

Title	SURF-based species identification system
Authors	Beatriz Otero ¹ , Eva Rodriguez ¹ and Jacint Ventura ²
Affiliations and contacts	<p>¹ Departament d'Arquitectura de Computadors, Universitat Politècnica de Catalunya, Campus Nord, Mòdul C6, Despatx 204, Jordi Girona 1-3, E-08034 Barcelona, Spain {botero, evar}@ac.upc.edu Tel./Fax: (+34) 934054046, (+34) 934017190 (+34) 934017055</p> <p>² Departament de Biologia animal, Biologia Vegetal i Ecologia, Facultat de Biociències, Universitat Autònoma de Barcelona, Cerdanyola del Valles, Spain</p>

SURF-based species identification system

Beatriz Otero^{1*}, Eva Rodriguez¹, Jacint Ventura²

¹ Departament d'Arquitectura de Computadors, Universitat Politècnica de Catalunya,
Campus Nord, Barcelona, Spain

² Departament de Biologia animal, Biologia Vegetal i Ecologia, Facultat de Biociències, Universitat
Autònoma de Barcelona, Cerdanyola del Valles, Spain
{botero, evar}@ac.upc.edu, jacint.ventura.queija@uab.cat

Corresponding author: Mòdul C6, Despatx 204, Jordi Girona 1-3, E-08034.

Tel./Fax: (+34) 934054046 / (+34) 934017055

Abstract The development of tools for the automated identification of species will reduce the burden of routine identifications conducted by many biologists. The design of these tools is difficult because it depends on the proper extraction of those most relevant characteristics of the image, namely, those unequivocally identify its species. The appropriate software for such extraction does not exist in all cases. This work proposes an architecture for the automated identification of the skulls of different mammalian species belonging to the order Eulipotyphla, which includes shrews, moles and hedgehogs, among others. Our system determines nine species of this mammalian group using existing object recognition techniques, identifying them based on a set of images of the skulls of these species in a digital image database. To validate the proposed architecture, mobile and web applications have been developed. These applications use the image recognition technology provided by the OpenCV library for the detection of the keypoints and matching of the images. The application extracts the descriptor of the input image using the Speed Up Robust Features (SURF) method and compares this descriptor against the image database for matching using a Euclidean distance based on the nearest-neighbor approach. The initial tests have achieved a reliability of 98%.

Keywords Species identification, image recognition system, OpenCV, SURF.

1 Introduction

Systematics can be defined as the science of biological classification and of the evolutionary relationships among organisms. Associated with systematics is taxonomy, a discipline that includes the description, naming and identification of organisms, and their organization in a classification system. In taxonomy, a group of organisms evolutionarily related within the classification system constitutes a taxon, with the species being the key taxon in biology. In fact, species identification is fundamental in any biological study and constitutes the basis of any research focused on animal biodiversity.

In mammals, in general, the skull (cranium and mandible) and teeth are essential structures for classifying and identifying both living and extinct species. In many mammalian groups, species identification based on such characteristics requires a high level of expertise and, frequently, the use of optical instruments. Thus, in some cases, species identification can be a complex and tedious task. A good example of this situation is found in Eulipotyphla, the mammalian order that includes solenodons (Solenodontidae), hedgehogs, gymnures (Erinaceomorpha), and moles and shrews (Soricomorpha). In the present study, we have selected some representative species of Iberian Eulipotyphla (one hedgehog, one mole and seven shrew species) to take a first step toward generating a digital method for mammalian species identification based on skull morphology. Our main aim is that this method will be helpful for studies or activities concerning mammalian diversity in which quick species identification is needed.

The system for automatic species identification presented here takes advantage of existing object recognition methods applied in the field of biology, in particular for plant species [15] and chimpanzee individuals [8]. Plants are identified from their leaves using the inner distance and from their bark using multi-block local binary patterns, while human face recognition algorithms are adapted for chimpanzee identification, combining local descriptors and holistic global features and making use of a decision fusion scheme. In this case, the authors use the Speed Up Robust Features (SURF) descriptor for local feature extraction of eye markings. Specifically, the architecture we have developed constitutes an innovative approach to species identification that makes use of object recognition based on the SURF method. Our procedure has been implemented and initially tested for those Eulipotyphla species considered, showing 98% reliability.

The main characteristics used to differentiate the Eulipotyphla species herein considered are skull size, presence or absence of zygomatic arch, tooth color and number of each type of tooth. We have used skulls of adult individuals to avoid the effect of morphological variation related to growth. However, it is worth mentioning that, in moles and shrews, the adult skull size and shape are attained early in the postnatal period, specifically, when the animal leaves the nest (usually around the three weeks after birth; see, e.g., [4]). Likewise, the other characteristics mentioned above do not vary between geographic populations.

