

# Towards Visual Loop Detection in Underwater Robotics using a Deep Neural Network

Antoni Burguera<sup>a</sup> and Francisco Bonin-Font<sup>b</sup>

Universitat de les Illes Balears, Ctra. Valldemossa Km. 7.5, Palma (Illes Balears), 07122, Spain

Keywords: Underwater Robotics, Loop Closing, Neural Network, SLAM.

Abstract: This paper constitutes a first step towards the use of Deep Neural Networks to fast and robustly detect underwater visual loops. The proposed architecture is based on an autoencoder, replacing the decoder part by a set of fully connected layers. Thanks to that it is possible to guide the training process by means of a global image descriptor built upon clusters of local SIFT features. After training, the NN builds two different descriptors of the input image. Both descriptors can be compared among different images to decide if they are likely to close a loop. The experiments, performed in coastal areas of Mallorca (Spain), evaluate both descriptors, show the ability of the presented approach to detect loop candidates and favourably compare our proposal to a previously existing method.

## 1 INTRODUCTION

One of the most important requirements to perform *Simultaneous Localization and Mapping* (SLAM) (Durrant-Whyte and Bailey, 2006) is the so called loop detection. This task, aimed at deciding if an area observed by the robot was previously visited, makes it possible to improve both the robot pose estimates and the map of the environment.

Since the use of cameras to perform SLAM has gained popularity in the last years (Taketomi et al., 2017), visual loop detection is nowadays one of the most prolific research fields in mobile robotics (Mur-Artal and Tardos, 2017). In this context, loop detection consists in deciding if two images depict overlapping parts of the environment.


The existing approaches to visual loop detection can be divided in three categories. The first one is based on matching local descriptors by means of robust techniques. For example, (Burguera et al., 2015) performs detection and matching of *Scale Invariant Feature Transform* (SIFT) between two images and searches for a sufficient number of consistent matches using *Random Sample Consensus* (RANSAC) to decide if they close a loop.


The second category relies on global image descriptors. These approaches build image descriptors

that can be easily compared to decide if they depict overlapping regions or not. As an example, (Negre-Carrasco et al., 2016) builds hash-based descriptors. The images whose descriptors are close in the Euclidean sense are more likely to close a loop. Other global descriptors, such as *Vector of Locally Aggregated Descriptors* (VLAD) (Jégou et al., 2010) or *Bag of Words* (BoW) (Ciarfuglia et al., 2012) have also proved to successfully detect visual loops.

The third category makes use of *Neural Networks* (NN). Some of these approaches (Merril and Huang, 2018) rely on the ability of *Convolutional Neural Networks* (CNN) to learn features thus removing the need for pre-engineered features such as SIFT. Other approaches propose a *Deep Neural Network* (DNN) to learn the parameters of a pre-engineered global descriptor (Arandjelovic et al., 2018).

Even though the aforementioned studies lead to high detection rates, they still have some important flaws. For example, methods relying on pre-engineered descriptors, either global or local, are usually constrained to certain types of environments since the used descriptors have been designed to detect some particularities that may not be present in all scenarios. Methods relying on DNN are more general in the sense that they can be trained for every specific environment where the robot has to be deployed. However, DNN require large amounts of data to be properly trained. They also need considerably high training times, and finally, even the loop detection

<sup>a</sup>  <https://orcid.org/0000-0003-2784-2307>

<sup>b</sup>  <https://orcid.org/0000-0003-1425-6907>

process can be long enough to be hardly affordable for real time operation. These problems are magnified in underwater robotics for the following reasons:

a) Underwater imagery (Bonin-Font et al., 2013) is affected by several problems such as reduced range, flickering, blur, haze, refraction, reflection or lack of illumination, among many others, that can be usually neglected in aerial or terrestrial robotics. This renders most of the existing methods based on pre-engineered descriptors less robust.

b) Even though DNN are good candidates to reduce the mentioned problem, the lack of NN pre-trained with underwater imagery and the technical difficulty to gather large underwater datasets makes it difficult to use them. Moreover, *Autonomous Underwater Vehicles* (AUV) tend to have limited computational power in order to reduce the payload and the battery usage and, so, on-line loop detection using DNN is usually too computationally expensive.

