Coral Reef Fish Detection and Recognition in Underwater Videos by Supervised Machine Learning: Comparison Between Deep Learning and HOG+SVM Methods

Sébastien Villon[1(✉)], Marc Chaumont[1,2], Gérard Subsol[2], Sébastien Villéger[3], Thomas Claverie[3], and David Mouillot[3]

[1] LIRMM, University of Montpellier/CNRS, Montpellier, France
villon@lirmm.fr
[2] University of Nîmes, Nîmes, France
[3] MARBEC, IRD/Ifremer/University of Montpellier/CNRS, Montpellier, France

Abstract. In this paper, we present two supervised machine learning methods to automatically detect and recognize coral reef fishes in underwater HD videos. The first method relies on a traditional two-step approach: extraction of HOG features and use of a SVM classifier. The second method is based on Deep Learning. We compare the results of the two methods on real data and discuss their strengths and weaknesses.

1 Introduction

Quantifying human impact on fish biodiversity in order to propose solutions to preserve submarine ecosystems is an important line of research for marine ecology. This quantification requires in situ sampling of the fish community. Measurements based on extraction-fishing give only limited data, and could lead to misinterpretation [1]. Moreover, the use of fishing, even for survey purposes, impacts the studied biodiversity.

Another standard method consists in two divers who note visual observations of fishes under water. This kind of survey is expensive in both time and money, and results are greatly impacted by divers' experience and fish behavior. Moreover, data acquisition remains limited by the human physical capacities [1].

A more recent method consists in acquiring underwater images or videos [3], with either a moving or a fixed camera. An expert will then be asked to detect, count and recognize fishes on a screen offline. At the moment, this task is performed entirely manually, and the amount of data is often too large to be completely analyzed on screen. Moreover, the latest technical improvements of HD camera allow recording fish communities for a long time at a very low cost. Significant examples of a huge amount of underwater HD images/videos, that have been collected for assessing fish biodiversity, are the 115 terabytes of the European project Fish4Knowledge [3], or the XL Catlin Seaview Survey Initiative[1].

[1] http://catlinseaviewsurvey.com/.

J. Blanc-Talon et al. (Eds.): ACIVS 2016, LNCS 10016, pp. 160–171, 2016.
DOI: 10.1007/978-3-319-48680-2_15

Fig. 1. Left, a 640×480 highly compressed frame extracted from the SeaClef database. Right, a 1280×720 HD frame from the MARBEC laboratory database.

The research community in image processing has been asked to propose algorithms in order to assist, and recently even automatize the detection/identification of fishes in images or videos. Recently, a challenge called Coral Reef Species Recognition has been proposed in the evaluation campaign SeaClef[2], which is based indeed on Fish4Knowledge data. Unfortunately, in this task, the video quality remains quite limited (640×480 pixels) whereas current acquisitions are in High Definition or even in 4K (see Fig. 1). This offers much more details for image processing but increases the processing time.

Among the issues and difficulties of detecting and recognizing fish in underwater videos, there are color variations due to the depth, lighting variations from one frame to another, the sediments and dirt which degrades the videos quality, or the seaweed which makes the background changing and moving [4]. The classification itself encounters other issues such as the variation of shape or color within the same species and moreover the variation of size and orientation due to the fish position. We chose not to avoid these issues, and to take into account all these problems as we work on videos acquired in natural conditions instead of controlled acquisitions [5]. We chose for this study to focus on the processing of each frame and not on the video.

Many methods to detect and recognize fishes in underwater videos were proposed these last years [3]. In general, the first step consists in selecting features based on the shape, color, context, specifics landmarks or texture [6]. Some algorithms use specific feature vectors computed at some landmarks. Other use more complex features such as SIFT [8–10] or *shape context* (SC) [5]. But in [11], the authors conclude that the Histogram Of Oriented Gradients feature leads to better results than both SIFT and SC.

