

OPEN

A shell dataset, for shell features extraction and recognition

DATA DESCRIPTOR

Qi Zhang¹, Jianhang Zhou¹, Jing He², Xiaodong Cun¹, Shaoning Zeng^{1,3} & Bob Zhang ^{1*}

Shells are very common objects in the world, often used for decorations, collections, academic research, etc. With tens of thousands of species, shells are not easy to identify manually. Until now, no one has proposed the recognition of shells using machine learning techniques. We initially present a shell dataset, containing 7894 shell species with 29622 samples, where totally 59244 shell images for shell features extraction and recognition are used. Three features of shells, namely colour, shape and texture were generated from 134 shell species with 10 samples, which were then validated by two different classifiers: *k*-nearest neighbours (*k*-NN) and random forest. Since the development of conchology is mature, we believe this dataset can represent a valuable resource for automatic shell recognition. The extracted features of shells are also useful in developing and optimizing new machine learning techniques. Furthermore, we hope more researchers can present new methods to extract shell features and develop new classifiers based on this dataset, in order to improve the recognition performance of shell species.

Background & Summary

In human history the utilization of shells has occurred for thousands of years. The cowrie shells are commonly found in Bronze Age sites in ancient China, and usually regarded as money or currency during the Shang and Zhou periods¹. In Western Europe, the Sowerby family was active and presented numerous works on molluscs, and its systematics from the late eighteenth century to mid twentieth century. According to statistics, the Sowerby family introduced the names of more than 2000 shell species and produced many books on the genera of shells^{2,3}.

Today shell collection and the development of conchology are on the uprise. The Sanibel Shell Festival has been held consecutively more than 70 years⁴. Meanwhile, many academic books or journals about shell research and classification have recently gained more popularity. A publisher called ConchBooks that specializes in shell research, has published more than 3000 books about shell research (<https://www.conchbooks.de/?t=1>).

Although there are so many works on shell collection and identification, it is still difficult to recognise shell species manually, as shells have tens of thousands of classes^{5,6}. Thus, this problem indeed hampers the passion of shell collecting amateurs and the development of conchology. With the growth of the Internet and the progress of artificial intelligence⁷, it is possible and useful to investigate shell classification using machine learning techniques.

In this article we present a large shell dataset, containing 7894 shell species with 59244 shell images. Each species has shell samples ranging from 1 to 87 respectively, and every shell sample has two photographs taken at different views: frontal and lateral by us (Fig. 1). As different shells have different colours, shapes and decorative patterns, which can be used to identify shell species by artificial intelligence, three shell features: colour, shape and texture were generated from shell sample images by some image processing methods. Two classifiers: *k*-NN⁸ and random forest⁹ were applied for evaluating the extracted shell features. Preliminary experiments in the technical validation section with positive results show the potential and effective capability of using this data successfully in automatic shell recognition.

We hope more researchers especially computer scientists attempt to re-use this shell dataset, propose novel feature extraction methods or new classification methods to improve the performance of shell recognition. Since this work just extracts three common features of a shell, some special features such as geometric patterns are not investigated¹⁰. The extracted features from shells are also useful for developing and optimizing new machine learning techniques. Due to the fact that only two simple classifiers: *k*-NN and random forest were used in this

¹Department of Computer and Information Science, University of Macau, Taipa, Macau, China. ²Citizen Scientist, Sanxin Road (N) 1755/117, Songjiang, Shanghai, China. ³School of Information Science and Technology, Huizhou University, Avenue Yanda, No. 46, Huizhou, Guangdong, China. *email: bobzhang@um.edu.mo