c) AUVs usually perform visual SLAM by navigating close to sea floor and using bottom-looking cameras. This configuration is not common in terrestrial robotics and most of the existing place recognition studies rely on forward-looking cameras. These differences pose additional problems if they are not properly taken into account. For example, two images closing a loop gathered by a bottom looking camera are likely to be significantly rotated one with respect to the other whilst rotation in the image plane is extremely unusual when using forward looking cameras. This complicates taking profit of terrestrial datasets to train these NN and jeopardizes the use of transfer learning from other NN trained with terrestrial data.

d) Performing large scale missions with an AUV is especially problematic mainly because of their strict dependence on batteries that cannot be recharged during the mission unless an expensive infrastructure, such as a support ship, is available. The common approach to deal with these scenarios is to divide the large mission in a set of smaller missions called sessions. In this context, called *Multi-Session SLAM* (Burguera and Bonin-Font, 2019), the AUV must detect loops not only within each session but also among sessions. Since the time between sessions can be arbitrarily large, ranging from a hours to months, the visual conditions and even the environment itself can be significantly different, thus requiring particularly robust methods.

This paper constitutes a first step towards visual loop detection in underwater scenarios dealing with the aforementioned problems and assuming an AUV with a bottom-looking camera. To this end, we explore the use of a NN simple enough to be easily train-

able with a small amount of images and fast enough to be used on-line in an AUV with low computational capabilities. We also explore the use of a pre-engineered global image descriptor to guide the training process. More specifically, our proposal is based on an autoencoder architecture with the decoder part changed to fully connected layers providing an output compatible with the mentioned global descriptor.

The NN is trained to output, for a given image, a descriptor that is similar to the descriptor of another image that closes a loop with the former image. After training, both the learnt descriptor and the latent representation in the inner convolutional layers can be used to fast and reliably compare images and detect loops.

The experimental results, performed in coastal areas of Mallorca (Spain), show the ability of this proposal to detect loops, compares it favourably to a previously existing method and also shows the superiority of the NN with respect to the raw use of the pre-engineered descriptor.

## 2 OVERVIEW

Image autoencoders (Rezende et al., 2014) are NN aimed at providing an output identical to the input image. These NN operate in two main steps. The first one, performed by the so called encoder, is in charge of mapping the input image into a latent space of smaller dimensionality. The second step, performed by the decoder, maps the latent space back to the original image space, thus increasing the dimensionality.

Even though the output of an autoencoder is not particularly useful, since it mimics the input, the learned latent representation is said to contain useful image features. Some implementations, such as the *Variational Autoencoders* (VAE), are aimed at learning semantically meaningful latent representations (Kingma and Welling, 2014).

Our proposal is to take advantage of these ideas to learn the features that define loop closings instead of those that define the images individually. Once learned, these features can be used to compare two images and decide if they depict overlapping areas or not. Accordingly, our proposal is similar to the use of global image descriptors, though the descriptor itself is learned instead of pre-engineered, thus providing a more general solution adaptable to different environments.

Since the goal is to learn loop closings, instead of using the same image as input and target our proposal is that input and target come from two different loop closing images. Contrarily to standard autoen-

coders, the image space is not a good choice for the NN output since it is not invariant to rotation or scaling, and barely robust in front of shifting. These transformations being the most relevant in visual loops using bottom-looking cameras an alternative space is required. In consequence, it is necessary to define a way to represent an image that is robust in front of these transformations. The proposed representation, called *Global Image Descriptor* (GID) is described in Section 3.

Using the GID makes it necessary to change the standard autoencoder architecture to be usable with it. Overall, the idea is to define an autoencoder based architecture able to learn the latent representation of loops by using one image as input and targeting the GID of another image that closes a loop with it, thus input and output spaces being different. This architecture is described in Section 4.

### 3 THE GLOBAL IMAGE DESCRIPTOR

The GID is not aimed at robust loop detection. Instead, its goal is to provide an appropriate space where two loop closing images can be projected so that both projections are comparable. This space is necessary, since, as it has been mentioned before, the image space used by standard autoencoders is not useful to perform loop detection.

In this way, the NN will be trained to mimic the GID of one image using another image that closes a loop with it as input. By targeting the GID, the NN will learn useful features of loop closing images similarly to autoencoders which learn useful features of the images themselves. Thus, the GID is only computed to train the system and is not required after training.

