**Preprints of the 21st IFAC World Congress (Virtual)**
**Berlin, Germany, July 12-17, 2020**

# Visual Loop Detection in Underwater Robotics: an Unsupervised Deep Learning Approach ⋆

**Antoni Burguera** * **Francisco Bonin-Font** *

* *Universitat de les Illes Balears, Ctra. Valldemossa Km. 7.5 Palma (Illes Balears), 07122 Spain*
*e-mail: {antoni.burguera,francisco.bonin}@uib.es.*

**Abstract:** This paper presents a novel Deep Neural Network aimed at fast and robust visual loop detection targeted to underwater images. In order to help the proposed network to learn the features that define loop closings, a global image descriptor built upon clusters of local SIFT descriptors is proposed. Also, a method allowing unsupervised training is presented, eliminating the need for a hand-labelled ground truth. Once trained, the Neural Network builds two descriptors of an image that can be easily compared to other image descriptors to ascertain if they close a loop or not. The experimental results, performed using real data gathered in coastal areas of Mallorca (Spain), show the validity of our proposal and favourably compares it to previously existing methods.

*Keywords:* Robot vision, underwater robotics, neural networks, loop detection, SLAM

## 1. INTRODUCTION

*Simultaneous Localization and Mapping* (SLAM), which is aimed at estimating the pose of a mobile robot while it builds a map of the environment, is one of the most important tasks in mobile robotics nowadays (Durrant-Whyte and Bailey, 2006). A fundamental problem in SLAM is the one of deciding whether the area observed by the robot was previously visited or not. Solving this problem, known as loop detection, is crucial to improve the robot pose estimates as well as the constructed map.

The use of cameras to detect loops has gained popularity in the last years (Davison et al., 2007), nowadays being the most popular approach (Mur-Artal and Tardos, 2017). Some of the existing studies rely on matching local descriptors. For example, Burguera et al. (2015) make use of *Scale Invariant Feature Transform* (SIFT) feature detection and matching combined with RANSAC to decide if two images depict an overlapping region of the environment and, thus, constitute a loop. Other approaches rely on global descriptors. Roughly speaking, they build image descriptors that can be easily compared to decide if there is overlapping between the corresponding images. For example, Negre-Carrasco et al. (2016) build hash-based descriptors called *Hash based Loop Closure* (HALOC) and compares them using Euclidean distance to decide if two images overlap or not. Other approaches based on global descriptors are *Vector of Locally Aggregated Descriptors* (VLAD) (Jégou et al., 2010) or *Bag of Words* (BoW) (Gálvez-López and Tardós, 2012).

When it comes to underwater environments, visual loop detection is affected by several problems that exist only up to a much lesser extent in terrestrial and aerial robotics (Bonin-Font et al., 2013; Hong et al., 2016). Reduced range, flickering or bad illumination, among many other, are the reasons why underwater visual loop closing is particularly challenging and has to rely on robust techniques.

Loop detection using *Deep Neural Networks* (DNN) has shown to be particularly robust, mainly because these methods learn the image descriptions depending on the environment in which they will be deployed instead of relying on general purpose pre-engineered features. For example, Arandjelovic et al. (2018) use a DNN to automatically parametrize the VLAD global image descriptors. Other researchers such as Merril and Huang (2018) show that the convolutional layers of a properly trained *Convolutional Neural Network* (CNN) can be used as image descriptors.

In spite of their exceptionally good results, these methods cannot be directly applied to underwater visual loop detection for three main reasons. First, most existing *Neural Networks* (NN) used to perform place recognition are slow to perform feature extraction or querying. Speed is of paramount importance when it comes to AUV because battery and space limitations usually constrain the computational power. Second, they need large amounts of training data, and this is particularly difficult in underwater environments due to the required equipment. Third, most of the existing approaches rely, up to a certain extent, on pre-trained networks (Razavian et al., 2014; Sünderhauf et al., 2015). In our case, transfer learning is not possible. On the one hand, because underwater scenarios are radically different to the kind of images used to train well known networks. On the other hand, because most of the pre-trained networks assume a forward looking camera

⋆ This work is partially supported by Ministry of Economy and Competitiveness under contracts DPI2017-86372-C3-3-R (AEI,FEDER,UE) and TIN2014-58662-R (AEI,FEDER,UE).

whilst most AUV make use of bottom-looking cameras. This difference in the camera orientation leads to viewpoint changes for which most pre-trained networks are not prepared.