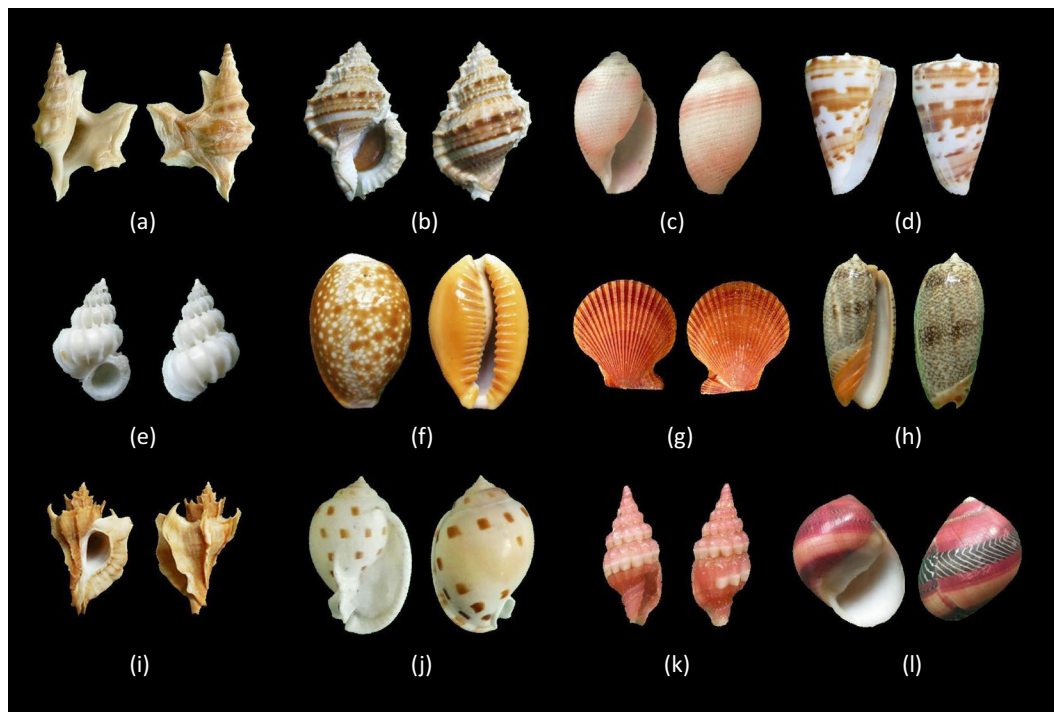


Fig. 1 A small variety of shell species that are part of this shell dataset. From top to bottom are (a) *Aporrhais pespelicani*, (b) *Bufonaria nana*, (c) *Bullina virgo*, (d) *Conus advertex*, (e) *Epitonium tokyoense*, (f) *Erosaria helvola*, (g) *Mimachlamys asperima*, (h) *Oliva reticulata*, (i) *Pteropurpura adunca*, (j) *Semicassis bisulcate*, (k) *Vexillum rubrum*, (l) *Vittina waigiensis*. Each shell sample contains two images of the frontal and lateral, and all shell samples are organized carefully with the photos taken based on this rule.

article to evaluate the shell classification performance, many state-of-the-art machine learning methods such as convolutional neural network^{11,12} may be reasonable ways to access shell classification results in the future.

Methods

In this work we first collected and reorganized the shell images. Afterwards, three shell features were extracted from the shell dataset by applying some image processing methods. These extracted features will be validated by two classifiers in the technical validation section to prove the quality of this shell dataset. Fig. 2 shows the procedures of generating shell features in this dataset.

Data Pre-processing. Our shell data collection contains 7894 shell species with 29622 samples, where each sample has two different views of colour images (JPG format). Each shell image was labelled with its scientific name and corresponding number, then resized to 300*400 pixels to be further processed to generate its features.

Colour feature extraction. Since colour feature extraction from shells has not been investigated before, we can refer to some leaf and flower recognition works, as leaf, flower and shell have different colours on their surface. Caglayan *et al.* applied three histograms to the red, green and blue channels of leaf images, before calculating the mean and standard deviation in each histogram as the colour features for classifying the *Flavia* dataset¹³. Mishra *et al.* used a RGB histogram to calculate the redness index, greenness index and blueness index values for identifying digital leaf and flower images¹⁴. Thus, we can extract colour based features from shell images similarly for shell classification. In this study, we generated colour features from a colour histogram¹⁵ in the red, green and blue channels of a shell image (Fig. 3). For one shell sample, there are two colour images taken at different views. Therefore, we can generate two 256*3 (where 256 represents the number of grayscales ranging from 0 to 255) matrixes for the first shell image and second shell image respectively, then combine them together to construct a 256*6 matrix. The black background colour in a shell image was analysed by the flood fill algorithm¹⁶, which generated a corresponding black background mask for each shell image. Therefore, the corresponding black background colour in a shell image can be eliminated by calculating the number of black background mask pixels, which maintains the effectiveness of the extracted shell colour feature. The mean (μ) and standard deviation (s) of the intensity values for the red, green and blue channels were calculated from this generated 256*6 matrix. Thus, 12 elements were used for delegating the shell colour feature (each colour channel would generate two values: mean and standard deviation respectively). Therefore, we are able to classify shell species by using this extracted data effectively.