For a GID to be useful in this context, it has to meet two main requirements. On the one hand, it must have a fixed length for images of the same size, since it will be targeted by a NN. On the other hand, it must be comparable between images, so that a loss function can be defined to train the network. The *Histogram of Oriented Gradients* (HOG) meets these two requirements and that is why it has been successfully used to perform place recognition (Merril and Huang, 2018).

However, when it comes to underwater imagery using a bottom-looking camera, an additional requirement arises: the GID must be invariant to large rotations and translations, significant scaling and, in case of Multi-Session SLAM, to changes in the environmental conditions. Unfortunately, HOG does not meet these requirements and a different GID has to be

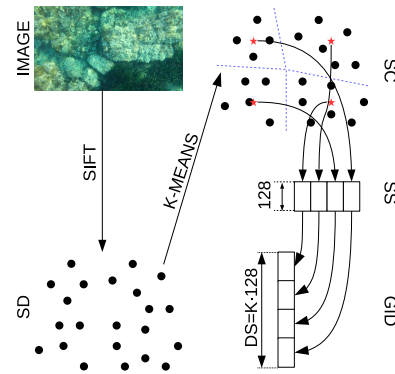


Figure 1: Summary of the GID building process.

used. Our proposal is to build the GID upon SIFT, which has shown to be invariant to all these changes.

Directly using SIFT is not possible since the two aforementioned main requirements are not met. For example, even though a SIFT descriptor has a fixed length, the number of descriptors can change from one image to another, being thus difficult to create a fixed length vector. Also, SIFT detectors do not find the features in any particular order. Because of that, although individual descriptors can be compared using Euclidean distance, an arbitrary ensemble of descriptors is not. Accordingly, a method to deal with these two issues is required.

Several approaches (Perronnin and Dance, 2007; Jégou et al., 2010), mostly of them based on the Fisher kernel (Jaakkola and Haussler, 1999), exist to achieve this goal. Our proposal is to aggregate the SIFT descriptors based on a distance criterion and sort the clusters depending on the number of corresponding descriptors. This process, summarized in Figure 1, is detailed next.

Given one image, the first step is to compute the SIFT descriptors  $SD = \{d_0, d_1, \dots, d_n\}$ . Each  $d_i$  is a vector of fixed size (128) but the number of descriptors  $n$  changes from one image to another. These descriptors can be compared using the L2-norm so that, ideally,  $\|d_i - d_j\| \simeq 0$  if and only if the regions around features  $i$  and  $j$  depict a visually similar region of the environment.

Afterwards, a codebook  $SC = \{c_0, c_1, \dots, c_K\}$  of  $K$  visual words is built by applying K-Means to  $SD$ . Each  $c_i$  is the centroid of the  $i$ -th cluster found by K-Means, which constitutes a representative of the visual appearance shared between the descriptors belonging to the cluster

Since each cluster contains descriptors corresponding to visually similar regions, the number of features assigned to the cluster represents the visual importance of the corresponding centroid. Because of that, our proposal is to sort the centroids in  $SC$  ac-

ording to the number of descriptors assigned to the corresponding cluster. Let  $SS$  denote the sorted  $SC$ . In this way, similar images will lead similarly sorted centroids. Let the GID be the result of flattening  $SS$  into a 1D tensor of size  $DS = K \cdot 128$ . In this way, since each descriptor and thus each centroid can be compared using the L2-norm, two GID coming from two different images can also be compared using the Euclidean distance.

## 4 THE NEURAL NETWORK

The proposed architecture, which is based on an autoencoder and follows the ideas by (Merril and Huang, 2018), is summarized in Figure 2-a. As it can be observed, our proposal has a set of convolutional and pooling layers that define the encoder aimed at reducing the data dimensionality. In particular, we use sets of convolutional layers with sigmoid activation functions and maxpooling.

The decoder significantly differs from the ones in autoencoders. Since the NN output is not an image but a GID, the decoder does not perform transposed convolutions and pooling. Instead, it goes from the latent representation to the GID space through a set of dense layers with sigmoid activation functions, the last one having the size of the GID.