This paper is focused on visual loop detection in underwater scenarios and centers its attention on the above mentioned problems. To this end, we propose an autoencoder based NN with low latency. The NN is trained using a pre-engineered description method to guide it when learning the best features to detect loops. Since our proposal is autoencoder based, the learned representation is smaller than the image itself, thus reducing the storage problems in the AUV on-board computers. Additionally, since the proposed network has significantly less parameters than state of the art NN, the required number of images to train the system, as well as the training time, is dramatically reduced.

As for training, an unsupervised approach not requiring hand-labelled loops is presented. This approach, which generates synthetic loops from real data, not only removes the need to hand-label couples of images but it also makes training the system with few real images possible.

## 2. OVERVIEW

An image autoencoder (Jimenez-Rezende et al., 2014) is a NN composed of two parts: an encoder, which maps the input image into a latent space and a decoder, which decodes it back to the original image space. The encoder is a succession of convolutional and pooling layers, each layer reducing the dimensionality of the previous one. The decoder inverts that process performing upscaling and interpolation. Autoencoders are trained using the same image both as input and as target data. In this way, autoencoders can learn latent representations of the input data which are smaller than the input data itself.

Contrarily to autoencoders, our goal is not to learn latent representations of images by themselves but of loops. This means that, instead of input and target data being the same image, they should come from two different images closing a loop. In this case the image space is not a good choice for the NN output since it is not invariant to rotation or scaling and barely to shifting, and these transformations are precisely the most relevant when using a bottom looking camera. Thus, a way to represent an image that is robust in front of these transformations is required. Let this representation be referred to as the *Global Image Descriptor* (GID). The process to build the GID is described in Section 3.

Our proposal is to adapt the standard autoencoder architecture so that it learns the latent representation of loops by targeting the GID of one of the loop closing images using the other image as input. This architecture is presented in Section 4.

Training this architecture requires couples of loop closing images. When it comes to underwater scenarios, it is difficult to obtain such training data and, more important, it is particularly tedious and error prone to label the loops since most of the images look similar to humans. That is why we propose a method to synthetically generate loop closing images in this kind of scenarios. A description of this method as well as a description of how to train the NN are provided in Section 5.

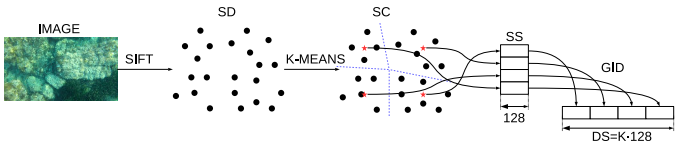


Fig. 1. Summary of the GID building process.

After training, our NN can be used to describe an image without having to compute the GID. This description can be compared to other images descriptors in order to decide whether they possibly close a loop or not. That is, a set of loop candidates is constructed. This process is detailed in Section 6.

## 3. THE GLOBAL IMAGE DESCRIPTOR

The main goal of the GID is not to be able to robustly discriminate between loop closing and non loop closing images but to provide an alternative to the image space that can be used as target data when training the proposed NN. Thus, the GID is only computed to train the system and is not required after training.

Merril and Huang (2018) proposed a GID based on the *Histogram of Oriented Gradients* (HOG) since it has a fixed length for images of the same size and can be easily compared using Euclidean distance. Also, HOG is robust enough to perform place recognition from terrestrial forward looking cameras since only changes in scale and very small rotations in the image plane appear. Unfortunately, this is not the case of underwater bottom looking cameras.

In the case of bottom looking cameras attached to an AUV, images closing a loop are often largely rotated one with respect to the other since the camera observes a plane parallel to the robot motion. Also, loop closing images can have different scale and illumination because the AUV can navigate up and down. Accordingly, we need a GID that is robust to changes in scale, rotation, illumination as well as to image shifts. Our proposal is to build the GID upon SIFT, which has shown to be invariant to all these changes.

Even though a SIFT descriptor has a fixed length, the number of descriptors changes from one image to another. Moreover, since SIFT is aimed at providing local descriptions of parts of an image, the SIFT descriptors are not found in any particular order. That is, SIFT descriptors cannot be directly used to describe an image since the GID would have different lengths depending on the image and, more important, the random order in which they are found would make it impossible to compare different GID. Accordingly, a method to deal with these two issues is required.