Shape feature extraction. Numerous works can be found in literature discussing the extraction of shape features in plant leaf recognition. One of them is the widely used Centroid Contour Distance (CCD)¹⁷. This

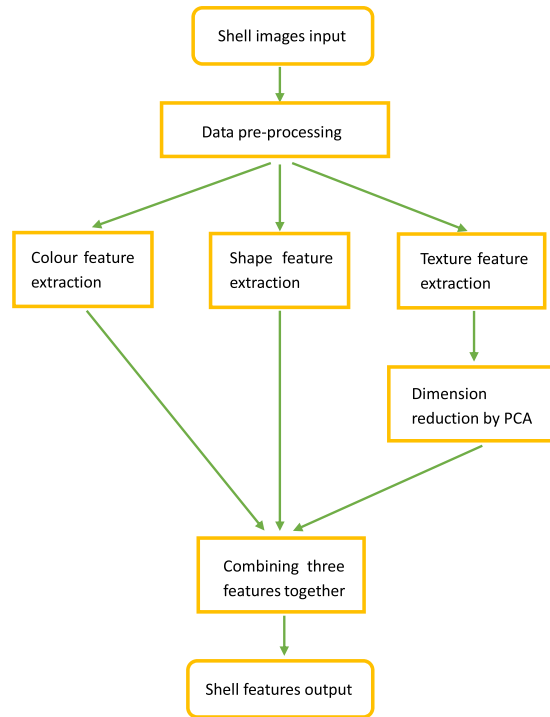


Fig. 2 The flow diagram of generating shell features.

method is able to find out the distance between the centroid point and the boundary point, which is useful to extract the targeted object outside boundary.

CCD can trace a targeted object contour by circling around its centroid (Fig. 4). The midpoint C can be regarded as the object's centroid, while P is one of the points on the boundary. The distance between the point P and the central point C is considered as the centroid-distance. When point P moves on the boundary of the object based on the angle α , the centroid-distance also changes. We can collect the various centroid-distances of one object based on the different angles, which are treated as the shape feature.

In this article, we used the CCD method to extract the shape feature of shell images. The shell image was converted to grayscale, where we applied a flood fill algorithm¹⁶ to detect the black background part and generate to corresponding a background mask. The black background mask was inverted to generate a targeted shell mask of each shell image. Therefore, the boundary of a shell can be obtained from the shell mask and further processed by the CCD method. In Fig. 5, the central point (red point) can be calculated from the shell's boundary (<https://www.mathworks.com/help/matlab/ref/polyshape.centroid.html>). And the point P (blue point from 1 to 72) on the shell's boundary from 0° moves to 360° every step by 5° (using an interval angle of $\alpha = 5^\circ$, totally $360/5 = 72$ steps), before calculating the distance between the central point and boundary point in every step, finally generating 72 distance points for one shell image.

Texture feature extraction. Different shells have different decorative patterns on their surfaces, which can be considered as its texture feature in shell classification. The Gabor filter¹⁸ is a linear filter widely used in texture analysis and recognition, showing its potential performance¹⁹. In this section, a 2-D Gabor filter is applied to the grayscale shell images to generate its texture feature, which is given by Eqs (1) and (2):

$$f(x, y, \omega, \sigma_x, \sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left\{-\frac{1}{2}\left[\left(\frac{x}{\sigma_x}\right)^2 + \left(\frac{y}{\sigma_y}\right)^2 + j\omega(x\cos\theta + y\sin\theta)\right]\right\} \quad (1)$$

$$r(x, y) = I(x, y) \times f(x, y, \omega, \sigma_x, \sigma_y) \quad (2)$$

where σ is the spatial width, ω is the frequency, and θ is the orientation. The different orientations (θ) and frequencies (ω) are key parameters in texture analysis and extraction. The $I(x, y)$ in Eq. (2) indicates the grayscale image of a shell texture, $f(x, y, \omega, \sigma_x, \sigma_y)$ is the Gabor filter with different settings in the frequencies and orientations, and $r(x, y)$ denotes the image filtered results by the Gabor filter.