The process of training the NN is summarized in Figure 2-b. The ground truth is composed of couples of underwater images closing loops between them. One of these two images is randomly chosen to be the NN input. Let this image be named  $I_{NN}$ . The other image, named  $I_{GID}$ , is used to compute the GID. This random selection prevents training biases and reduce overfitting since the training sets will slightly differ between epochs.

The training is aimed at reconstructing the GID corresponding to  $I_{GID}$  given  $I_{NN}$ . Let the GID reconstructed by the NN be referred to as the *Learned GID* (LGID). The L2 loss function is used to compare the GID and the LGID since the Euclidean distance is a good metric to compare two GID.

Once the system is trained, it can be used to build either the LGID or the *Learned Features* (LF). The LF is the output of the encoder part and constitutes the latent space, as shown in Figure 2-a. To ease notation, let  $D_i$  denote the LGID or the LF, indistinctly, obtained from image  $I_i$ . The effects of using the former or the latter will be experimentally assessed in Section 5.

The Euclidean distance between  $D_i$  and  $D_j$  provides information about how likely is for  $I_i$  and  $I_j$  to close a loop. When using LF as descriptor, the

Table 1: Number of database images, query images and loops each dataset.

	Database	Query	Loops
<b>Dataset 1</b>	183	24	34
<b>Dataset 2</b>	177	25	26
<b>Dataset 3</b>	244	24	26

Euclidean distance is computed by first flattening LF into a 1D tensor. Deciding loop closings solely with this information would require the existence of a threshold  $\delta$  so that  $I_i$  and  $I_j$  close a loop if and only if  $\|D_i - D_j\| \leq \delta$ . Instead of using such threshold, our proposal takes advantage of how loop closings are used in visual SLAM.

Basically, when performing visual SLAM, the most recent image is matched against all the previously gathered images. Our proposal is, thus, to select a subset of the previously gathered images as loop candidates and confirm these loops in a posterior step. The loop confirmation step is out of the scope of this paper.

The whole process is divided in two steps. The first step, called *description*, builds  $D_i$  for all the gathered images. We distinguish between  $D_t$ , which comes from the most recent image or *query image*, and  $D_{0:t-1}$  which come from the remaining images or *database* images. Due to the incremental nature of SLAM, the query image will become a database image in further steps. Thus, our proposal only requires computing  $D_t$ , since  $D_{0:t-1}$  are already computed from previous steps.

Afterwards, the *candidate selection* process is performed. During this step, the query image is compared to all the database images by computing the Euclidean distance between  $D_t$  and all the  $D_{0:t-1}$ . The  $N$  database images leading to smaller Euclidean distances are selected and constitute the set  $C_t = \{C_0, C_1, C_{N-1}\}$  of loop candidates. That is,  $C_t$  contains the  $N$  images that are more likely to close a loop with  $I_t$ .

## 5 EXPERIMENTAL RESULTS

Three datasets of RGB images have been gathered in coastal areas of Mallorca (Spain) using an AUV with a bottom looking camera. Each dataset is divided in two parts: database images and query images. There are no loop closing images within each of these parts, but each query image closes a loop with at least one database image. The loop closings have been manually identified and constitute the ground truth. Table 1 shows the number of images and loop closings in each dataset. All the images are resized to a resolution of

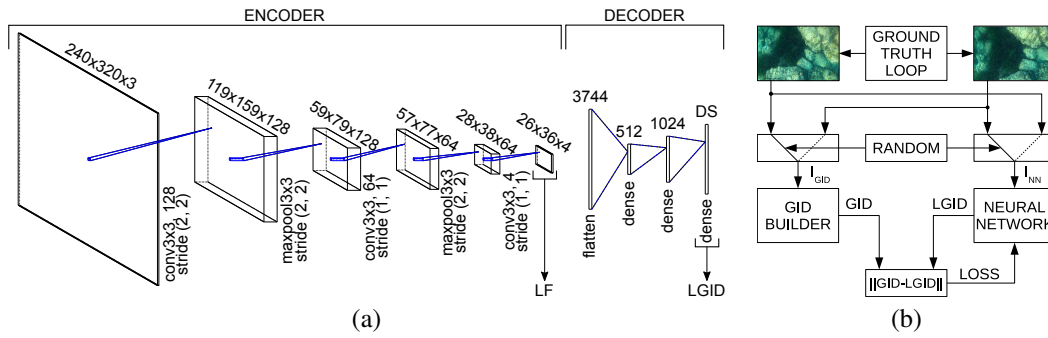


Figure 2: (a) The Neural Network architecture (b) The training process.