In our system, the input is an image of a cranium captured using a digital camera or a mobile device. The image is sent to a server that determines the species, matching it against the species stored in the database. Finally, the matching species is sent to the client and shows the result, with some descriptive information regarding the species. Two prototypes have been implemented to test the solution, a mobile and a web application. We can find similar initiatives in other areas of a different nature. For example, Bay in [2] and Ruf in [14] propose an interactive guide that recognizes the objects in a museum (paintings, sculptures, ceramics, etc.). They also implement a mobile application to validate the proposed solution. To our knowledge, there is no any similar system for vertebrate species identification based on skull images.

The rest of the article is organized as follows. Section 2 presents related works in the image recognition field. Section 3 proposes the architecture for the species identification system.

Section 4 presents the experimental results. Finally, Section 5 presents the conclusions and new directions for future work.

2 Related works

In the last decades, two different approaches have evolved in the object recognition field. One of the first object recognition approaches was based on the use of feature vectors to characterize objects, with the selected ones usually being global features [12]. Many global features have been proposed in this field, among the most popular of which are color histograms [16]. Later, local features [17] were increasingly used as robust image representations to recognize objects or scenes. The literature on local feature detection is vast because many different important research directions have been followed. A large number of researchers have studied the curvature of contours to find corners; others have analyzed image intensities or used color-based or model-based methods; and other research lines have tried to reproduce processes in the human brain. Nevertheless, more recent works focus on techniques for the detection of invariant features. In 1999, Lowe [9, 10] proposed one of the first object recognition systems that used local features invariant to image scaling, translation, rotation and, partially, illumination, generated by the Scale Invariant Feature Transform (SIFT) method. Some variants of SIFT were subsequently developed, such as PCA-SIFT [5], which uses Principal Component Analysis (PCA), a standard technique for dimensionality reduction, instead of a histogram to normalize the gradient patch, resulting in a feature vector significantly smaller than the SIFT one; and GLOH [11], Gradient Location and Orientation Histogram, proposed by Mikolajczyk and Schmid, which divided the patch into a radial and angular grid instead of a 4x4 regular grid, obtaining higher robustness and distinctiveness. In 2006, Bay et al. [1, 3] proposed a novel scale and rotation invariant feature detector, called SURF, which surpasses SIFT in speed and recognition accuracy, as demonstrated in [1]. This was achieved because SURF uses a Hessian-based matrix for the detector and a distribution-based descriptor. The upright version of SURF, U-SURF [3], has also been developed for object recognition applications invariant to image rotation, providing better results in terms of performance because it does not perform orientation-related operations.

SURF and SIFT, as well as their variations, have been successfully used in image retrieval systems [7], as well as, in fast and robust object recognition systems, for example, in the recognition of objects in a museum [2, 14]. In such systems, users take pictures of different objects of interest in a museum, and the system provides a detailed description for each of the objects recognized by the system. In the application scenario described in this paper, the user takes a picture of a cranium of the species considered, which will be compared with all the images in the system database. The species is determined whether it matches any of the species in the database.

3 System description

In this section, we propose an architecture for the automatic identification of mammalian species based on skull morphology. The system determines the species using existing object recognition techniques, identifying it based on a set of pictures representative of the different species in a

digital image database. It is worth noting that the architecture has been designed with the aim that it be usable for a wider range of mammalian groups. As a first step of this project, we have tested this system using a sample formed exclusively from the Eulipotyphla species considered.

The proposed architecture consists of two main modules: the client and the server. The client module (see Figure 1) allows the user to interact with the system. It acquires and sends sample data (a digital image) and receives the result (the species name). The client also has been designed to support additional computation, such as feature extraction from images. This functionality is provided to the user when she has a mobile device that offers good hardware performance or for low data transmission rates because in this latter case it is much more efficient to send the metadata of the image (feature vector) rather than the image itself.

In the first prototype implemented, the client only acts as a peripheral that receives and displays information to the user. It does not perform image analysis operations. The server performs all the operations required (feature extraction and image matching) to identify the species to which the image belongs.

The subsequent parts of this section provide detailed descriptions for the different modules of the architecture.

3.1 Client module

The *Client module* consists of two submodules, the *GUI client* and the *Image analysis*. The main functionalities offered by this module are the user interaction with the system and the image analysis. The former captures an image, sends it to the server module and receives the result; that is, the biological species to which the input image belongs. The latter optionally pre-processes the image, if necessary, and computes the feature vector.

