

# Tutorial: Machine Learning

The objective of this tutorial is to classify images by using machine learning techniques. Some images, as illustrated in Figure 1 of the Kimia database will be used [1, 2].

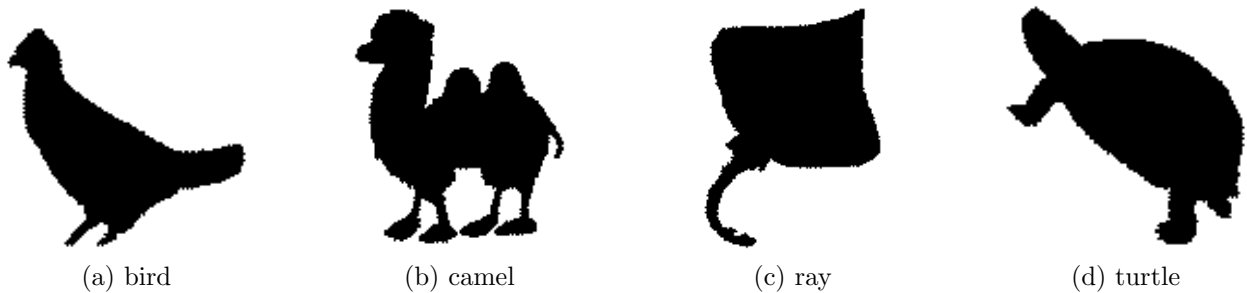


Figure 1: The different processes will be applied on images from the Kimia database. Here are some examples.

## 1 Feature extraction

The image database used in this tutorial is composed of 18 classes. Each class contains 12 images. All these 216 images come from the Kimia database. In order to classify these images, we first have to extract some features of each binary image.



1. For each image in the database, extract a number of different features, denoted *nbFeatures*.
2. Organize these features in an array of *nbFeatures* lines and 216 columns. In this way, each column *i* represents the different features of the image *i*.



You can use the Matlab function `regionprops` to extract some features such as area, eccentricity, perimeter, solidity...



You can use the Python function `skimage.measure.regionprops` to extract the same geometrical features.

## 2 Image classification

In order to classify the images, we are going to use neural networks. Pattern recognition networks are feedforward networks that can be trained to classify inputs according to target classes. The inputs are the features of each image. The target data are the labels, indicating the class of each image.



1. Build the target data, representing the class of each image. The target data for pattern recognition networks should consist of vectors of all zero values except for a 1 in element  $i$ , where  $i$  is the class they are to represent. In our example, the target data will be an array of 18 lines and 216 columns.
2. The database will be divided into a training set (75%) and a test set (25%).
3. Run the training task and classify the test images.
4. Show the classification confusion matrix as well as the overall performance.
5. Try to run the same process. What can you conclude?
6. Try to change the parameters of the network to improve the classification performance.



Look at the Matlab function `patternnet` to make the classification. Look at the field `divideParam` of your pattern network for performing the division.



Look at the Python module `sklearn.neural_network.MLPClassifier` to make the classification. Notice that your data may be normalized before performing the classification.

## References

- [1] <http://vision.lms.brown.edu/content/available-software-and-databases>. 1
- [2] Daniel Sharvit, Jacky Chan, Huseyin Tek, and Benjamin B Kimia. Symmetry-based indexing of image databases. In *Content-Based Access of Image and Video Libraries, 1998. Proceedings. IEEE Workshop on*, pages 56–62. IEEE, 1998. 1