Figure 3). Therefore, data augmentation is necessary for optimal model performance.

## 5 | FEATURE EXTRACTION

SVMorph offers two descriptors for feature extraction from the augmented training dataset: HOG (Dalal & Triggs, 2005) and LBP (Baraldi & Panniggiani, 1995). To classify organisms that display a combination of various elements and that produce local edge and gradient information, we recommend HOG as the major feature descriptor. HOG feature extraction is implemented by the *extractHOGFeatures* function in the Computer Vision Toolbox in MATLAB 2020b (https://www.mathworks.com/products/computer-vision.html). Once implemented, HOG divides images into small spatial regions called *cells*, computes discrete histograms for each cell, and then assembles the cells into larger spatial regions called *blocks* to normalize histograms of all the cells in a block (Dalal & Triggs, 2005), which helps to maintain better invariance and to illuminate changes or shadowing (Figure 4). For texture classification tasks, we recommend LBP be used together with HOG. LBP calculates neighboring pixel gray values to a center pixel and encodes them as a binary number (Baraldi & Panniggiani, 1995), which also endows LBP the advantage of computational simplicity (Figure 5). To further reduce computation time, we use the *extractLBPFeatures* function in the Computer Vision Toolbox of MATLAB 2020b (https://www.mathworks.com/products/computer-vision.html) to implement a uniform LBP that corresponds to the number of spatial bitwise transitions in LBPs. For example, there would be only 59, instead of 256, total patterns for an eight-bit LBP operator. Finally, both HOG and LBP (if any) feature vectors from each image are extracted and combined for subsequent downstream classification.

## 6 | CLASSIFIER TRAINING

To train multiclass SVM models, we implement the *fitcecoc* function in the Statistics and Machine Learning Toolbox (https://www.mathworks.com/products/statistics.html) in MATLAB 2020b. That function adopts a one-versus-one coding design that includes a set of binary SVM classifiers for each possible pair in all classes, so for a model of $n$ classes, $n(n-1)/2$ binary SVM classifiers are trained. We improve classification accuracy by applying the error-correcting output code multiclass model, which is helpful to deal with multiclass classification problems based on multiple binary classifications (Dietterich & Bakiri, 1994). For the classification tasks using example datasets, all feature extraction and classifier training were performed on a computer with an Intel i7-10700K (3.80G Hz) processor and 32 GB of RAM under a 64-bit Windows operating system.

## 7 | CLASSIFIER TESTING

During the evaluation step, we used $k$-fold cross-validation, which partitions all samples into $k$ subsets randomly, to examine classifier performances. While a single subset is reserved as test data, the

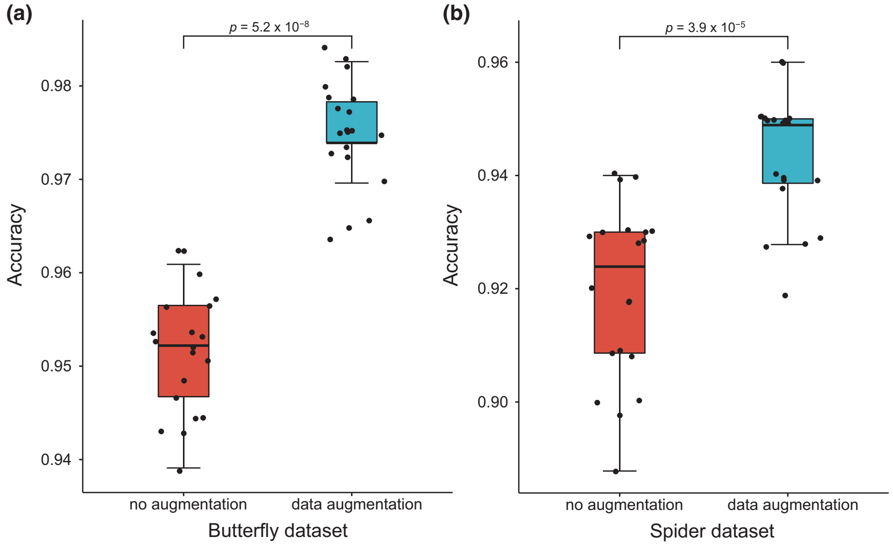

**FIGURE 3** Data augmentation effects on model performance. To compare classification performances before and after data augmentation, we calculated accuracy for each replicate (20 replicates for each group) of 10-fold cross-validations for the butterfly (a) and spider (b) datasets, where data are median (*SD*). The average accuracy for the butterfly dataset increases from 0.9513 (0.0064) to 0.9748 (0.0054) after augmentation and from 0.9196 (0.0154) to 0.9433 (0.0110) for the spider dataset after augmentation. Both (a) and (b) show a significantly higher degree of accuracy with augmentation than without augmentation. Wilcoxon rank-sum test. Boxes enclose scores within the first and third quartiles, and the whiskers show the minimum and maximum values