In the 2015 SeaClef contest [23], the best results have shown that Deep Learning can achieve a better classification for fish detection than SVM or other classical methods. This may be due to the fact that in Deep Learning, features are automatically built by the classifier itself, in an optimal way. The winner of the SeaClef contest used several Deep classifiers and fused the results to obtain the definitive scores. Unfortunately, we will not be able to compare our approach

[2] http://www.imageclef.org/lifeclef/2016/sea.

with his as the databases are different (we have a higher definition and mobile cameras).

In this paper, we propose also to explore the performances of fish detection and classification by Deep Learning. In particular, we assess the results with respect to a more classical method based on a combination of HOG feature extraction and SVM classification. For this, we will use High Definition videos acquired for an actual marine ecology study. In Sect. 2, we briefly present Deep-learning and SVM+HOG methods. In Sect. 3, we detail the implementation in particular the multi-resolution approach and data preprocessing. In Sect. 4 we present the results and compare both methods. In Sect. 5, we present some future work.

2 Presentation of the Methods

2.1 Histogram of Oriented Gradients + Support Vector Machine

The Histogram of Oriented Gradients [12] characterizes an object in an image based on its contours by using the distribution of the orientations of local gradients. As shown in [13], HOG features may lead to better results even in a complex classification task as ours, where a fish can be hidden in coral reefs or occluded by another fish.

The Support Vector Machine (SVM) [7] is a supervised method to classify feature vectors. SVM method represents each vector in a high dimensional space, mapped so that the samples of the different classes are separated by a clear gap that is as wide as possible. Support vector machines have been used in a lot of applications and have shown good overall results [14–17].

2.2 Deep Learning

Since the 2012 ImageNet competition, and new computational power accessible through latest GPU, Neural Network came back as a strong possibility for classification tasks [18]. Moreover, by integrating convolutional layers, Deep Neural Networks (DNN) are able to both create features vectors and classify them.

Neural network is a mathematical model which tries to mimic human brains [19]. Like SVM, neural networks may classify feature vectors after a training phase. A neural network is composed of interconnected nodes called neurons and each neuron of each layer receives a signal from the neurons of the previous layer. This signal is modified according to an activation function and transferred to the neurons of the next layer.

We can define for the neuron n, the first operation $\alpha^{(n)}$ as:

$$\alpha^{(n)}(\mathbf{x}^{(n)}) = \sum_{i=1}^{c} w_i^{(n)} x_i^{(n)} \tag{1}$$

where $\mathbf{x}$ is the input vector, a given neuron, c the number of connections of this neuron, $w_i^{(n)}$ the weight of rank i of a neuron n, and $x_i^{(n)}$ the input of rank i of a neuron n.

We can then define the output of a neuron n as $\sigma^{(n)}$ with $f^{(n)}$ the activation function:

$$\sigma^{(n)}(\mathbf{x}^{(n)}) = f^{(n)}(\alpha^{(n)}(\mathbf{x}^{(n)})) \tag{2}$$

Each layer of a neural network except the first one which receive the feature vector and the last one are called hidden layers. During the learning process, the network will have its parameters modified in order to optimize its classification rate of learning.

We will use the back-propagation. Given a feature vector representing an object from class $k \in \{1..K\}$ as network input, we compare the expected value (100 % of probability to belong to class k) to the results obtained by the network, and we compute the error of the output layer. Then, the error is back-propagated from the layer j to the neurons of the layer $j-1$. Finally, the weight of each neuron is updated according to a gradient-based descent in order to get a computed value closer to the expected value [20].

To make a network able to build its own feature, we move from a simple network to a Convolutional Neural Network (CNN). One or more convolutional layers are connected between the input layer and the hidden layers. Each convolutional layer transforms the signal sent from the previous layer using convolutional kernels, an activation function breaking the linearity and a pooling phase which reduces the image and strengthens the learning by selecting significant pixels (the highest value from a region for instance). The last convolutional layer eventually concatenates all the information in one feature vector and sends it to another layer or to a classifier.