Several approaches exist to achieve this goal (Perronnin and Dance, 2007; Jégou et al., 2010), most of them based on the Fisher kernel (Jaakkola and Haussler, 1999). Our proposal is to aggregate the SIFT features based on a distance criterion in the descriptor space and sort the clusters depending on the number of corresponding

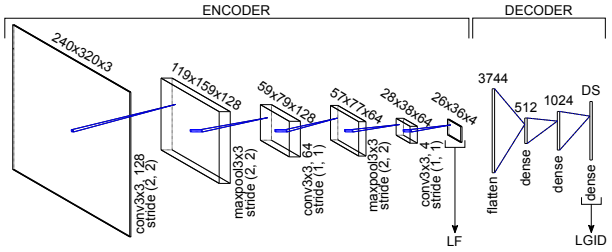


Fig. 2. The Neural Network architecture

descriptors. This process, summarized in Figure 1, is detailed next.

Given one image, the first step is to compute the SIFT descriptors $SD = \{d_0, d_1, \cdots, d_n\}$. Each $d_i$ is a vector of fixed size 128. However, 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 $SC = \{c_0, c_1, \cdots, c_K\}$ of $K$ visual words is built by applying K-Means to $SD$. Since the descriptors are comparable using the L2-norm, K-Means has to be applied using the Euclidean distance. 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. That is, the more descriptors assigned to a cluster, the more relevant the cluster is. Because of that, our proposal is to sort the centroids in $SC$ according to the number of descriptors assigned to the corresponding cluster. Let $SS$ denote the sorted $SC$. In this way, similar images will not only lead to similar descriptors but also to similarly sorted centroids. Since each descriptor and thus each centroid can be compared using the L2-norm, two $SS$ coming from two different images can also be compared using the Euclidean distance.

To facilitate further usage of $SS$ in a NN, it is normalized to the range [0,1] and converted to a 1D tensor by flattening the data. This tensor of size $DS = K \cdot 128$ constitutes the GID. Overall, two images that are rotated, scaled or shifted one with respect to the other will produce GIDs that are similar in Euclidean terms.

## 4. THE NEURAL NETWORK

The proposed architecture, which is based on an autoencoder, is summarized in Figure 2. As it can be observed, our proposal has an encoder which is a set of convolutional and pooling layers, 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, in our case, 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 fully

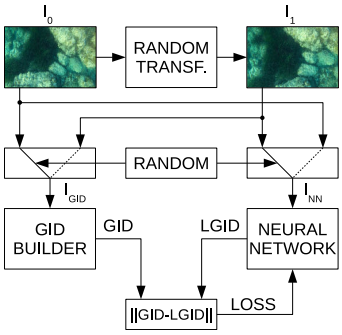


Fig. 3. The training process.

connected layers with sigmoid activation functions, the last one having the size of the GID.

After training, the NN could be used in two different ways. On the one hand, the encoder output (that is, the latent representation) could be used to compare images. Let the encoder output be referred to as the *Learned Features* (LF). On the other hand, the output of the last dense layer could also be used to the same end. Let this output be referred to as the *Learned Global Image Descriptor* (LGID). Both the LF and the LGID should be similar between images closing a loop. This will be experimentally assessed in Section 7.

## 5. UNSUPERVISED TRAINING

Training the proposed NN requires pairs of loop closing images. One of the images in each pair will be the NN input whilst the other is used to compute its GID, which constitutes the target data. For the NN to be properly trained, a large number of loop closing images is required. The main problem here is that hand labelling the training data is tedious and error prone. That is why a method to build synthetic loops is advisable, thus making our system unsupervised.

The overall training process, including the synthetic loop generation, is illustrated in Figure 3. Given one underwater image $I_0$, our goal is, first, to build an image $I_1$ that depicts the same part of the ocean floor but from a different viewpoint. Taking into account that our proposal is to deal with underwater bottom looking cameras attached to an AUV, this change in viewpoint will result only in rotations over the image plane as well as scaling and shifting. Accordingly, $I_1$ is built by applying a random rotation, scaling and shifting to $I_0$. Random changes in brightness and contrast are also performed to simulate different illumination conditions. Mirroring is applied to fill the pixels in $I_1$ that have no corresponding pixel in $I_0$ after the transformation.

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 since in this way neither the synthetic nor the original image will always be used for the same purpose.

The training is aimed at reconstructing the GID corresponding to $I_{GID}$ given $I_{NN}$. As stated in Section 3, the Euclidean distance is a good metric to compare the

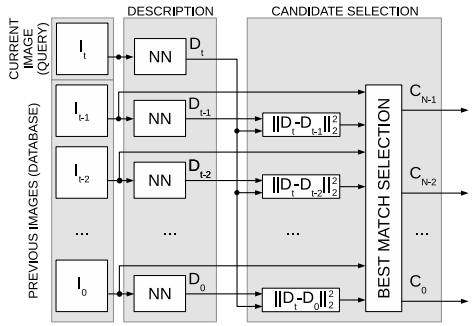


Fig. 4. Online usage of the proposed architecture.

proposed GID. For this reason, the L2 loss function is used to compare the GID with its reconstruction LGID.