320×240 pixels prior to their use.

The two different NN outputs, namely LF and LGID as shown in Figure 2-a, have been tested and evaluated. Our proposal has been compared to the *Deep Loop Closure* (DLC) approach by (Merril and Huang, 2018) and to the direct use of the GID proposed in Section 3. In this case, loop candidates are obtained by directly computing distances between GIDs without using the NN.

As for the DLC, the original algorithm has been slightly modified to get better results in front of underwater images and, thus, to provide a fair comparison. In particular, the DLC synthetic loop generation has been changed to the same supervised approach used with our methods so that it can be trained using exactly the same data than our proposal. Taking into account that the DLC is also autoencoder based, these changes make it possible to also test LF and LGID. Let DLC-LF and DLC-LGID to refer to these two cases.

The system has been trained, validated and tested using all the valid combinations of the three datasets. The only hyperparameter that has been tuned during validation is the number of epochs. Let the notation  $T_xV_yS_z$  denote a system trained with dataset  $x$ , validated with dataset  $y$  and tested with dataset  $z$ . Only the combinations where  $x$ ,  $y$  and  $z$  are different are considered valid.

In order to evaluate the quality of the loop candidates we proceeded as follows. For each query image  $I_t$  in each dataset the set of loop candidates  $C_t$  has been computed using the two variations of our approach (LGID and LF), the two variations of DLC (DLC-LGID and DLC-LF) and GID as described before.

As an example, Figure 3 shows some of the candidate loops in each dataset. The first row shows a query image of each dataset whilst the remaining rows depict the first three candidates found by our approach. In this example, the LF has been used to select the candidates and, in each case, the NN was

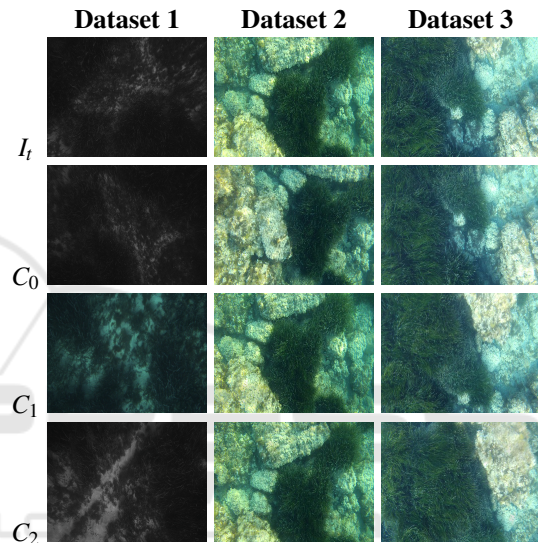


Figure 3: Examples of candidate loops in each dataset.

trained and validated using datasets other than the one being tested. As it can be observed, actual loops are within the candidate sets in all cases.

To quantify the quality of the loop candidates, the number  $N$  of items in  $C_t$  has been set to values ranging from 1% to 100% of the number of database images in the dataset. In each case the percentage of query images for which at least one actual loop was in the candidate set has been computed. Let this percentage be referred to as the *hit ratio*.

Figure 4 shows some of the obtained hit ratios as a function of the percentage of database images used to size  $C_t$ . The labels in the examples corresponding to our proposal and to DLC specify the training, validation and test sets using the aforementioned  $T_xV_yS_z$  notation. The examples corresponding to GID do not use that notation since GID is not trained nor validated, and thus only the tested dataset is specified as  $S_x$ ,  $x$  being the dataset number. Results using GID are significantly worse than those using the deep learning approaches. This suggests that even though GID