There are different justifications for the choice of frequency (ω) and orientation (θ). Jain *et al.*¹⁹, used only four orientations ($0^\circ, 45^\circ, 90^\circ, 135^\circ$) to reduce the computational cost, and selected frequencies based on psy-chophysical studies. The previous work by Recio *et al.*²⁰, applied a set of frequencies and orientations which are determined empirically. In Cope *et al.*'s work²¹, the frequencies of the Gabor filter were chosen from 0 to 7. Based on the work

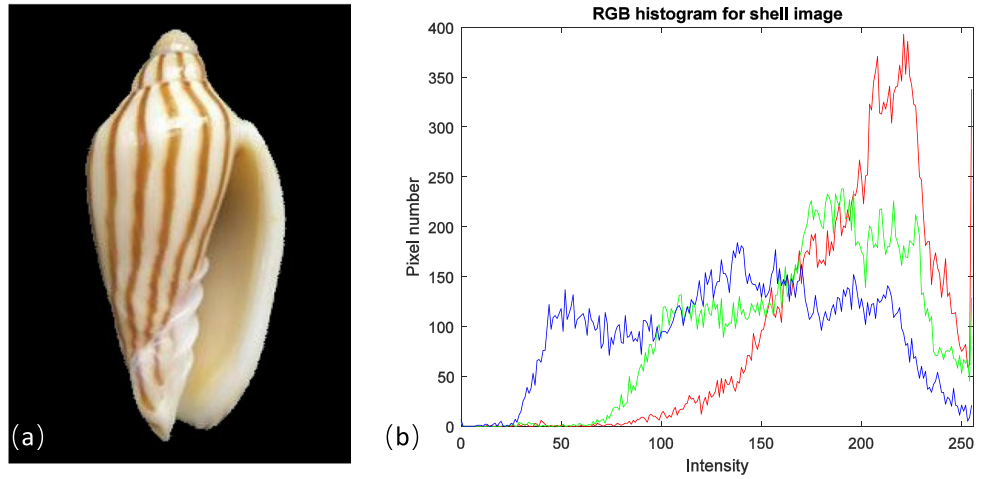


Fig. 3 The RGB histogram generated from a shell image. Figure (a) indicates a colour shell image (scientific name: *Amoria dampieria*, front view, image size: 300*400, ID number in shell dataset: *Amoria_dampieria_10_A*), while plot (b) shows its corresponding colour histograms (red, green and blue curves) for the red, green and blue channels of this shell.

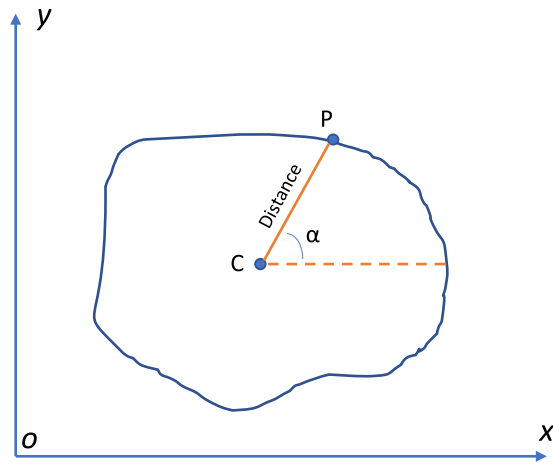


Fig. 4 The principle of centroid contour distance.

of others and our many experiments, four different orientations ($0^\circ, 45^\circ, 90^\circ, 135^\circ$) with five different frequencies ($\omega = 5, 10, 15, 20, 25$) were chosen for the Gabor filter settings. Thus, we totally have 20 different Gabor filters to analyse the shell texture. As the computational cost and time for extracting the texture feature by applying Gabor filters is very high, we choose the first image (frontal view) of all shell samples for texture analysis. For each shell sample, 200 small patches (each patch is sized 20*20 pixel) from the surface of the shell were randomly selected, then transformed to greyscale images. Next, 20 different Gabor filters were applied to these 200 small patches in one shell sample, generating 4000 responses. Each response was then calculated to produce 3 features via the following equations:

$$\text{Average value: } \sum_{(i,j) \in W} \frac{r_{ij}}{|W|} \tag{3}$$

$$\text{Energy: } \sum_{(i,j) \in W} \frac{r_{ij}^2}{|W|} \tag{4}$$

$$\text{Entropy: } - \sum_{(i,j) \in W} \frac{|r_{ij}|}{|W|} \log \frac{|r_{ij}|}{|W|} \tag{5}$$

where W is the current patch, r_{ij} is the response for the current filter at pixel (i, j) , and $|W|$ is the number of pixels in each small patch.