## 6. OBTAINING LOOP CANDIDATES

Once the system is trained, it can be used to build either the LGID or the LF as shown in Figure 2. To ease notation, let $D_i$ denote the LGID or the LF, indistinctly, obtained from image $I_i$.

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. However, 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 fixed size of the previously gathered images as loop candidates and confirm these loops in a posterior step.

The whole process is summarized in Figure 4. 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 the smallest Euclidean distances are selected and constitute the set $\mathcal{C}_t = \{C_0, C_1, \cdots, C_{N-1}\}$ of candidates to close a loop with the query (loop candidates). That is, $\mathcal{C}_t$ contains the $N$ images that are likely to close a loop with $I_t$.

## 7. EXPERIMENTAL RESULTS

In order to evaluate our proposal, 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. 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 320×240 pixels prior to their use.

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

| | **Database** | **Query** | **Loops** |
|---|---|---|---|
| **Dataset 1** | 183 | 24 | 34 |
| **Dataset 2** | 177 | 25 | 26 |
| **Dataset 3** | 244 | 24 | 26 |

Table 2. The four tested variations of the NN approach.

| **Name** | **Training** | **NN output** |
|---|---|---|
| **SLGID** | Supervised | LGID |
| **SLF** | Supervised | LF |
| **ULGID** | Unsupervised | LGID |
| **ULF** | Unsupervised | LF |

Four different variations of our proposal have been tested. These variations are the combinations of using different training methods and different NN outputs. As for training methods, we have tested both the unsupervised approach described in Section 5 and a supervised approach that uses only the hand labelled loops in the datasets as training data. As for NN outputs, as stated previously, both the encoder output (LF) and the decoder output (LGID) have been tested. Table 2 summarizes the four variations and defines a name to ease further references.

Our proposal has been compared to the direct use of the GID proposed in Section 3 by directly computing distances between GIDs without using the NN. It has also been compared to the *Deep Loop Closure* (DLC) approach by Merril and Huang (2018). To provide a fair comparison, the DLC synthetic loop generator has been changed to our proposal (Section 5) since the original DLC method assumed a forward looking camera. Thanks to that, it is possible to test the four variations shown in Table 2 also with DLC. To name these variations, the prefix DLC will be used. For example, DLC-SLGID refers to DLC without using synthetic loops during training and using the LGID output to search loop candidates.

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 *TxVySz* 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 $\mathcal{C}_t$ has been computed using the four variations of our approach shown in Table 2, the corresponding four variations of DLC and GID as described before.

As an example, Figure 5 shows some of the candidate loops in each dataset according to ULF. The first column shows a query image of each dataset whilst the remaining columns

Table 3. AUC values of our proposal and DLC for all the tested configurations. The gray cells emphasize the best method for each of the four tested variations.

| | T1V2S3 | T1V3S2 | T2V1S3 | T2V3S1 | T3V1S2 | T3V2S1 | Average |
|---|---|---|---|---|---|---|---|
| **SLGID** | 88.75% | 83.60% | 88.17% | 73.58% | 88.24% | 75.96% | 83.05% |
| **DLC-SLGID** | 88.08% | 80.84% | 86.54% | 78.71% | 84.60% | 69.79% | 81.43% |
| **SLF** | 88.08% | 86.16% | 91.83% | 73.79% | 88.44% | 75.88% | 84.03% |
| **DLC-SLF** | 90.46% | 87.56% | 87.88% | 77.21% | 87.56% | 69.92% | 83.43% |
| **ULGID** | 89.50% | 82.76% | 92.50% | 73.50% | 86.48% | 76.38% | 83.52% |
| **DLC-ULGID** | 87.12% | 82.16% | 85.62% | 69.00% | 81.20% | 70.75% | 79.31% |
| **ULF** | 88.71% | 90.32% | 91.38% | 70.62% | 88.52% | 72.75% | 83.72% |
| **DLC-ULF** | 90.96% | 89.52% | 89.04% | 72.71% | 87.68% | 68.46% | 83.06% |

Table 4. AUC values of all the tested methods aggregated per dataset. The gray cells emphasize the best method between supervised and unsupervised for each of the four tested variations.