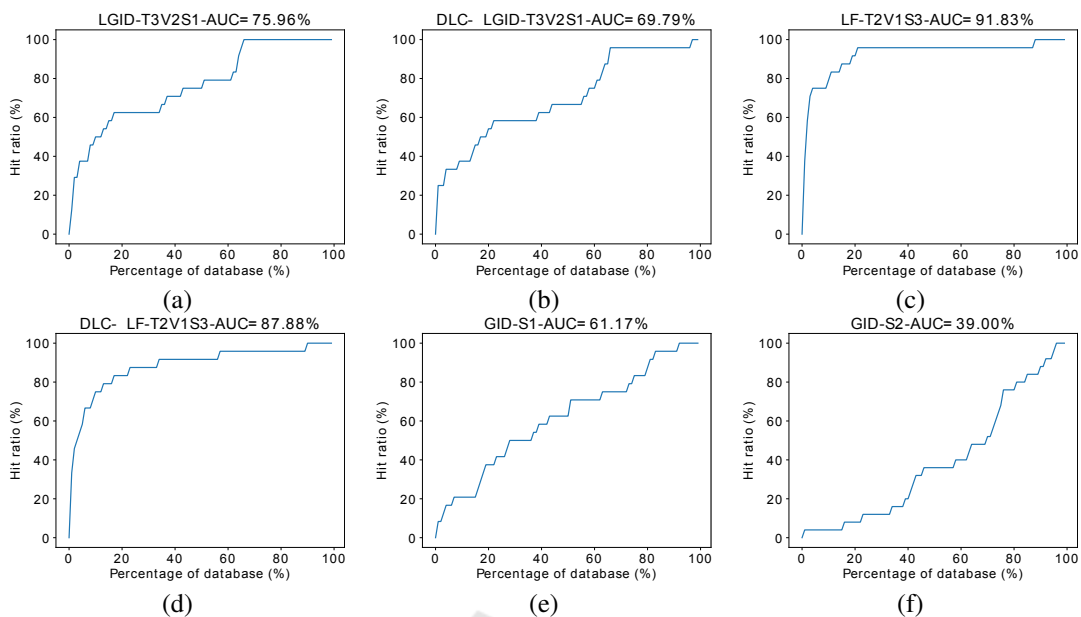


Figure 4: Examples of hit ratio evolution of (a) our proposal using LGID, (b) DLC using LGID, (c) our proposal using LF, (d) DLC using LF, (e)-(f) direct use of GID.

Table 2: AUC values of our proposal for all the tested configurations. The gray cells correspond to the cases in which our approach surpasses DLC.

	LGID	DLC-LGID	LF	DLC-LF
<b>T1V2S3</b>	88.75%	88.08%	88.08%	90.46%
<b>T1V3S2</b>	83.60%	80.84%	86.16%	87.56%
<b>T2V1S3</b>	88.17%	86.54%	91.83%	87.88%
<b>T2V3S1</b>	73.58%	78.71%	73.79%	77.21%
<b>T3V1S2</b>	88.24%	84.60%	88.44%	87.56%
<b>T3V2S1</b>	75.96%	69.79%	75.88%	69.92%
<b>Average</b>	83.05%	81.43%	84.03%	83.43%

is not well suited to find loops, it is to provide useful information to train a NN to achieve this goal.

Let us define the *Area Under the Curve* (AUC) as the percentage of the whole space of possibilities below the hit ratio curves. In the previous examples, the AUC is shown on the top of each graph. Since the hit ratio always increases with the size of the candidate set, approaches with large hit ratios for small candidate sets will be responsible for large AUC values. Having large hit ratios within small candidate sets means that the candidate set has been accurately constructed. Accordingly, the AUC is a good way to measure the quality of the candidate set.

Table 2 shows the AUC corresponding to our proposal and to DLC for each valid combination of training, validation and testing. Overall, our proposal surpasses DLC in average, being the improvements more clear when using LGID. Also, it can be observed that LF, both using our proposal and DLC, leads to better results than LGID. This suggests that the NN itself is

Table 3: AUC values of the tested approaches. The gray cells emphasize the best result for each dataset.

	Dataset 1	Dataset 2	Dataset 3	Average
LGID	75.77%	85.92%	88.46%	83.38%
LF	74.84%	87.30%	89.96%	84.03%
DLC-LGID	74.25%	82.72%	87.31%	81.43%
DLC-LF	73.57%	87.56%	89.17%	83.43%
GID	61.17%	39.00%	45.75%	48.64%

more important than the specific GID used to train it.

By aggregating the previous results per dataset, it is possible to compare them to the AUC corresponding to the direct use of the GID. Table 3 summarizes these results. As it can be observed, LF is the best method in average and the GID alone leads to poor results, thus being useless to search loops by itself.