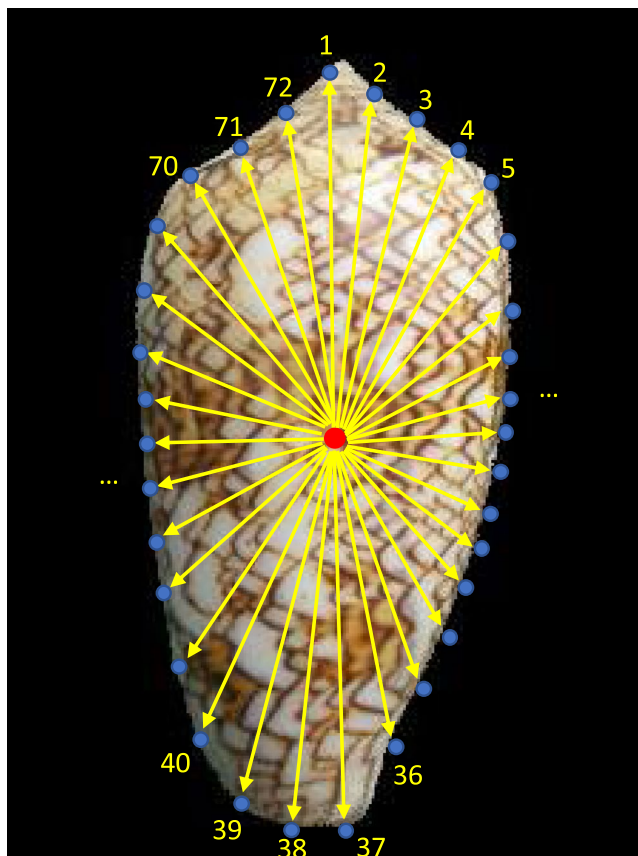


Fig. 5 The number of boundary points based on an interval angle of $\alpha = 5^\circ$. The red point is this shell image's central point, the blue points are obtained by using an interval angle of $\alpha = 5^\circ$, and the distances between the central point and boundary points are calculated and regarded as the shape feature of a shell.

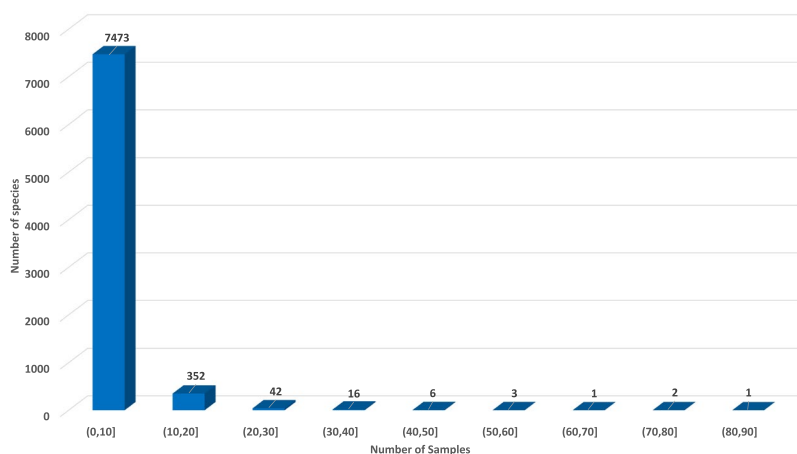


Fig. 6 The distribution of sample numbers of all shell species. This dataset is a highly imbalanced shell data with 7894 species. Most of the shell species have less than 10 samples, while a few of them have over 30 samples.

After applying the aforementioned texture extractors, each shell sample forms a 4000×3 matrix. Here, the values of each row of this matrix were arranged from small to large, which is regarded as texture feature. As the size of the preliminary texture feature is too large to deal with, the principal component analysis (PCA) was applied to reduce the dimension of the shell texture matrix while preserving the features that contribute most to the variance in this dataset²². As over 95% of the variance in the dataset come from the first ten projected features, we utilize them as the texture feature. Hence, each shell sample texture is quantized by PCA to a 10 elements predetermined vector, which is considered as the final texture feature of a shell.

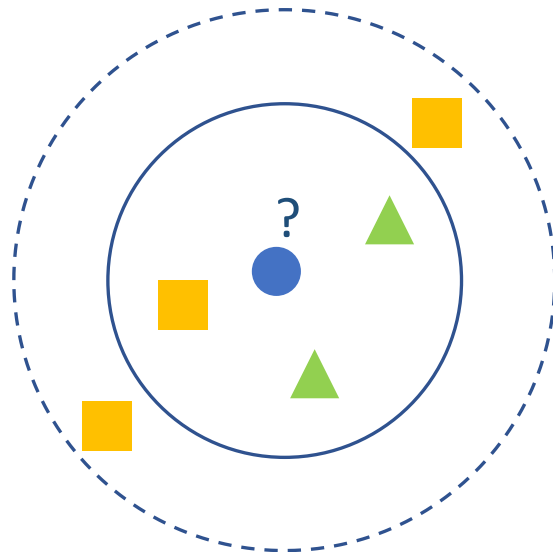


Fig. 7 The illustration of k -NN classification. The test sample (blue circle) would be categorized to the first class of green triangles when $k=3$ (solid line circle), as there are 2 green triangles and only 1 orange square inside the inner circle. However, it would be assigned to the second class of orange squares when $k=5$ (dashed line circle), since there are 3 orange squares and only 2 green triangles inside the outer circle.