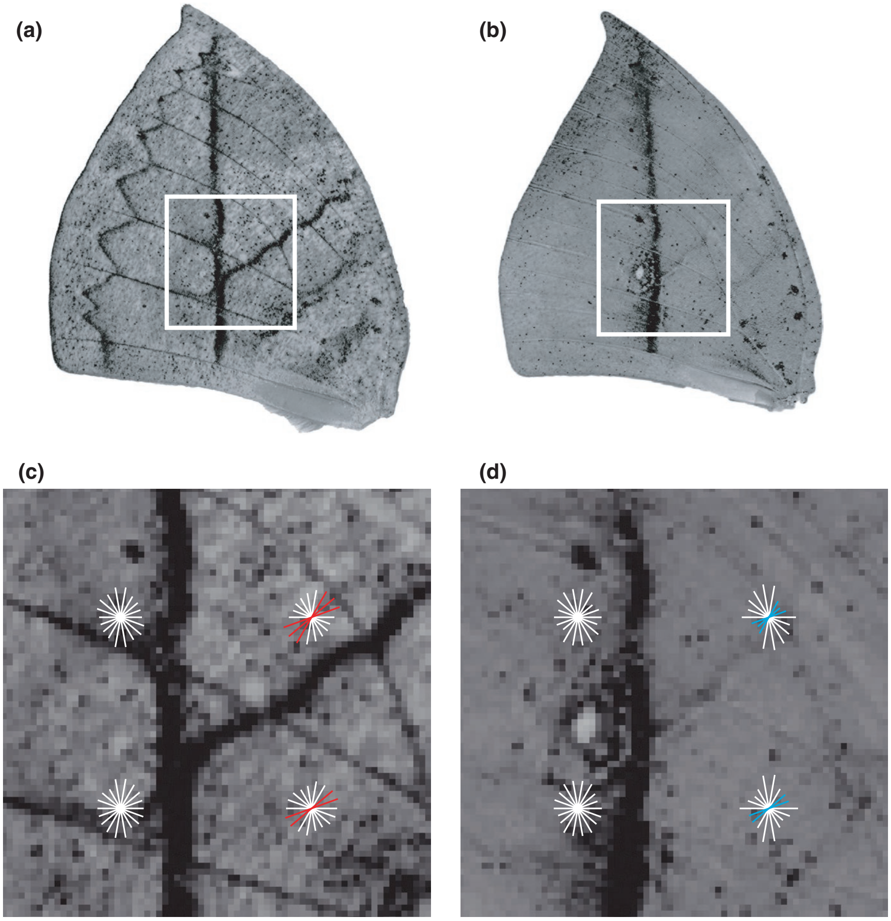

**FIGURE 4** Local histogram of oriented gradient (HOG) feature extractions from the butterfly wing patterns. We chose focal regions (in white boxes) from typical 256 × 256-pixel image data of *Kallima inachus* Morph 3 (a) and Morph 2 (b), with a cell size of 32 × 32 pixels and a block size of 2 × 2 cells. (c, d) Rose plots of the local HOG features extracted from each specific area in the white boxes in (a) and (b), respectively. Petal length indicates the gradient orientation distribution of a direction within a cell. Petals showing sharp differences between two wing forms are highlighted in red (c) and blue (d)

remaining *k*-1 subsets are used to train the model. A confusion matrix is an *n* × *n* table (for *n* classes), whose rows represent true labels and columns represent predicted labels. The elements on the main diagonal of a confusion matrix indicate correct predictions and, ideally, all off-diagonal elements equal to zero. The percentage of correct predictions in all test samples defines the classification model's accuracy and, based on the proportion of all main diagonal values in the confusion matrix, is computed as follows:

$$\text{accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \tag{1}$$

where TP, TN, FP, and FN stand for true positive, true negative, false positive, and false negative, respectively. To more specifically evaluate a model's performance, we used the precision, recall, and F1 scores that we computed for each class, where

$$\text{precision} = \text{TP} / (\text{TP} + \text{FP}), \tag{2}$$

$$\text{recall} = \text{TP} / (\text{TP} + \text{FN}), \tag{3}$$

$$\text{F1 score} = 2 \times \text{recall} \times \text{precision} / (\text{recall} + \text{precision}). \tag{4}$$

For each of our models, we calculated a confusion matrix for each round of 10-fold cross-validations and evaluated the measures calculated from the matrices. We then performed 20 replicates of 10-fold cross-validations for each classifier and used the average of 20 results to calculate an estimate of the model's performance. For each model, we created a receiver operating characteristics (ROC) curve by plotting the true-positive rate against the false-positive rate at various threshold values. For ROC calculation, all samples were randomly halved into a training dataset and a test dataset and then the training dataset's data were augmented. We used the FitPosterior argument of the *fitcecoc* function to obtain posterior probabilities for the test sample labels. For our multiclass classification model, we first computed ROC scores for each class against the rest of the classes by using the *roc_curve* function in the scikit-learn toolbox (Pedregosa et al., 2011) in Python 3.9 and then calculated a macro-averaged ROC as the mean of the binary patterns, which assumes all classes are equally weighted. Area under the curve (AUC) of the macro-averaged ROC for each model was computed to illustrate their performances (Figure 6).

## 8 | OUTPUT

Our optimally performing, trained and tested classifiers can be used for relevant data classification tasks. Using the *prediction.m* script in SVMorph, we designed classifiers to output predicted labels directly. New data that have undergone preprocessing can be classified using the established classifier, and that results in the most relevant labels being assigned to the image data (Figure 7).