For the training phase, a CNN is given a database consisting of couples $(I^i, l^i)_{i=1}^{i=N}$ with I^i, the image $i \in \{1, ..., N\}$ and l^i its label. Basically, in our application, the label $l^i \in \{1, ..., L\}$ is the fish species.

3 Practical Implementation

3.1 Data Preprocessing

The choice of learning data is a crucial point. We worked with biology experts of the MARBEC laboratory to label many videos. We cropped some frames of the videos and created a training database composed of 13000 fish thumbnails. The thumbnail size varies from a minimum of 20×40 pixels to a maximum of 150×200 pixels. Each thumbnail contains only one labeled-fish as shown on Fig. 2.

We decided to keep only the species with more than 450 thumbnails. We also widen the database by applying rotations and symmetries in order to capture all the possible position of the fishes. Table 1 lists the retained species.

Due to the highly textured natural background, we also added a class for the background. This class is constituted with *random* thumbnails of the background which were randomly selected in frames and *specific* background thumbnails which were taken around the fish thumbnails.

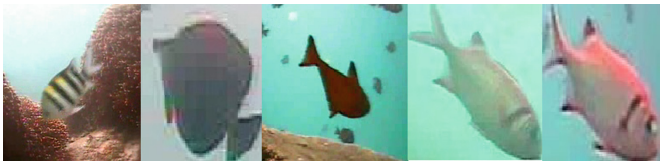

Fig. 2. Some training thumbnails from the MARBEC database

Table 1. Fish species in the learning database

Species	Thumbnails	Rotations and symmetries
Acanthurus lineatus	493	2465
Acanthurus nigrofuscus	1455	3923
Chromis ternatensis	951	4755
Chromis viridis/Chromis atripectoralis	523	2619
Pomacentrus sulfureus	766	3830
Pseudanthias squamipinnis	1180	5900
Zebrasoma scopas	488	2400
Ctenochatus striatus	1400	4000

To be able to do multi-resolution classification, all the background thumbnails were taken with random dimensions, from 40×60 pixels to 400×500 pixels.

Finally, in order to improve the localization accuracy, we decided to create another class called *part of fish*, to ensure that the network does not focus on a distinctive part of a fish as a stripe, a fin, the head, but processes the fish as a whole. We also created a class *fish* which contains some unknown fishes to make the method able to recognize any fish even though it is not in the learning database. However, this class must contain less samples in order to be sure that a fish will most likely be labeled by its specific class rather than the generic class *fish*. Finally, we added 3 classes to our initial thumbnail database as listed in Table 2.

Table 2. Classes added to the species database

Class Label	Samples
Random/specific background	116,820/91,247
Part of Fish	55,848
Fish	970

3.2 Detection/Recognition Pipeline

The HOG+SVM and the Deep Learning methods process the video frames through the same pipeline (see Fig. 3). We chose for this study to process each frame independently without introducing any object tracking algorithm. First, we pass a multi-resolution sliding window through the frame. For each position of the window, the method gives a probability score for each class. We also compute a motion probability score based on the comparison of the current and the previous frame. We then compare the probability scores given by the classifier to some predefined thresholds. If the scores are over the thresholds, we output a bounding box corresponding to the window position. At the end, for each position, we will fuse all the bounding boxes found at different resolutions.