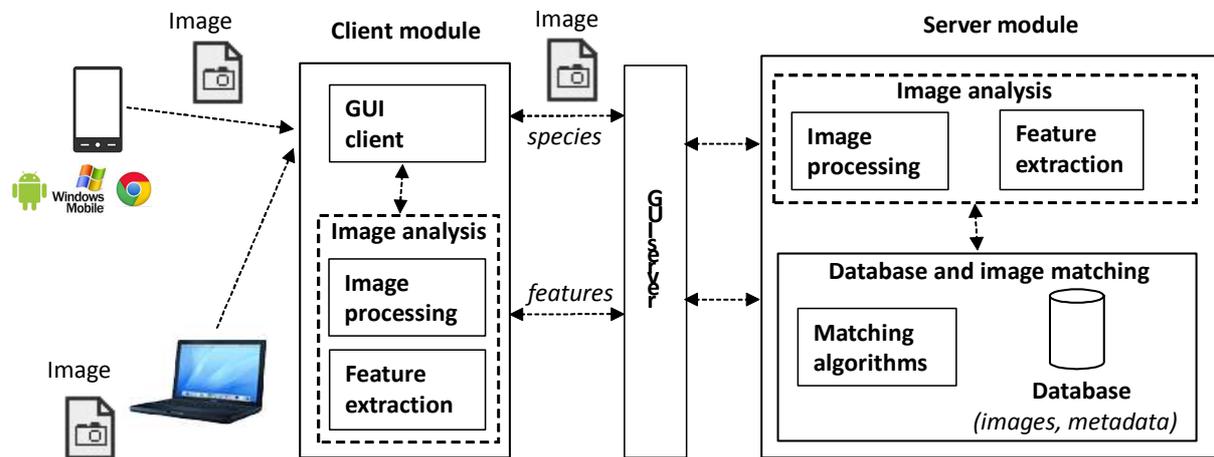


Fig. 1 Overall architecture to identify the biological species

3.1.1 GUI client

The *GUI client* submodule is the core of the mobile application and consists of a Graphic User Interface (GUI). When the user launches the application, this submodule creates the directory hierarchy, if it has not been created before, and obtains a temporary file to store the input image. Then it calls an application able to take photographs (if there are more than one, the system allows the user to choose between them). This application takes control of the mobile screen and, when the user takes a photo, the camera application returns the result to the *GUI client* submodule. If the user has not cancelled the action (pressing the “BACK” button, for example), then he will have a photograph stored in the path previously established. At this point, the submodule performs two tasks: displaying the photograph taken and sending it to the server module to obtain the result (the matching species if there is one).

Figure 2 shows two screenshots of the mobile application. The first screenshot is the presentation view, which contains two buttons, one to access the camera for taking the image and one to display the saved images. The second screenshot, when the captured image (input image) belongs to the Pygmy shrew, *Sorex minutus*, is the result view that shows the information (i.e., image, name, geographic range, habitat, etc.) of the biological species obtained. The “BACK” button located at the bottom returns to the initial view of the application to start the process again.

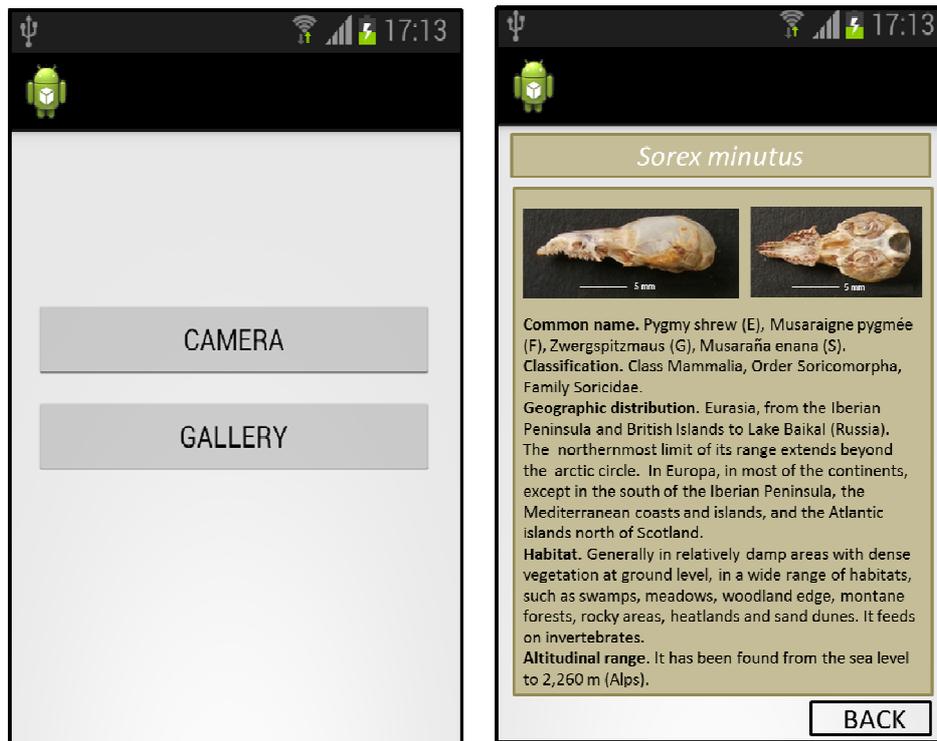


Fig. 2 Screenshot of the mobile application

3.1.2 Image analysis

This submodule can be present in the client or server modules. For the first prototype, it has been implemented in the server because programming feature extraction on the mobile platform is cumbersome and the runtime will increase. This submodule is described in detail in section 3.2.1.