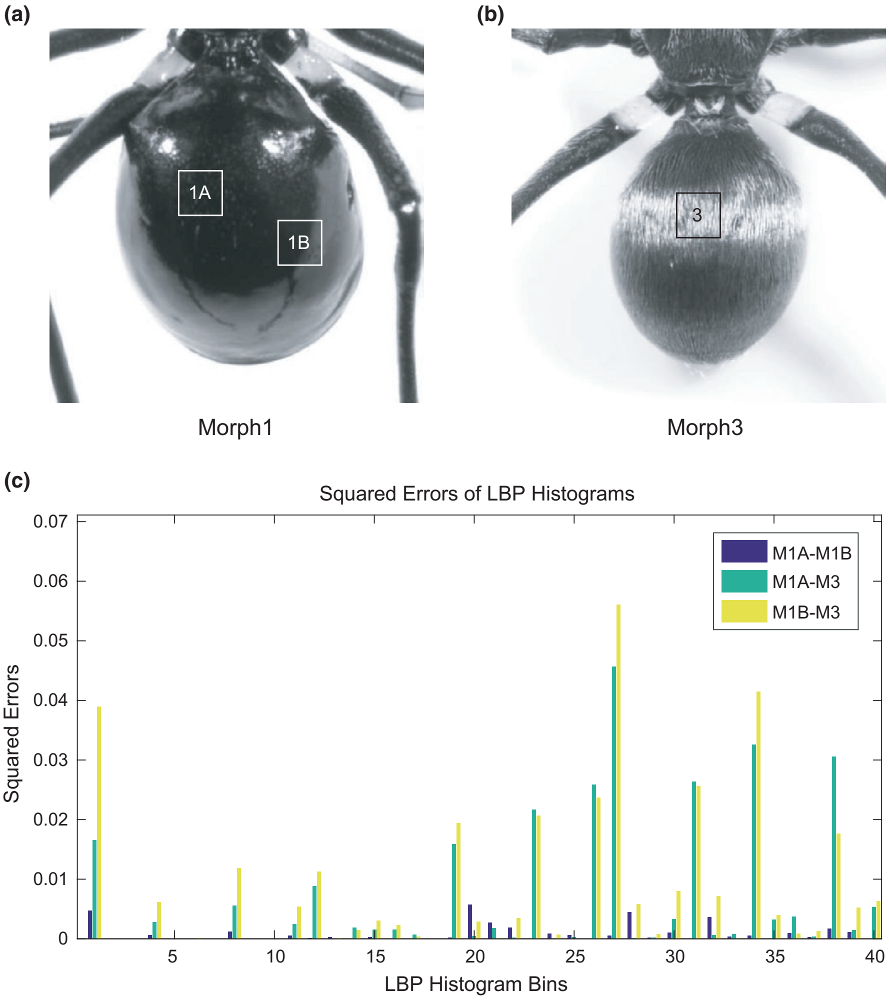

**FIGURE 5** Local binary pattern (LBP) feature extraction of spider body textures. We chose focal regions (in a white or black box) from 512 × 512-pixel typical image data of *Toxeus* genus Morph 1 (a) and Morph 3 (b), with a cell size of 64 × 64 pixels. (c) Squared errors of the histograms calculated for each cell in boxes 1A and 1B (a) and 3 (b) compare LBP features extracted from the focal regions. Squared errors are relatively smaller for regions with similar textures (M1A–M1B) and relatively larger for regions with different textures (M1A–M3 and M1B–M3)

## 9 | USAGE EXAMPLES

### 9.1 | Wing pattern classification for dead-leaf butterflies

Dead-leaf butterflies (*Kallima* spp.) are well known for the leaf patterns on the ventral sides of their wings and are thus a classic example of morphological adaptation (Suzuki et al., 2019). Traditionally, those complex morphological variations have been difficult to quantify precisely and efficiently, so we employed SVMorph to establish an SVM model and to classify those polymorphic leaf wing characters. According to the prior knowledge, we labeled them from Morph 1 to Morph 5 (Figure 2a).

During data acquisition and preprocessing, we photoscanned (HP LaserJet Pro M227fdw) ventral forewing images from 230 K. *inachus* individuals. The subsequent RGB PDF images each had <600 dpi resolution, were extracted using Adobe Photoshop (Adobe Photoshop CC 2018 v19.1.9, Adobe Systems Inc.), and were merged together for batch operations. Then, we preprocessed all the images by first transforming them into eight-bit grayscale format with brightness and contrast adjusted to 10 and −50, respectively, and then cropped them into separate wing images using the *imagecrop.m* script in SVMorph. Then, the images used for the training dataset were classified and labeled manually according to our a priori knowledge ($n_{\text{Morph 1}} = 77$, $n_{\text{Morph 2}} = 56$, $n_{\text{Morph 3}} = 60$, $n_{\text{Morph 4}} = 18$ and $n_{\text{Morph 5}} = 19$). During data augmentation, preprocessed 256 × 256-pixel wing images were randomly cropped to 224 × 224 (Figure 2c). Then, we randomly introduced Gaussian white noise with means ranging from 0.1 to 0.2 and variances ranging from 0.02 to 0.05 (Figure 2c). Contrast values used as input limits were randomly adjusted at low intensity values ranging from 0.0 to 0.1 and high intensity values ranging from 0.9 to 1.0 (Figure 2c). Since the butterfly wing patterns appear in a combination of various elements, we selected HOG as the major feature descriptor. In order to capture relative large-scale wing pattern information while saving computation time, we set 32 × 32-pixel cells and 2 × 2-cell blocks for butterfly images of 256 × 256 pixels. We used nine orientation bins that were evenly spaced from 0 to 180 degrees to encode oriented gradient information (Figure 4) and extracted 1,764-dimensional HOG feature vectors from each image. Considering those high-dimensional feature vectors, we selected the SVM model with the linear kernel function to reduce computational complexity and overfitting, which also guarantees that the model has fine generalization properties. To evaluate the butterfly classifier's performance, we evaluated accuracy, precision, recall, and F1 scores (Table 1) by using 20 replicates of 10-fold cross-validation. Using data augmentation, the butterfly classifier's overall accuracy