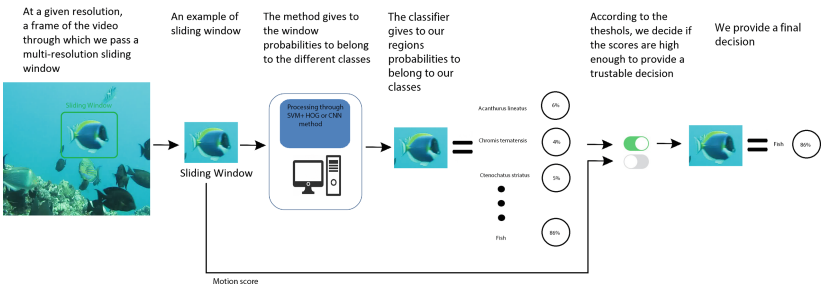

Fig. 3. Detection/Recognition pipeline

Multi-resolution Sliding Window. In order to deal with multi-resolution detection, the size of the sliding window varies from 1/18 of the frame at least, and 1/1 at most. This allows to recognizes fishes with a minimum of 60 pixel length and 40 pixel width, and a maximum equal to the full size of the frame. The sliding window is displaced by a stride equals to a third of the window width.

HOG + SVM. We divide each thumbnail in 10 zones (one zone is the complete thumbnail, and the 9 others are the thumbnail divided in 9 even parts). For each zone, we compute a HOG feature over 8 direction axes, and we concatenate all these HOG features in a unique feature vector. For each fish species, we fed a SVM with all the corresponding thumbnail features as a class, and all the other thumbnails features (other species and background) in order to obtain a specific classifier.

The SVM we used non-linear SVR (Support Vector Machine for regression) implemented using the library libsvm[3] with a Gaussian radial basis function kernel. We obtained a clean separation for the training database (over 85 % of

[3] https://www.csie.ntu.edu.tw/~cjlin/libsvm/.

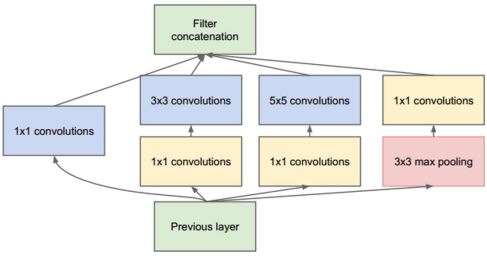

Fig. 4. An inception module, as presented in Szegedy et al. [22]

backgrounds thumbnails have a regression score lesser than 0.5, and 85 % of fishes thumbnails have a regression score greater than 0.5) We built as many classifiers as there are classes, and each classifier discriminates one class against all the others. In the end, if none of the classifiers class the window as a fish, it will be classified as "background".

Deep Learning. The architecture of our network follows the GoogLeNet's with 27 layers, 9 inception layers, and a soft-max classifier. Once we have a list of cropped thumbnails and their labels, we send them to our network. We use inception layers (Fig. 4) based on GoogLeNet architecture [22]. The inceptions here allows us to reduce the dimension of the picture to one pixel, and therefore not to be dependent of the dimensional impact. We adapted some parameters such as the size of the strides and the first convolutions adapted to the size of our thumbnails, which allowed us to achieve better results than a classic architecture (e.g. [18]).

3.3 Post-processing and Bounding Box Fusion

For each sliding window, we define a motion score by computing the average absolute difference with the window at the same position in the previous frame. Based on the hypothesis that most of the fishes are moving, we use this score for the final detection decision.

After processing all the resolutions of a frame, we obtain a list of bounding boxes, and for each bounding box a probability of belonging to a class. Yet the classification remains ambiguous if there is more than one bounding box corresponding to the same potential fish (see Fig. 5).

To suppress the redundant bounding boxes, we first keep the boxes whose probabilities are above a given probability threshold T ($T = 98\,\%$) in the case of Fig. 5. So if the motion score is greater than our motion threshold and the probabilities to belong to a class of fish is greater than 98 %, we keep the box. Then, we fuse the remaining boxes following based on the following properties:

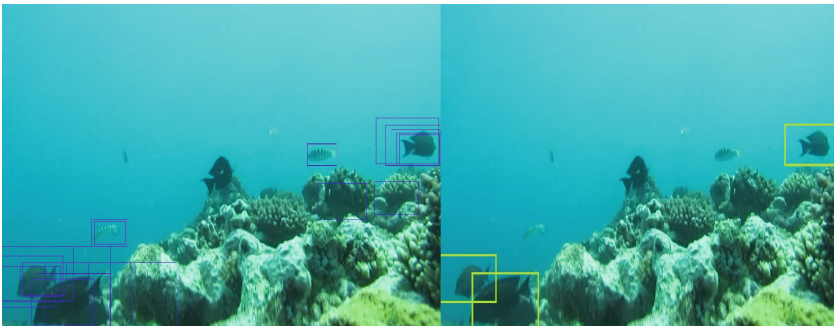

Fig. 5. Results of the detection before (left) and after (right) post-processing. Note that two fishes in the center are not displayed because they do not belong to one of the species which have been learned.

if two bounding boxes are labeled with the same species, and their overlap ratio is greater than 30 %[4], we suppress the bounding box with the lower probability.

4 Results, Comparison and Discussion

We used 4 test videos (which are different from the training videos) to experiment our complete process. The 4 videos were taken on coral reefs, and the diver was holding the camera which then slightly moves. The video acquisition were not deep, and therefore we had a lot of colors on both the fishes and the background but the light is moving with the waves, bringing many distortions. The videos are very different in terms of fish species, background texture, colors, fish density, etc. Biology experts from MARBEC selected 400 frames all over the videos and defined ground-truth bounding boxes of all the visible fishes in the frame.

To determine if a detected bounding box is correct, we compute its overlap ratio with the ground truth bounding box. If this value is over a threshold λ, then the detection is considered as true positive, otherwise it is labeled as false positive. On the opposite, if a ground truth bounding box has no over the threshold overlap with any detected bounding box, it counts as a true negative. We chose to put the value 0.5 to λ.

Results (recall, precision and F-Measure) of the entire detection/recognition process with Deep-Learning method is given in Table 3 with T = 98 %.

We also show on Fig. 6 the relation between recall and precision with respect to the threshold T. The differences come mostly from the texture of the background, but also from the species, as some fishes are easier to detect (bright colors, stripes...).

We can now compare in Table 4 the results of the two methods, for the same threshold. It seems that the discrimination of the HOG+SVM is less efficient

[4] The overlap ratio is defined as $OA = IS/US$ with IS the intersection surface and US the union surface.

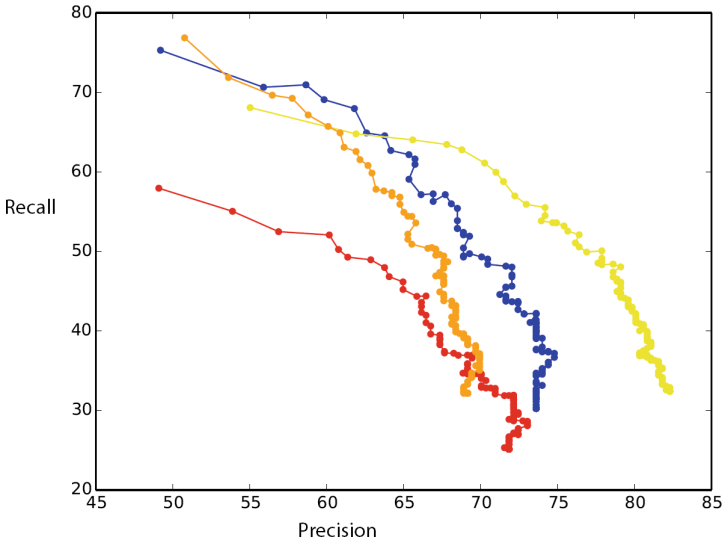

Fig. 6. ROC Curves of the Deep Learning method on the four test videos, with the threshold T as parameter. (Color figure online)

Table 3. Results with the Deep Learning method (T = 98 %)

Video	Precision	Recall	F-measure
1655	0.58	0.69	0.62
1654	0.68	0.63	0.65
1547	0.77	0.64	0.70
1546	0.60	0.52	0.55

Table 4. F-measure of the two methods with the same parameter

Video	F-measure from HOG+SVM	F-measure Deep Learning
1655	0.28	0.62
1654	0.24	0.65
1547	0.49	0.64
1546	0.14	0.55

than the CNN's. Indeed, the F-measure of the HOG+SVM is always below 49 % whereas it is always above 55 % for the CNN.