Data Records

A shell statistics figure has been plotted to show the shell sample distribution of the entire dataset (Fig. 6). The reorganized shell images dataset contains 29622 samples, where the complete 59244 images are available from the file `all_shell_images_2nd.zip` and can be downloaded. The sample number of each shell species is recorded as the file `all_shell_species_inventory_revised.xlsx` and is also available for download. All three features of the shell samples are extracted and recorded as files `all_color_features`, `all_shape_features`, and `all_texture_features` respectively, which are also available for download. 134 shell species' images are analysed by three (colour, shape and texture) feature extraction methods, which is available as the file `shell_species_134_data.zip`. The colour feature extracted by the aforementioned colour feature extraction method (refer to Method: Colour feature extraction section), is also available as the file `colour_feature_raw.xlsx`, where post-processing by calculating the mean (μ) and standard deviation (s) can be found in the file `colour_feature_processing.xlsx`. Both are available for download. The shape feature extracted via the shape feature extraction method (see Method: Shape feature extraction section), is also available as the file `shape_feature.xlsx` ready for download. The texture feature extracted using the texture feature extraction method (refer to Method: Texture feature extraction section), is accessible as the file `texture_feature_raw.xlsx` along with its post-processing by PCA as the file `texture_feature_processing.xlsx`. Both are available for download. The complete shell dataset is openly available at the figshare repository²³.

Technical Validation

Shell database. In order to prove the effectiveness and potentiality of the shell dataset, the extracted shell features were applied by two different classifiers: k -NN and random forest for shell recognition. It could be noted that this shell dataset is strongly unbalanced in terms of samples per shell species in Fig. 6. Thus, we choose all shell species (totally 134 species) with 10 samples (Online-only Table 1) in this study to validate the fairness and effectiveness of extracted features of this shell dataset. Afterwards, a total of 1340 samples were chosen for validation. The F1-score and accuracy of shell recognition is 78.23% and 77.39% respectively, when applying k -NN with the three features. Therefore, this proposed dataset can be considered as a robust and effective approach for shell species classification.

k -nearest neighbours. k -NN is a commonly used supervised learning method. Its working principle is very simple: it attempts to find the nearest k training samples in the training dataset based on a distance measurement, allowing it to predict the results using the information of these k neighbored samples in a testing dataset. The voting method usually can be applied in classification tasks, which generally chooses the most class markers in the k training samples as the prediction result⁸. Fig. 7 shows the schematic diagram of k -NN, where it is obvious to see that the k value is an important parameter. The classification results would be significantly different with different k values settings. In this validation, we investigated different k values to assess their performances for shell recognition and found the best classification results using the k -NN method.

Random forest. The random forest is a classifier containing multiple decision trees in the training dataset, and its output class is determined by the mode of the classes of the individual trees⁹. It is based on a decision tree with bagging, which is further introduced with random attribute selection during the training process. Specifically, it randomly selects a subset containing k attributes for each node of the sub-decision trees, before