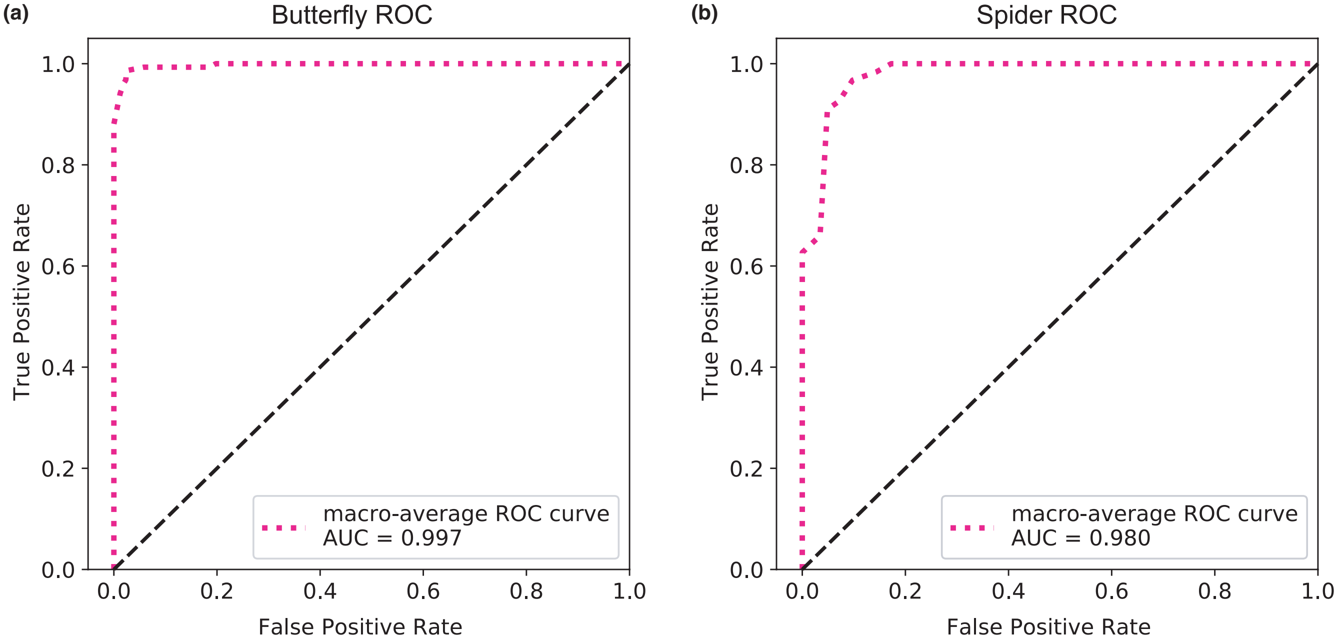

**FIGURE 6** Receiver operating characteristic (ROC) curves for multiclass classifications. Macro-averaged ROCs and areas under the curve (AUCs) illustrate the predictive abilities of the trained butterfly (a) and spider (b) classifiers. ROC curves plot the true-positive rate against the false-positive rate at various threshold settings, and the threshold settings are determined by posterior probability distributions of all the samples. Both ROC curves in (a) and (b) are far from the diagonals, indicating a high level of model performance relative to random guessing