3.2 Server module

The main functions of this module are receiving the digital image, extracting its features, matching them with the ones in the image database, determining the biological species name and sending it to the client module. This module is composed of the *image analysis* and *database and image matching* submodules.

3.2.1 Image analysis

This submodule performs the off-line extraction of the low-level metadata from the new species images in the database or the on-the-fly extraction of the same metadata from a digital image submitted by a user mobile application. This vector's features are stored in the database along with the new image. Initially, the system carries out the analysis of all the images associated with the reference species and includes such data (feature vector and image) in the database. Additionally, for each new digital image provided by the user, the server module performs the extraction of its features and matches them against all the reference images in the database.

This module consists of the *image processing* and the *feature extraction* submodules. We have implemented both submodules using the OpenCV software. The first submodule applies functions to convert the input image to its equivalent grayscale image. This conversion is needed to subsequently obtain the image features and to apply the image recognition methods. The second submodule detects the image features using specific image recognition methods. These methods detect the interest points in the image (key points) that need to be matched against the set of images in the database. Figure 3 shows an example of the key points obtained for one shrew's skull image using a specific detection method. In this figure, the interest points are identified by the white circles.

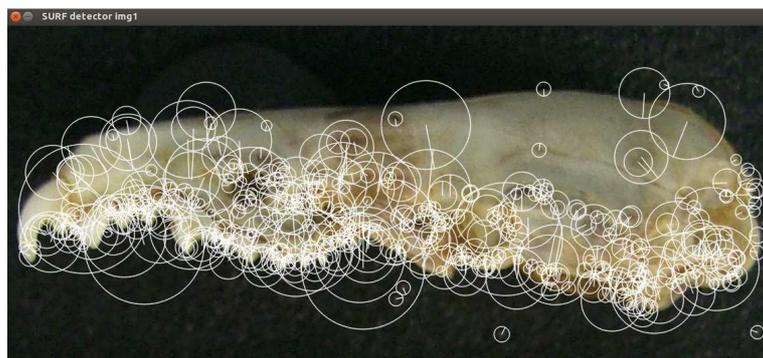


Fig. 3 Interest points detected for the *Crocidura russula* species' skull image using `surf.detect()` function

3.2.2 Database and image matching

This second submodule is formed by the database and the *matching* submodule. The main functionality of the database is to store (for the all images stored in the database) the image itself, its descriptor and the species name. The database is implemented and accessed using the SQLite administration system. SQLite allows efficient access to data; also, it is based on .db files, which allows easy portability.

The *matching* submodule calculates the key points and the feature descriptor vectors of the image using matching algorithms. Figure 4 shows an example of the matching points for two pictures using the Fast Library Approximate Nearest Neighbor (FLANN) matching algorithm. In this figure the lines identify the resultant key points, in both images, for the best matches. The total number of key points with this characteristic determines the size of the matching vector. This submodule is implemented using the Java programming language and the OpenCV library.



Fig. 4 Matching of interest points for the two images of the skulls of shrews of the *Crocidura russula* species, using the FLANN matching algorithm

4 Experimental results

In this work, we have implemented two prototypes to analyze the performance of the system. The first prototype uses a mobile client, while the second employs a client Web application. The next part of this section provides details of the application scenarios, the software and hardware used and the processing time obtained for each case.

4.1 Material

The training sample data used in this study are skulls of nine eulipotyphlan species belonging to the Collection of the Animal Biodiversity Research Group of the Universitat Autònoma de Barcelona. The species selected are the following: Erinaceomorpha, Western hedgehog, *Erinaceus europaeus*; Soricomorpha, common mole, *Talpa europea*; greater white-toothed shrew, *Crocidura russula*; pygmy white-toothed shrew, *Suncus etruscus*; common shrew, *Sorex araneus*; Millet's shrew, *Sorex coronatus*; pygmy shrew, *Sorex minutus*; water shrew, *Neomys fodiens*; and Miller's water shrew, *Neomys anomalus*. We have chosen these species because

they are closely related from an evolutionary point of view and all of them show clearly identifiable traits in skull and dentition. For each species, we have used two individuals.

The mammalian skull is a complex structure that houses and protects the brain and the receptors of major senses, and forms structures related to feeding. Two main parts compose the mammalian skull: an upper cranium, with braincase and rostral regions; and the mandible or lower jaw. The skull is completed with the hyoid apparatus, a component of the visceral skeleton that partially supports the tongue.

The braincase is a “box” formed by different bones that together protect the brain. Attached or associated with the braincase are the auditory bullae (not present in all mammals), which house the middle and inner ears, and other structures, such as the occipital condyles, which articulate the cranium with the first cervical vertebra (atlas), and numerous processes and ridges that serve as points of attachment for muscles. Among the numerous bones that compose the rostral region, there are the paired premaxillae and maxillae, the tooth-bearing bones of the upper jaw. Many mammals show zygomatic arches, conspicuous bony arches formed by the contribution of three bones that constitute the ventral and lateral borders of the orbits and the temporal (lateral) region. Compared to the cranium, the mandible is a simple structure, as it is composed of only two elements, the left and right dentary bones.