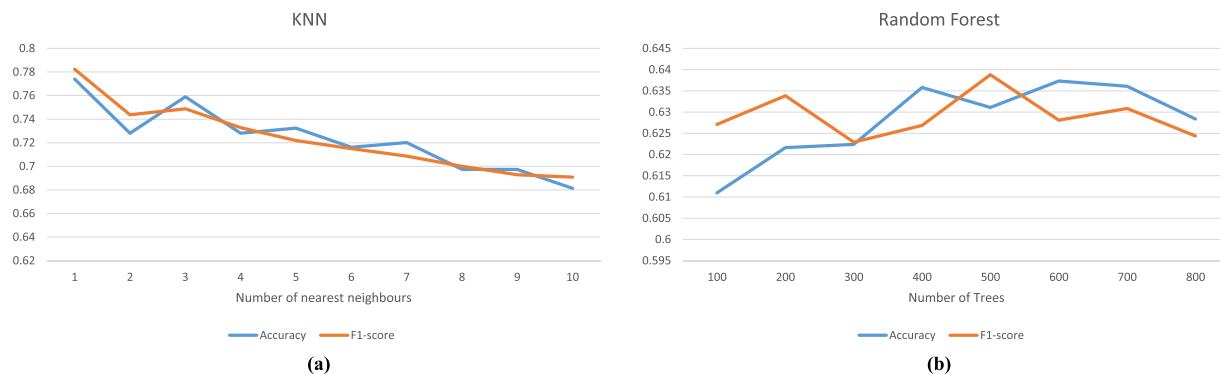


Fig. 8 The performance of different k values and tree numbers in k -NN and random forest classification. The performance of different k values and tree numbers in k -NN and random forest classification. (a) presents the accuracy of shell recognition related to the k nearest neighbour number for k -NN using three features, while (b) shows the accuracy of shell recognition related to the tree number in random forest when using three features.

Classifiers	F1-score (%)	Accuracy (%)
k -NN ($k=1$)	50.35 ± 0.0149	50.87 ± 0.0132
Random forest	34.88 ± 0.0079	36.09 ± 0.0126

Table 1. Shell classification performance using colour feature.

Classifiers	F1-score (%)	Accuracy (%)
k -NN ($k=1$)	67.66 ± 0.0130	66.71 ± 0.0134
Random forest	59.32 ± 0.0115	58.11 ± 0.0116

Table 2. Shell classification performance using shape feature.

Classifiers	F1-score (%)	Accuracy (%)
k -NN ($k=1$)	15.17 ± 0.0113	16.39 ± 0.0090
Random forest	15.92 ± 0.0094	17.85 ± 0.0082

Table 3. Shell classification performance using texture feature.

Classifiers	F1-score (%)	Accuracy (%)
k -NN ($k=1$)	78.23 ± 0.0119	77.39 ± 0.0107
Random forest	62.81 ± 0.0126	63.73 ± 0.0098

Table 4. Shell classification performance using 3 features.

choosing the best attribute from this subset for partitioning. Random forest is a simple, easy to implement and low computational method, showing good performances in many practical tasks. Here, we implemented random forest for shell recognition to evaluate the usability of this dataset.

F1-score. To take the disparity of the samples in each class into account, we use an additional F1-score metric to perform evaluation²⁴. Since F1-score is a compound of precision and recall, we utilize it as a comprehensive metric to provide performance evaluation.

The F1-score is described as follows:

$$F1 - score = \frac{2\sum_{i=1}^n TP_i}{2\sum_{i=1}^n TP_i + \sum_{i=1}^n FP_i + \sum_{i=1}^n FN_i} \quad (6)$$

where n represents the number of classes, and TP_i , TP_i , FP_i , FN_i are true positive, true negative, false positive and false negative of the i th class, respectively.

Experiments. The two introduced classifiers were applied to evaluate this shell dataset. In particular, each extracted feature from a shell was first assessed individually. The parameters in the classifiers such as k were fine-tuned in order to obtain the best results. This process was repeated 30 times randomly, and the average of the 30 runs was used as final results. Next, three features from a shell were combined together to construct a vector with 166 dimensions, which was also evaluated by k -NN and random forest. The procedures of parameter selection, the proportion (70%) of the training dataset and the number of repeating times are the same to the individual extracted feature.

Experimental results. Tables 1–4 shows the F1-score and classification accuracy results of using shell colour feature, shape feature, texture feature and the combination of the three features with a confidence level ($\alpha = 95\%$) respectively. It should be noted that the shape feature is the most effective characteristic in shell classification, followed by the colour and texture features. When combining the three features, the final result is much better than any single feature, proving the validity of this shell dataset. Figure 8(a) shows the F1-score and accuracy related to the k value selection for k -NN (when three features are used simultaneously). The highest performance is reached when k is 1, which is regarded as the final classification result. In Fig. 8(b), F1-score and accuracy do not achieve their highest value with the same parameter using random forest. Here, we select the parameter with the highest accuracy, which is $T = 600$ as the final classification result. The F1-score and accuracy of shell recognition is 78.23% and 77.39% respectively, when applying k -NN with the three features (Table 4). Therefore, this proposed dataset can be considered as a robust and effective approach for shell species classification.

Usage Notes

We provide the code of extracting the three features for shell images (can be found in the Code availability section). In addition, the features of 134 shell species with 10 samples were extracted and analysed by two classifiers k -NN and random forest in the technical validation section, which can be found in github: https://github.com/zqplus/shell-recognition/blob/master/ReadMe_how%20to%20generate%20shell%20features%20%26%20load%20data%20for%20classification/Shell_env. Researchers can directly use post-processing shell features data to find more effective machine learning methods for improving the performance of shell recognition, or attempt to investigate new algorithms to deal with shell species with small samples, even present new feature extraction methods for shell recognition based on the collected shell images.