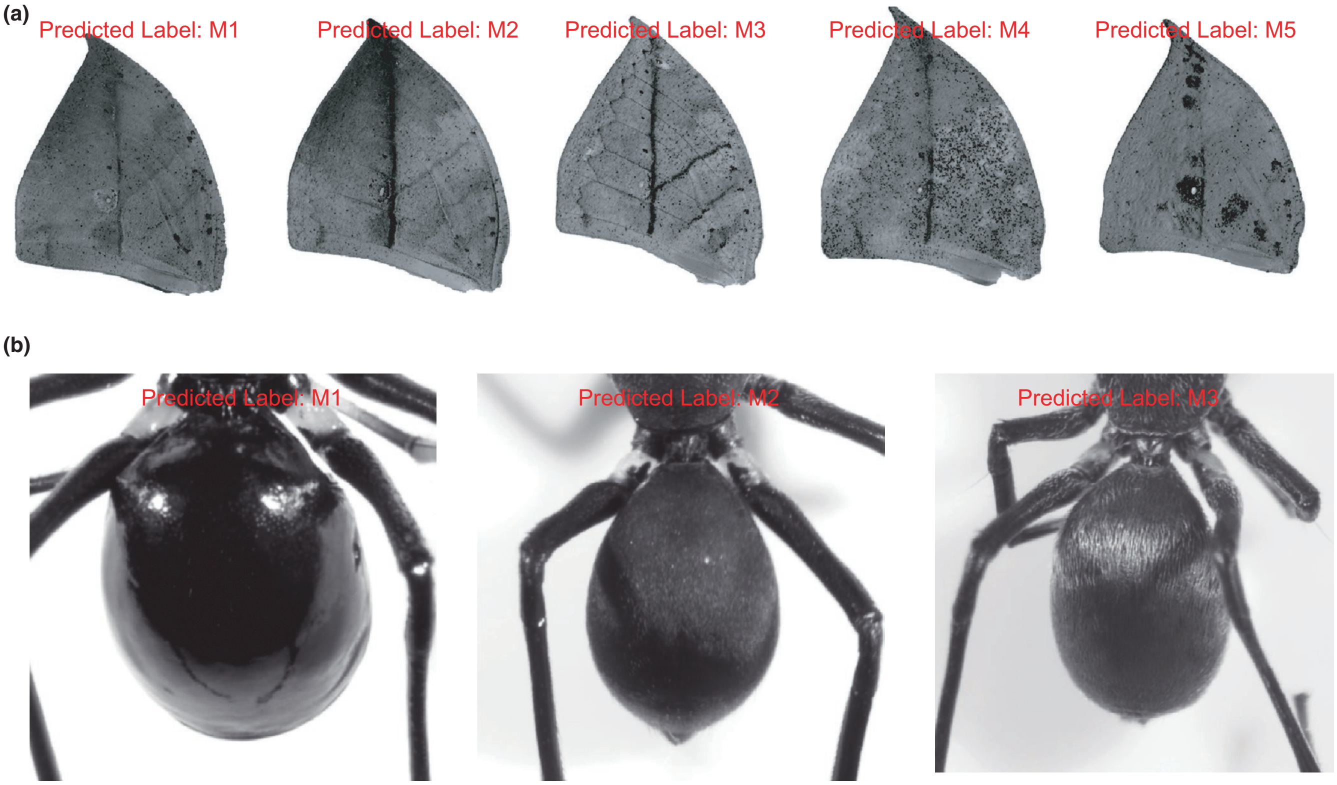

**FIGURE 7** Classifier output that can be used for classification tasks. Examples of the output labels predicted by the trained classifiers for dead-leaf butterflies (a) and jumping spiders (b) using test data. Labels M1–M5 correspond, respectively, to butterfly morphs 1–5 shown in Figure 2a, and labels M1–M3 correspond, respectively, to spider morphs 1–3 shown in Figure 2b

**TABLE 1** Evaluation measures for the butterfly wing classifier (with data augmentation)

| | M1 | M2 | M3 | M4 | M5 | overall |
|---|---|---|---|---|---|---|
| Precision | 0.9792 | 0.9300 | 1.0000 | 1.0000 | 1.0000 | 0.9818 |
| Recall | 0.9760 | 0.9920 | 0.9833 | 1.0000 | 0.8684 | 0.9639 |
| F1 score | 0.9776 | 0.9599 | 0.9916 | 1.0000 | 0.9290 | 0.9716 |
| Accuracy | — | — | — | — | — | 0.9748 |

*Note:* M1–M5 correspond, respectively, to butterfly morphs 1–5 shown in Figure 2a.

**TABLE 2** Evaluation measures for the jumping spider classifier (with data augmentation)

| | M1 | M2 | M3 | overall |
|---|---|---|---|---|
| Precision | 0.9921 | 0.9014 | 0.9734 | 0.9556 |
| Recall | 0.8342 | 0.9726 | 0.9645 | 0.9238 |
| F1 score | 0.9051 | 0.9354 | 0.9688 | 0.9365 |
| Accuracy | — | — | — | 0.9433 |

*Note:* M1–M3 correspond, respectively, to spider morphs 1–3 shown in Figure 2b.

reached 97.4%, while most classes had fine precision, recall, and F1 scores. However, Morph 5 had a relatively low recall value (86.8%), most likely due to its small sample size or high within-group variation. The trained model's macro-averaged ROC curve showed very good predictive ability for butterfly classification (Figure 6a).

Now using the trained classifier, we classified the preprocessed test data and labeled it accordingly (Figure 7a). The subsequent computational feature extraction and classification processes for this *K. inachus* dataset with 1,764-dimensional feature vectors took approximately 0.03 s per image to run. According to our results, the butterfly image data were classified and assigned accurately, efficiently, and appropriately to the abovementioned morphs, thus demonstrating the power of SVMorph to extract and distinguish complex wing patterns.