Teeth are very important structures in the identification of living mammalian species. Likewise, many fossil lineages are described only on the basis of their teeth. Most mammals have dentition with different type of teeth (heterodont dentition). In a typical mammal pattern, the following sort of teeth can be recognized, in an anteroposterior direction: incisors (I), canines (C), premolars (P) and molars (M). In the cranium, the incisors, the anterior-most teeth, are rooted in the premaxillae, the anterior-most bones, and the other teeth are rooted in the maxillae bones. In the lower jaw, teeth are rooted in the dentary bones. Incisors are structurally simple but sometimes are highly modified. Canines are single-cusped teeth. Premolars may have one cusp or may be similar to molars, which have more than one cusp. In shrews the posterior incisors, the canines, and the anterior premolars may all be single-cusped teeth that are difficult to distinguish from one another. These are collectively termed the “unicuspids”, and a particular tooth may be referred to as, for example, the second upper unicuspid.

Considering all these features, Table 1 shows the steps that must be followed to identify these species by considering skull and dentition traits. Note that this procedure requires the use of a stereomicroscope to identify and count the number of unicuspid.

The test sample consists of a series of photos for these species with a lateral view (in total 155 images). These images are stored in the database with additional information (their descriptors and specific species' information, e. g., species' name, tooth color). Figure 5 shows a subset of images, for 9 different species, in the database. All the reference images were captured by a LUMIX DMC-LX3 camera, while the images taken by the user were captured with mobile device cameras (Samsung Galaxy Mega and iPhone 5.0).

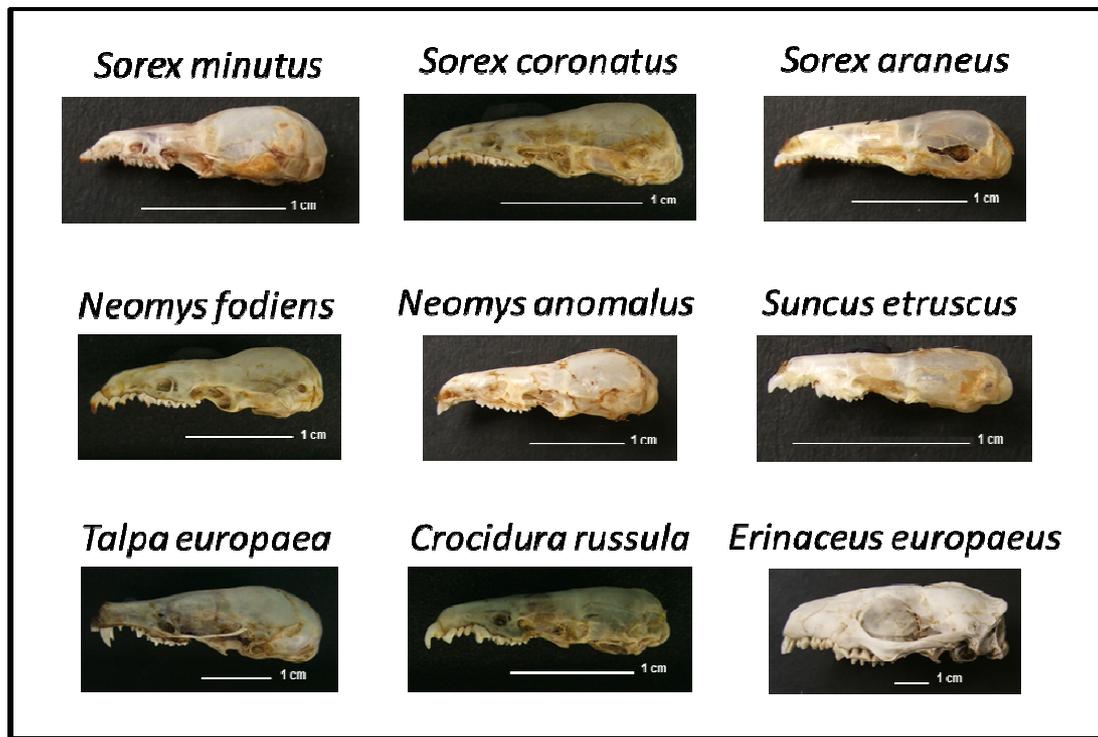


Fig. 5 Subset of lateral view images (for 9 different species) in the database

Table 1 Identification key for the species considered in this study based on skull and dentition characteristics