As we can observe in Fig. 7, the Deep Learning method approach efficiently recognizes fishes on different resolutions even when there is a strongly textured background and is able to distinguishes fishes which are close.

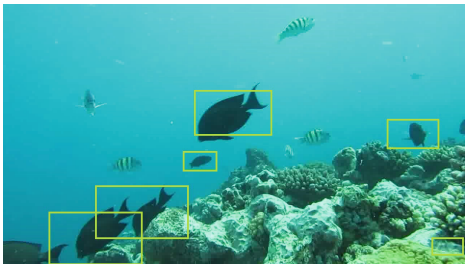

Fig. 7. The Deep Learning method succeeds in detecting fishes partially occluded by coral (bottom left)

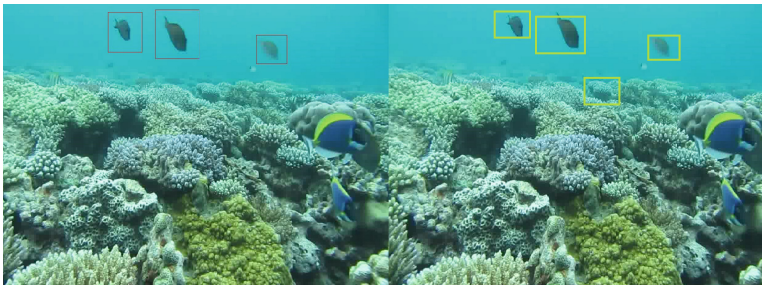

Fig. 8. A rock detected as a fish. On the left, the ground truth, on the right, the results of our processing

On the other hand, parts of the coral can be misclassified. In Fig. 8, we were able to detect the three fishes we were supposed to, but we also detected a part of the coral reef which presents features we can also find on fishes such as an enlighten top and a darker bottom, an oval shape, etc.

5 Future Work

In this paper, we have presented two methods to detect and recognize fishes in underwater videos.

When we apply the Deep Learning method directly on test thumbnails, which consists in recognizing if a thumbnail belong to a class, we reach a F-score of 98 %. We believe that the results can not really be improved as long as we keep the same network architecture. According to this, we focused our work on the post and pre-processing. The reduction of performance on a frame, in most case, comes from fishes which overlap or occlude and from confusion with the background. We tried to improve the method by adding three more classes and also through the use of a well chosen overlap decision.

At the moment, the Deep Learning method gives quite good results. A possible way to treat errors is to integrate the temporal aspect in a more advanced way by implementing a fish tracking algorithm.

Acknowledgement. This work has been carried out thanks to the support of the LabEx NUMEV project (n° ANR-10-LABX-20) funded by the "Investissements d'Avenir" French Government program, managed by the French National Research Agency (ANR). We thank very much Jérôme Pasquet and Lionel Pibre for scientific discussions.