### 9.2 | Body texture classification in jumping spiders

*Toxeus* spp. jumping spiders are highly polymorphic mimics that resemble unpalatable *Polyrhachis* ants and thus have a protective advantage (Yamasaki & Ahmad, 2013), providing an excellent example with which to test SVMorph. Among them, members of the species complex of *T. maxillosus*, *T. magna*, and *T. globose* are closely related and have three mimicry morphs according to the body texture features that we named from Morph 1 to Morph 3 (Figure 2b).

Using a microscope (Nikon SMZ18) with an attached digital camera (Nikon DS-Ri2), we photographed 99 individual spiders. Raw images were captured using SHR Plan Apo 1×, at zoom magnification of 0.75× with Nikon C-FLED2 LED Light Source. We directly transformed the images into eight-bit grayscale format and zoomed into the abdominal areas (Figure 2b). Finally, we classified and labeled the images manually according to their phenotype information ($n_{\mathrm{Morph}\ 1} = 19$, $n_{\mathrm{Morph}\ 2} = 42$, and $n_{\mathrm{Morph}\ 3} = 38$). Since the spider data were more sensitive than the previous butterfly wing data, we performed data augmentation in the same way as we did for the butterfly data except for noise addition and contrast adjustment (Figure 2d). The images were filtered using an averaging filter with a hsize of [3 3] and the convolution option. During feature extraction, we used the same HOG parameters that we had used for the butterfly wing images and extracted 8,100-dimensional HOG feature vectors from each image. In addition, we introduced LBP to extract local texture information. To preserve more local details, we divided the original 512 × 512-pixel images into 64 × 64-pixel spatial regions and then calculated uniform LBP histograms for each cell (Figure 5). Specifically, a one-radius circular pattern was used to compute the LBP of eight neighbors for each pixel, and rotationally invariant features were not considered. Then, L2 normalization was applied to each cell's LBP histogram. As a result, 3,776-dimensional LBP feature vectors were extracted from each image of the spider dataset, and both HOG and LBP descriptors were combined into 11876-dimensional feature vectors for spider image classification. As with the butterfly data, we applied the SVM model with the linear kernel function to those high-dimensional feature vectors. By using 20 replicates of 10-fold cross-validations, we calculated accuracy, precision, recall, and F1 scores that helped us evaluate the jumping spider classifier's performance (Table 2). With data augmentation, the overall accuracy of the jumping spider classifier was over 94.3% and, as it was with the butterfly classifier, jumping spider Morph 1, which had the smallest sample size, also had a relatively low recall value (83.4%). These results suggest that as training sample size increases, classification performance should improve. The macro-averaged ROC curve showed a considerable predictive ability for spider classification, comparable to that of the butterfly classification model performance (Figure 6).

Finally, the preprocessed spider test data were classified using the trained classifier and labeled accordingly (Figure 7b). The feature extraction and classification process for the spider dataset with 11876-dimensional feature vectors took approximately 0.04 s per image to compute. Our results showed that the spider image data were precisely classified and assigned to the abovementioned morphs, thus suggesting that these spiders could be classified according to their body textures and without any intermediate morph.

### 9.3 | Performance of SVMorph and other methods

To evaluate the performance of SVMorph, we compared the accuracy and running time for SVMorph and other methods including the morphology-based taxonomy and three established deep learning (Convolutional Neural Network, CNN)-based models such as AlexNet (Krizhevsky et al., 2012), GoogLeNet (Szegedy et al., 2015), and VGG

(Simonyan & Zisserman, 2014) (Table S1). All the methods were tested using both butterfly and spider datasets on one Intel i7-8700 (3.20G Hz) CPU with 16 GB RAM. We also tested three CNN-based methods on a Quadro P400 GPU, but the VGG-16 did not complete the task owing to memory overflow, indicating a limitation of GPU usage on such scenarios. For each CNN-based method, the accuracy was the mean value of ten separate runs to overcome the randomness of CNN model training, whereas SVMorph yielded consistent results. For each method, the time was the mean value of ten separate runs. For the butterfly dataset, we have not established a reliable morphology-based taxonomy, whereas for the spider dataset, we quantified the spider body textures using a scanning electron microscope (SEM). The SEM images showed that the different body textures were due to different cuticle structures and seta lengths. We collected six individuals for each phenotype and measured three setae for each individual, which took about two days. Relative to the manual morphology-based taxonomy, all the CNN-based methods and SVMorph cost less time. However, one of the CNN-based methods showed relatively lower accuracy in analyzing both datasets. The SVMorph method performed better with considerably high accuracy and the least amount of running time when handling the two datasets, suggesting its application potential on specific and complex classification tasks, and the deep learning-based algorithms focus on the generalization ability instead of specificity.