Step	Features	Species name
1	With zygomatic arches. First upper incisive with 1 cusp.	Go to step 2.
1	Without zygomatic arches. First upper incisive with 2 cusps.	Go to step 3.
2	Nasal sutures visible. Robust zygomatic arches. 20 teeth.	<i>Erinaceus europaeus</i> .
2	Nasals fused to the skull. Thin and fragile zygomatic arches. 22 teeth.	<i>Talpa europaea</i> .
3	Teeth with red cusps.	Go to step 4.
3	Teeth not pigmented.	Go to step 6.
4	4 unicuspid. 18 teeth.	<i>Neomys fodiens</i> or <i>N. anomalus</i> .
4	5 unicuspid. 20 teeth.	Go to step 5.
5	3rd unicuspid larger than the 2nd one.	<i>Sorex minutus</i> .
5	2nd unicuspid larger than the 3rd one.	<i>Sorex araneus</i> or <i>Sorex coronatus</i> .
6	3 unicuspid. 16 teeth.	<i>Crocidura russula</i> .
6	4 unicuspid. 18 teeth.	<i>Suncus etruscus</i> .

4.2 Software and hardware

In this work, we use the JAVA programming language to implement the client mobile application and PHP to develop the Web application. The Web application server used is Glassfish, which has a powerful, complete and robust servlet container. The web services, which provide the image processing and matching functionalities, make use of the OpenCV library for the Java programming language.

Finally, to evaluate the response time of the experiments, we tested the system using a virtual private server equipped with a 2.80 GHz Intel(R) Xeon (TM) CPU, 384 MB RAM and Debian Linux 14.01.

4.3 Process application description

Initially, the user takes a photo using a mobile device. This image is then sent as input to the *Image Processing Web service* to be subsequently analyzed (species identification). Next, the *Feature Extraction Web service* is called, which extracts the image features; the *Image Matching Web service* then matches the user photo against the images in the database. The image matching process uses different methods combined with different detection algorithms and descriptor matchers. The *Matching Web service* then selects the image in the database with the most matching points, and returns to the client, as a response, the species' name, which the client presents to the user, together with relevant information for the species. If it is not possible to find an image in the database similar to the image captured by the user, then the server sends a message to the client so indicating.

4.4 Processing time

In previous sections we mentioned that SURF and SIFT, as well as their variations, have been successfully used in fast and robust object recognition systems. For this reason, in this subsection we study the behavior of both methods to select one of them. Figures 6 and 7 show, respectively, the CPU time required to obtain the descriptor vector for 15 skull images and the CPU time for the matching. For all the cases, the SURF method outperforms the SIFT method because SURF uses less CPU time than the SIFT method to obtain the image descriptor vector and to do the matching with the database image set. That is because the SURF descriptor vectors are of lower dimensions than the SIFT ones. Moreover, the SURF method reported 143 matches more than the SIFT method. Then, we have chosen to use the SURF method when performing the experiments.