## 6 CONCLUSION AND FUTURE WORK

This paper constitutes a first step towards the use of Deep Neural Networks to fast and robustly detect underwater visual loops. The proposed approach follows the structure of an autoencoder and replaces the decoder by a set of fully connected layers. This change makes it possible to use a global image descriptor, built upon clusters of local SIFT features, to guide the training process.

Once trained, the NN builds two different descriptors of the input image. One of these descrip-

tors, called LF, is the encoder output whilst the other, called LGID, is the decoder output. Both descriptors can be compared among different images to decide if they are likely to close a loop. The experimental results show the ability of the presented approach to detect loop candidates and compare favourably to a previously existing method.

Since our proposal has shown to be trainable with a small set of images and the architecture is simple enough to provide on-line sets of loop candidates, it constitutes a promising approach to be embedded into a full underwater visual SLAM system. Even though it cannot replace a full SLAM loop closing layer, since it does not compute the relative motion between images, it can strongly reduce the computational load of such module by feeding it only with images that most likely will close a loop.

Accordingly, our lines of future research are as follows. First, we are now working on a strategy to confirm or deny the loop candidates as well as to compute the relative motion between the confirmed loops. In this way, our proposal could be embedded into a full SLAM system. Second, even though small datasets have shown to be sufficient to reach good results, larger datasets would probably lead to better candidate sets. For this reason, we are also working on a method to synthetically generate loops from underwater imagery. This would constitute a weakly supervised approach and would allow training the NN with arbitrarily large datasets. Our final goal would be to integrate the whole loop detection not only into a SLAM system but also into a Multi-Session SLAM system, which will definitely prove the ability of our proposal to detect loops in a really challenging scenario.

## ACKNOWLEDGEMENTS

This work is partially supported by the Spanish Ministry of Economy and Competitiveness under contract DPI2017-86372-C3-3-R (AEI,FEDER,UE).

## REFERENCES

- Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., and Sivic, J. (2018). NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1437–1451.
- Bonin-Font, F., Burguera, A., and Oliver, G. (2013). New solutions in underwater imaging and vision systems. In *Imaging Marine Life: Macrophotography and Microscopy Approaches for Marine Biology*, pages 23–47.
- Burguera, A. and Bonin-Font, F. (2019). A trajectory-based approach to multi-session underwater visual slam using global image signatures. *MDPI Journal of Marine Science and Engineering*, 7(8).
- Burguera, A., Bonin-Font, F., and Oliver, G. (2015). Trajectory-based visual localization in underwater surveying missions. *Sensors (Switzerland)*, 15(1):1708–1735.
- Ciarfuglia, T. A., Costante, G., Valigi, P., and Ricci, E. (2012). A discriminative approach for appearance based loop closing. In *IEEE International Conference on Intelligent Robots and Systems*, pages 3837–3843.
- Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping (SLAM): part I The Essential Algorithms. *Robotics & Automation Magazine*, 2:99–110.
- Jaakkola, T. S. and Haussler, D. (1999). Exploiting generative models in discriminative classifiers. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 487–493.
- Jégou, H., Douze, M., Schmid, C., and Pérez, P. (2010). Aggregating local descriptors into a compact image representation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3304–3311.
- Kingma, D. P. and Welling, M. (2014). Auto-Encoding Variational Bayes (VAE, reparameterization trick). *ICLR 2014*, (MI):1–14.
- Merril, N. and Huang, G. (2018). Lightweight Unsupervised Deep Loop Closure. In *Robotics: Science and Systems*.
- Mur-Artal, R. and Tardos, J. D. (2017). ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262.
- Negre-Carrasco, P. L., Bonin-Font, F., and Oliver-Codina, G. (2016). Global image signature for visual loop-closure detection. *Autonomous Robots*, 40(8):1403–1417.
- Perronnin, F. and Dance, C. (2007). Fisher kernels on visual vocabularies for image categorization. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.
- Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic Back-propagation and Variational Inference in Deep Latent Gaussian Models. *Proceedings of The 31st ...*, 32:1278–1286.
- Taketomi, T., Uchiyama, H., and Ikeda, S. (2017). Visual SLAM algorithms: a survey from 2010 to 2016. *IPSA Transactions on Computer Vision and Applications*, 9(1).