## 10 | DISCUSSION

The scales and setae of invertebrates are single-cellular cuticular structures extending from the cuticle, which play an important role in generating phenotypic diversity and can have a profound impact on fitness (Tian et al., 2019). Taken the two sample datasets as an example, the dead-leaf butterflies in *K. inachus* display a remarkable diversity of leaf wing patterns for masquerade (Protas & Patel, 2008), whereas different morphs of *Toxeus* jumping spiders mimic different ant species to gain a protective advantage (Yamasaki, 2015). These cuticular structures accommodate not only color variation but also shape variation such as length, width, and diameter of a cross section, and pilosity variation (Matsuoka & Monteiro, 2018). Therefore, the complexity in cuticular structures makes it difficult to classify the phenotypes and understand the underlying mechanisms of phenotypic diversity. To characterize the wing coloration of butterflies, one may measure wavelength reflectance in at least three locations: the main vein, lateral veins, and spots. To quantify the texture patterns, one may need to place setae on scanning electron microscopy (SEM) stubs, sputter-coat them with gold–palladium, and measure their length and width under SEM. However, it is difficult and inefficient to apply these quantifying methods to large-scale datasets. SVMorph is a convenient and efficient method for identifying variations of cuticular structures such as setae and scales. The method obtained in this study can be used for analyzing wing patterns and body textures, which could fully develop identification systems of phenotypic polymorphism for such nonmodel organisms and bridge the gap between genotype and phenotype. Furthermore, due to the generality of the proposed method, it can be used with no major modification for other tasks, such as analyzing cuticular structures in other invertebrates.

## 11 | CONCLUSIONS

SVMorph, a fast and accurate pipeline for handling classification tasks with complex patterns and textures, is particularly helpful for batch processing large image data applications. Also, based on the performance of a trained classifier, it can be used to examine a priori hypotheses of organismal classifications or to extract and investigate images with rare morphological features that cannot be appropriately labeled. Moreover, SVMorph is very modular and easy to schedule, thus giving the researcher the ability to perform each step independently and interactively with other applications. In summary, SVMorph efficiently characterizes and classifies morphological characters of nonmodel organisms.

### ACKNOWLEDGMENTS

We thank Qingyi Zhou and Yubo Zhang for helpful discussion. We thank the National Center for Protein Sciences at Peking University for assistance with SEM imaging. This project was supported by grants from the Beijing Natural Science Foundation (JQ19021 to WZ), the National Natural Science Foundation of China (31871271 to WZ), the Peking-Tsinghua Center for Life Science to WZ, the State Key Laboratory of Protein and Plant Gene Research and Qidong-SLS Innovation Fund to WZ, and the China Postdoctoral Science Foundation (2020M680214 to FL). The authors declare that they have no competing interests.

### CONFLICT OF INTEREST

None declared.

### AUTHOR CONTRIBUTION

**Dequn Teng:** Conceptualization (supporting); Data curation (equal); Formal analysis (lead); Methodology (lead); Resources (lead); Software (lead); Validation (equal); Visualization (lead); Writing-original draft (equal); Writing-review & editing (supporting). **Fengyuan Li:** Data curation (equal); Formal analysis (supporting); Funding acquisition (supporting); Resources (supporting); Software (supporting); Validation (equal); Visualization (supporting); Writing-original draft (supporting); Writing-review & editing (supporting). **Wei Zhang:** Conceptualization (lead); Data curation (equal); Formal analysis (supporting); Funding acquisition (lead); Investigation (lead); Methodology (equal); Project administration (lead); Resources (equal); Software (supporting); Supervision (lead); Validation (equal); Visualization (equal); Writing-original draft (lead); Writing-review & editing (lead).

### OPEN RESEARCH BADGES

This article has been awarded Open Data, Open Materials Badges. All materials and data are publicly accessible via the Open Science Framework at https://github.com/TDQ233/SVMorph.

## DATA AVAILABILITY STATEMENT

SVMorph is available at GitHub (https://github.com/TDQ233/SVMorph) and Dryad (https://doi.org/10.5061/dryad.tdz08kq0g).

## ORCID

*Wei Zhang* https://orcid.org/0000-0002-6644-7046

## REFERENCES

Allen, W. L., & Higham, J. P. (2015). Assessing the potential information content of multicomponent visual signals: a machine learning approach. *Proceedings of the Royal Society B: Biological Sciences*, *282*(1802), 20142284. https://doi.org/10.1098/rspb.2014.2284

Baraldi, A., & Panniggiani, F. (1995). An investigation of the textural characteristics associated with gray level cooccurrence matrix statistical parameters. *IEEE Transactions on Geoscience and Remote Sensing*, *33*, 293–304. https://doi.org/10.1109/TGRS.1995.8746010

Bookstein, F. L. (1996). Biometrics, biomathematics and the morphometric synthesis. *Bulletin of Mathematical Biology*, *58*, 313–365. https://doi.org/10.1007/BF02458311

Cavicchi, S., Pezzoli, C., & Giorgi, G. (1981). Correlation between characters as related to developmental pattern in *Drosophila*. *Genetica*, *56*, 189–195. https://doi.org/10.1007/BF00057559

Crnojević, V., Panić, M., Brkljač, B., Ćulibrk, D., Ačanski, J., & Vujić, A. (2014). (2014) Image processing method for automatic discrimination of hoverfly species. *Mathematical Problems in Engineering*, *2014*, 1–12. https://doi.org/10.1155/2014/986271

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *1*, 886–893.

Dao, T., Gu, A., Ratner, A. J., Smith, V., De Sa, C., & Re, C. (2019). A kernel theory of modern data augmentation. *Proceedings of Machine Learning Research*, *97*, 1528–1537.