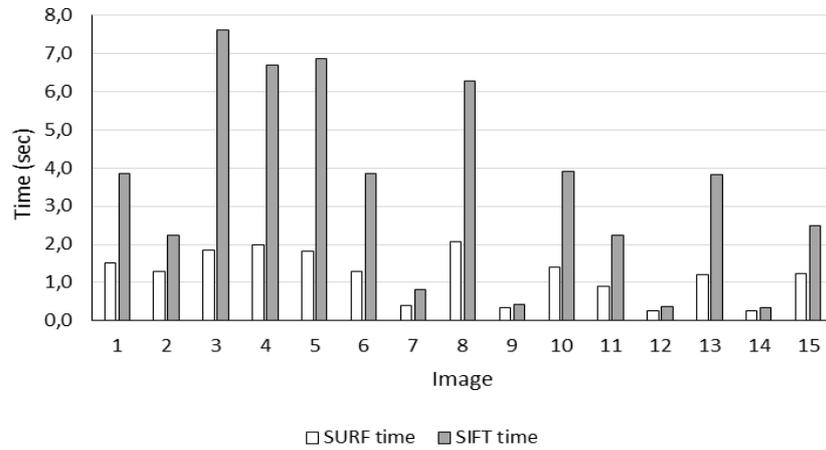


Fig. 6 Descriptor vector processing time in seconds using SURF and SIFT methods

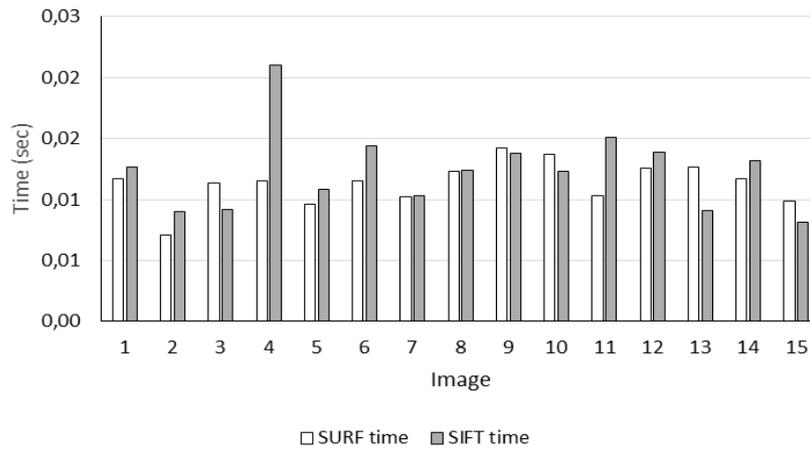


Fig. 7 Matching time in seconds for the SURF and SIFT methods

Once the method is chosen, it is necessary to determine the variant to be used (SURF-64 or SURF-128) as well as the detection algorithm and the descriptor matchers. Table 2 lists the experimental results for the average times obtained for the matching process using different detection algorithms and descriptor matchers. In this table, the highest CPU time was registered for the *Hessian* detector and the *BruteForce* descriptor. These times are close to those obtained for the *FlannBased* descriptor. However, although other methods in the table require less CPU time, in this paper, we use the third combination (*Hessian-Laplace* detector and *FlannBased* descriptor) because this combination is the one that has the best matching results.

Table 2 Computational time for matching the mobile image with 155 images in the database using different detection algorithms and descriptor matchers

Method	Detection algorithm (depending on the threshold)	Descriptor matchers	CPU time to read descriptors and match images (secs)	
SURF-64	<i>Hessian</i>	<i>FlannBased</i>	2.77	
	<i>Fast-Hessian</i>		2.29	
	<i>Hessian-Laplace</i>		1.46	
	<i>Hessian</i>	<i>BruteForce</i>	3.06	
	<i>Fast-Hessian</i>		2.16	
	<i>Hessian-Laplace</i>		0.9	
	SURF-128	<i>Hessian</i>	<i>BruteForce-L1</i>	2.94
		<i>Fast-Hessian</i>		2.04
		<i>Hessian-Laplace</i>		0.86
SURF-128		<i>Hessian</i>	<i>FlannBased</i>	2.88
		<i>Fast-Hessian</i>		2.29
		<i>Hessian-Laplace</i>		1.44
		<i>Hessian</i>	<i>BruteForce</i>	3.07
		<i>Fast-Hessian</i>		2.17
		<i>Hessian-Laplace</i>		0.9
	SURF-128	<i>Hessian</i>	<i>BruteForce-L1</i>	2.91
		<i>Fast-Hessian</i>		2.04
		<i>Hessian-Laplace</i>		0.86

The total time for the manual species' identification depends on the species to be recognized and the expertise of the researcher who performs the task. Thus, considering these factors, the identification process takes approximately 30 seconds, if it is performed by an expert or from 5 to 10 minutes if it is performed by an inexperienced person.

For the experiments conducted, the total average response time for the mobile application is 10.05 seconds, while for the Web application, it is 6.1 sec. In both cases, most of this time is used to upload the input image. For example, the image upload time in the mobile application represents the 45.7% of the total response time, while the remaining time is devoted to image processing, feature extraction and matching operations (which represent the 22.9%, 16.9% and 14.5% of the total time respectively). Note that the time may vary depending on the connection network and the network data transfer speed. However, despite this, the results show that the mobile application is competitive with the manual identification process, even when the person making the recognition of the species is an expert.

An interesting objective for the future is to reduce the response time by performing some computations, such as feature extraction, in the mobile device, thereby avoiding the image upload to the server, or even performing all the computations in the mobile device, taking advantage of the ORB algorithm for image processing in smart devices, which is two orders of

magnitude faster than SIFT or SURF, as demonstrated in [13], and which has been successfully used in different image detection applications for mobile devices [6, 13].

For all tests, the species has been identified with a reliability of 98%. The species identification process fails in two cases: i) when two species are very similar in skull morphology (e.g., *Sorex araneus* with *Sorex coronatus*) and ii) when the image background is not black or the light intensity is not adequate.

As mentioned in the Introduction section, the prototype does not take into account the effect of the intraspecific size variation (maxilla length) on the species identification process. Nevertheless, it is worth noting that this source of variation in the other characteristics considered here is not significant. Regardless, to increase the robustness of the species identification method, further objectives are to include more species of Eulipotyphla in the database and increase the number of photos of all species.

4.5 Mobile resource consumption

Here, we briefly quantify resource consumptions of our mobile application in terms of power demand, CPU utilization and memory usage. To do so, we use the PowerTutor Android application to measure the amount of energy spent by our application, while the Dalvik Debug Monitor Server tool allows quantifying CPU and memory utilization. Once the identification process starts, it spends only 0.75% of the full battery energy, takes 15% of the CPU capacity, and requires 250Kb of the mobile RAM. These results correspond to average values over the observed consumption data when processing our whole set of 300 images using three basic mobile devices (100 images per device). As one can see, our identification application is actually a non-demanding mobile application.