Code availability

The code for extracting the three features from a shell can be found here: https://github.com/zqplus/shell-recognition/tree/master/ReadMe_how%20to%20generate%20shell%20features%20%26%20load%20data%20for%20classification.

Received: 11 March 2019; Accepted: 17 September 2019;

Published online: 22 October 2019

References

- Li, Y. T. On the Function of Cowries in Shang and Western Zhou China. *Journal of East Asian Archaeology*. **5**, 1–26 (2003).
- Sowerby, G. B. *Thesaurus Conchyliorum Or Monographs of Genera of Shells*. (London: Sowerby, 45, Great Russell Street, Bloomsbury, 1866).
- Petit, R. E. George Brettingham Sowerby, I, II, III: their conchological publications and molluscan taxa. **2189**, 1–218 (2009).
- Schriner, H. *Sanibel Shell Show History*, <https://sanibelsHELLclub.com/sanibel-shell-show-history/> (2019).
- Abbott, R. T., Dance, S. P. & Abbott, T. *Compendium of seashells*. (New York: EP Dutton, 1983).
- Lorenz, F. *Cowries: A guide to the gastropod family Cypræidae*. Volume 1: Biology and systematics (Hackenheim: Conchbooks, 2017).
- Russell, S. J. & Norvig, P. *Artificial intelligence: a modern approach*. (Pearson Education Limited, 2016).
- Altman, N. S. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*. **46**, 175–185 (1992).
- Breiman, L. Random forests. *Machine learning*. **45**, 5–32 (2001).
- Raup, D. M. Geometric analysis of shell coiling: general problems. *Journal of Paleontology*. 1178–1190 (1966).
- Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*. 1097–1105 (2012).
- LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature*. **521**, 436–444 (2015).
- Caglayan, A., Guclu, O. & Can, A. B. A plant recognition approach using shape and color features in leaf images. *International Conference on Image Analysis and Processing*. 161–170 (2013).
- Mishra, P. K., Maurya, S. K., Singh, R. K. & Misra, A. K. A semi automatic plant identification based on digital leaf and flower images. *International Conference on Advances In Engineering, Science And Management*. 68–73 (2012).
- Shapiro, L. & Stockman, G. C. *Computer vision*. (Prentice Hall, 2001).
- Soille, P. *Morphological image analysis: principles and applications*. (Springer Science & Business Media, 2013).
- Hasim, A., Herdiyeni, Y. & Douady, S. Leaf shape recognition using centroid contour distance. *IOP conference series: earth and environmental science* **31**, 012002 (2016).
- Grigorescu, S. E., Petkov, N. & Kruizinga, P. Comparison of texture features based on Gabor filters. *IEEE Transactions on Image processing*. **11**, 1160–1167 (2002).
- Jain, A. K. & Farrokhnia, F. Unsupervised texture segmentation using Gabor filters. *Pattern recognition*. **24**, 1167–1186 (1991).
- Recio, J. A. R., Fernández, L. A. R. & Fernández-Sarriá, A. Use of Gabor filters for texture classification of digital images. *Física de la Tierra*. **17**, 47–59 (2005).
- Cope, J. S., Remagnino, P., Barman, S. & Wilkin, P. Plant texture classification using gabor co-occurrences. *International Symposium on Visual Computing*. 669–677 (2010).
- Lever, J., Krzywinski, M. & Altman, N. Points of significance: Principal component analysis. *Nature Methods*. **14**, 641–642 (2017).
- Zhang, Q. et al. A shell dataset, for shell features extraction and recognition. *figshare*, <https://doi.org/10.6084/m9.figshare.c.4428335> (2019).
- Sasaki, Y. The truth of the F-measure. *Teach Tutor mater*. **1**, 1–5 (2007).

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 61602540) and University of Macau (MYRG2018-00053-FST).

Author contributions

Q.Z.: Experiment Design, Data Organizing, Data Processing, Data Analysis, Drafting of Manuscript. J.Z.: Data Organizing, Data Analysis, Drafting of Manuscript. J.H.: Data Acquisition, Data Organizing. X.C.: Data Processing, Data Analysis. S.Z.: Data Processing, Data Analysis. B.Z.: Experiment Design, Supervision.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to B.Z.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

The Creative Commons Public Domain Dedication waiver <http://creativecommons.org/publicdomain/zero/1.0/> applies to the metadata files associated with this article.

© The Author(s) 2019