Dietterich, T. G., & Bakiri, G. (1994). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, *2*, 263–286. https://doi.org/10.1613/jair.105

Dobens, A. C., & Dobens, L. L. (2013). FijiWings: An open source toolkit for semiautomated morphometric analysis of insect wings. *G3 Genes|genomes|genetics*, *3*, 1443–1449. https://doi.org/10.1534/g3.113.006676

Grant, P. R., & Grant, B. R. (2006). Evolution of character displacement in Darwin's finches. *Science*, *313*, 224–226. https://doi.org/10.1126/science.1128374

Guerra, D., Pezzoli, M. C., Giorgi, G., Garoia, F., & Cavicchi, S. (1997). Developmental constraints in the *Drosophila* wing. *Heredity (Edinb)*, *79*(Pt 6), 564–571. https://doi.org/10.1038/hdy.1997.200

Houle, D., Mezey, J., Galpern, P., & Carter, A. (2003). Automated measurement of *Drosophila* wings. *BMC Evolutionary Biology*, *3*, 25.

Joron, M., Jiggins, C. D., Papanicolaou, A., & McMillan, W. O. (2006). *Heliconius* wing patterns: An evo-devo model for understanding phenotypic diversity. *Heredity (Edinb)*, *97*, 157–167. https://doi.org/10.1038/sj.hdy.6800873

Klingenberg, C. P. (2011). MorphoJ: An integrated software package for geometric morphometrics. *Molecular Ecology Resources*, *11*, 353–357. https://doi.org/10.1111/j.1755-0998.2010.02924.x

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, *25*, 1097–1105.

Le Poul, Y., Whibley, A., Chouteau, M., Prunier, F., Llaurens, V., & Joron, M. (2014). Evolution of dominance mechanisms at a butterfly mimicry supergene. *Nature Communications*, *5*, 5644. https://doi.org/10.1038/ncomms6644

Loh, S. Y. M., Ogawa, Y., Kawana, S., Tamura, K., & Lee, H. K. (2017). Semi-automated quantitative *Drosophila* wings measurements. *BMC Bioinformatics*, *18*, 319. https://doi.org/10.1186/s12859-017-1720-y

Lürig, M., Donoughe, S., Svensson, E. I., Porto, A., & Tsuboi, M. (2021). Computer vision, machine learning, and the promise of phenomics in ecology and evolutionary biology. *Frontiers in Ecology and Evolution*.

Matsuoka, Y., & Monteiro, A. (2018). Melanin pathway genes regulate color and morphology of butterfly wing scales. *Cell Reports*, *24*, 56–65. https://doi.org/10.1016/j.celrep.2018.05.092

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., & Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Peichel, C. L., Nereng, K. S., Ohgi, K. A., Cole, B. L., Colosimo, P. F., Buerkle, C. A., Schluter, D., & Kingsley, D. M. (2001). The genetic architecture of divergence between threespine stickleback species. *Nature*, *414*, 901–905. https://doi.org/10.1038/414901a

Protas, M. E., & Patel, N. H. (2008). Evolution of coloration patterns. *Annual Review of Cell and Developmental Biology*, *24*, 425–446. https://doi.org/10.1146/annurev.cellbio.24.110707.175302

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Suzuki, T. K., Tomita, S., & Sezutsu, H. (2019). Multicomponent structures in camouflage and mimicry in butterfly wing patterns. *Journal of Morphology*, *280*, 149–166. https://doi.org/10.1002/jmor.20927

Szegedy, C., Liu, W., Jia, Y. Q., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (Cvpr)*, *2015*, 1–9.

Tang, Y. C., Zhang, Y. Q., Chawla, N. V., & Krasser, S. (2009). SVMs modeling for highly imbalanced classification. *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*, *39*, 281–288. https://doi.org/10.1109/TSMCB.2008.2002909

Tian, L., Rahman, S. R., Ezray, B. D., Franzini, L., Strange, J. P., Lhomme, P., & Hines, H. M. (2019). A homeotic shift late in development drives mimetic color variation in a bumble bee. *Proceedings of the National Academy of Sciences of the United States of America*, *116*, 11857–11865. https://doi.org/10.1073/pnas.1900365116

Van Belleghem, S. M., Papa, R., Ortiz-Zuazaga, H., Hendrickx, F., Jiggins, C. D., McMillan, W. O., & Counterman, B. A. (2018). patternize: An R package for quantifying colour pattern variation. *Methods in Ecology and Evolution*, *9*, 390–398. https://doi.org/10.1111/2041-210X.12853

Yamasaki, T. (2015). Studies on taxonomy, biogeography and mimicry of the genus *Myrmarachne* in Southeast Asia. *Acta Arachnologica*, *64*, 49–56.

Yamasaki, T., & Ahmad, A. H. (2013). Taxonomic study of the genus *Myrmarachne* of Borneo (Araneae: Salticidae). *Zootaxa*, *3710*, 501–556.

Yang, H. P., Ma, C. S., Wen, H., Zhan, Q. B., & Wang, X. L. (2015). A tool for developing an automatic insect identification system based on wing outlines. *Scientific Reports*, *5*, 12786. https://doi.org/10.1038/srep12786

## SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section.

**How to cite this article:** Teng, D., Li, F., & Zhang, W. (2021). Using comprehensive machine-learning models to classify complex morphological characters. *Ecology and Evolution*, 11, 10421–10431. https://doi.org/10.1002/ece3.7845