5 Conclusions and future work

This paper proposes a generic architecture for the automated identification of different mammalian families. It determines the biological species of mammals using existing object recognition techniques. A first simple prototype has been implemented for a set of species belonging to the mammalian order Eulipotyphla; a species identification reliability of 98% has been achieved. This architecture makes use of object recognition techniques; more specifically, it is based on the SURF method. This first prototype implements the feature extraction part in the server. The response time for the species identification mobile application is acceptable, being between 10 and 12 seconds. These results reduce the time dedicated by a non-expert biologist in the species identification process by up to 66.6%. Moreover, the probability of successful identification is 98%.

An interesting approach to improving the mobile application response time is to implement the image analysis module in the client; or even implement all the server modules in the client, making use of the ORB algorithm (instead of SURF), which has been successfully used in different image detection applications. The first option would reduce the amount of information that needs to be sent to the server, minimizing the upload time of the image, because in this case

only the feature vector is sent. In the second option, there would be no time spent transferring the image to the server because the entire process is performed on the mobile device. In this last case, the successful implementation will depend of the capabilities (processing speed and memory) of the mobile device and the response time of the ORB algorithm.

Another future research line is to include in the database additional photos of other individuals of the same species to study the influence of the individual variation factor on biological species identification. Additionally, the design of this prototype will be extended for species identification in other mammalian groups.

Acknowledgments

The authors would like to thank A. Quiroga Bertolin for helpful support in the first prototype implementation.

References

1. Bay H, Ess A, Tuytelaars T, Van Gool L (2008) Speeded-Up Robust Features (SURF). *Journal Computer Vision and Image Understanding* archive, 110(3): 346-359. DOI: 10.1007/11744023_32
2. Bay H, Fasel B, Van Gool L (2006) Interactive museum guide: fast and robust recognition of museum objects. *Proceedings of the First International Workshop on Mobile Vision (IMV'06)*, Graz, Austria.
3. Bay H, Tuytelaars T, Van Gool L (2006) SURF: Speeded Up Robust Features. *Proceedings of the ninth European Conference on Computer Vision*, 3951:404-417. DOI: 10.1007/11744023_32
4. Churchfield S (1990) *The natural history of shrews*. Comstock Publishing Associates, Ithaca, New York.
5. Ke Y, Sukthankar R (2004) PCA-SIFT: A more distinctive representation for local image descriptors. *Computer Vision and Pattern Recognition*. DOI: 10.1109/CVPR.2004.1315206
6. Kim Y, Park J, Moon I, Oh C (2014) Performance Analysis of ORB Image Matching Based on Android. *International Journal of Software Engineering and Its Applications*, 8(3):11-20.
7. Lee Y, Kim Y (2015) Efficient image retrieval using advanced SURF and DCD on mobile platform. *Multimedia Tools and Applications* 74:2289–2299. DOI 10.1007/s11042-014-2129-5
8. Loos A, Ernst A (2013) An automated chimpanzee identification system using face detection and recognition. *EURASIP Journal on Image and Video Processing*, 2013:49, 1-17. DOI: 10.1186/1687-5281-2013-49
9. Lowe D G (1999) Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2:1150-1157. DOI: 10.1109/ICCV.1999.790410
10. Lowe D (2004) Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*, 60(2):91–110. DOI: 10.1023/B:VISI.0000029664.99615.94
11. Mikolajczyk K and Schmid C (2005) A performance evaluation of local descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630. DOI: 10.1109/TPAMI.2005.188
12. Murphy K, Torralba A, Eaton D, Freeman W (2006) Object Detection and Localization Using Local and Global Features. *Toward Category-Level Object Recognition*, LNCS 4170:382-400. DOI: 10.1007/11957959_20

13. Rublee E, Rabaud V, Konolige K, Bradski G (2011). ORB: an efficient alternative to SIFT or SURF. IEEE International Conference on Computer Vision (ICCV), 2564-2571. DOI: 10.1109/ICCV.2011.6126544
14. Ruf B, Kokiopoulou E, Detyniecki M (2008) Mobile museum guide based on Fast SIFT recognition. In: Adaptive Multimedia Retrieval. Identifying, Summarizing, and Recommending Image and Music. Springer Berlin Heidelberg, Volume 5811 of the series Lecture Notes in Computer Science, pp. 170-183. DOI: 10.1007/978-3-642-14758-6_14
15. Sixta T. Image and Video-based Recognition of Natural Objects. Diploma Thesis. Czech Technical University in Prague. Faculty of Electrical Engineering, pages 50, Prague, 2011.
16. Swain M, Ballard D (1991) Color indexing. International Journal in Computer Vision, 7(1): 11-32. DOI: 10.1007/BF00130487
17. Tuytelaars T, Mikolajczyk K (2008) Local invariant features detectors: A Survey. Journal Foundations and Trends in Computer, Graphics and Vision archive, 3(3): 177-280. DOI: 10.1561/06000000017a