References

1. Mallet, D., Pelletier, D.: Underwater video techniques for observing coastal marine biodiversity: a review of sixty years of publications (1952–2012). Fish. Res. **154**, 44–62 (2014)
2. Boom, B.J., Huang, P.X., Beyan, C., et al.: Long-term underwater camera surveillance for monitoring and analysis of fish populations. In: VAIB12 (2012)
3. Fisher, R.B., Chen-Burger, Y.-H., Giordano, D., Hardman, L., Lin, F.-P. (eds.): Fish4Knowledge: Collecting and Analyzing Massive Coral Reef Fish Video Data. ISRL, vol. 104. Springer, Heidelberg (2016)
4. Alsmadi, M.K.S., Omar, K.B., Noah, S.A., et al.: Fish recognition based on the combination between robust feature selection, image segmentation and geometrical parameter techniques using Artificial Neural Network and Decision Tree. J. Comput. Sci. **6**(10), 1088–1094 (2010)
5. Rova, A., Mori, G., Dill, L.M.: One fish, two fish, butterfish, trumpeter: recognizing fish in underwater video. In: Machine Vision Applications, pp. 404–407 (2007)
6. Spampinato, C., Giordano, D., Di Salvo, R.: Automatic fish classification for underwater species behavior understanding. In: Proceedings of the First ACM International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams, pp. 45–50 (2010)
7. Hearst, M.A., Dumais, S.T., Osman, E., et al.: Support vector machines. IEEE Intell. Syst. Appl. **13**(4), 18–28 (1998). MLA
8. Matai, J., Kastner, R., Cutter Jr., G.R.: Automated techniques for detection, recognition of fishes using computer vision algorithms. In: Williams, K., Rooper, C., Harms, J. (eds.) NOAA Technical Memorandum NMFS-F/SPO-121, Report of the National Marine Fisheries Service Automated Image Processing Workshop, 4–7 September 2010, Seattle, Washington (2010)
9. Shiau, Y.-H., Lin, S.-I., Chen, Y.-H., et al.: Fish observation, detection, recognition, verification in the real world. In: Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV), p. 1 (2012)
10. Blanc, K., Lingrand, D., Precioso, F.: Fish species recognition from video using SVM classifier. In: Proceedings of the 3rd ACM International Workshop on Multimedia Analysis for Ecological Data, pp. 1–6. ACM (2014)
11. Zhu, Q., Yeh, M.-C., Cheng, K.-T., et al.: Fast human detection using a cascade of histograms of oriented gradients. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1491–1498. IEEE (2006)
12. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, pp. 886–893. IEEE (2005)

13. Pasquet, J., Chaumont, M., Subsol, G.: Comparaison de la segmentation pixel et segmentation objet pour la détection d'objets multiples et variables dans des images. In: CORESA: COmpression et REprésentation des Signaux Audiovisuels, Reims (2014). (in French)
14. Das, S., Mirnalinee, T.T., Varghese, K.: Use of salient features for the design of a multistage framework to extract roads from high-resolution multispectral satellite images. IEEE Trans. Geosci. Remote Sens. **49**(10), 3906–3931 (2011)
15. Sun, X., Wang, H., Fu, K.: Automatic detection of geospatial objects using taxonomic semantics. IEEE Geosci. Remote Sens. Lett. **7**(1), 23–27 (2010)
16. Zhang, W., Sun, X., Fu, K., et al.: Object detection in high-resolution remote sensing images using rotation invariant parts based model. IEEE Geosci. Remote Sens. Lett. **11**(1), 74–78 (2014)
17. Zhang, W., Sun, X., Wang, H., et al.: A generic discriminative part-based model for geospatial object detection in optical remote sensing images. ISPRS J. Photogrammetry Remote Sens. **99**, 30–44 (2015)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
19. Atkinson, P.M., Tatnall, A.R.L.: Introduction neural networks in remote sensing. Int. J. Remote Sens. **18**(4), 699–709 (1997)
20. Schmidhuber, J.: Deep learning in neural networks: an overview. Neural Netw. **61**, 85–117 (2015)
21. Lecun, Y., Bottou, L., Bengio, Y., et al.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
22. Szegedy, C., Liu, W., Jia, Y.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
23. Joly, A., et al.: LifeCLEF 2015: multimedia life species identification challenges. In: Mothe, J., Savoy, J., Kamps, J., Pinel-Sauvagnat, K., Jones, G.J.F., SanJuan, E., Cappellato, L., Ferro, N. (eds.) CLEF 2015. LNCS, vol. 9283, pp. 462–483. Springer, Heidelberg (2015). doi:10.1007/978-3-319-24027-5_46