| | Dataset 1 | Dataset 2 | Dataset 3 | Average |
|---|---|---|---|---|
| SLGID | 75.77% | 85.92% | 88.46% | 83.38% |
| ULGID | 74.94% | 84.62% | 91.00% | 83.52% |
| DLC-SLGID | 74.25% | 82.72% | 87.31% | 81.42% |
| DLC-ULGID | 69.88% | 81.68% | 86.37% | 79.31% |
| SLF | 74.84% | 87.30% | 89.96% | 84.03% |
| ULF | 71.69% | 89.42% | 90.05% | 83.72% |
| DLC-SLF | 73.57% | 87.56% | 89.17% | 83.43% |
| DLC-ULF | 70.59% | 88.60% | 90.00% | 83.06% |
| GID | 61.17% | 39.00% | 45.75% | 48.64% |

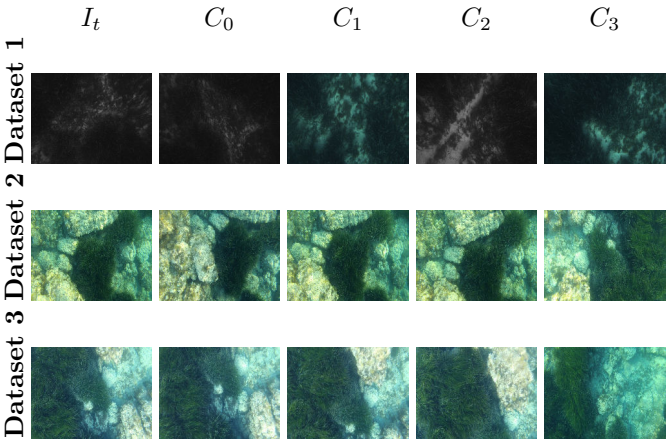


Fig. 5. Examples of candidate loops in each dataset.

depict the first four candidates found by our approach. As it can be observed, actual loops are within the candidate sets in all cases.

To quantify the quality of the loop candidates, we proceeded as follows. The number $N$ of items in $\mathcal{C}_t$ 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 6 shows some of the obtained hit ratios as a function of the percentage of database images. The labels in the examples corresponding to our proposal and to DLC specify the training, validation and test sets using the aforementioned TxVySz notation. The examples corresponding to GID do not use that notation since GID is neither trained nor validated, and thus only the tested dataset is specified as $Sx$, $x$ being the dataset number. Results using GID are significantly worse than those using the deep learning

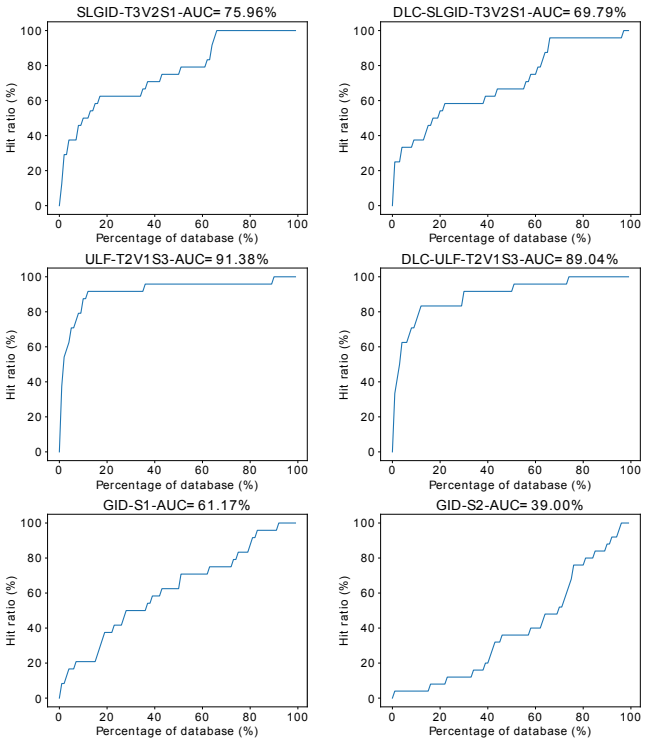


Fig. 6. Examples of hit ratio evolution.

approaches. This suggests that even though GID is not well suited to find loops, it is to provide useful information to train a NN to achieve this goal.

The evolution of the hit ratio with respect to $N$ (the number of candidates expressed as a percentage of database images) defines a curve so that the better the approach the larger the area below the curve. Let this *Area Under the Curve* (AUC), defined as the percentage of the whole space of possibilities that falls below the curve, be used as a quality measure of the overall behaviour of a loop detection method.

Table 3 shows the AUC corresponding to the four variations of our proposal and DLC for each valid combination of training, validation and testing. Our proposal leads to better average results either using LF or LGID and independently of using the supervised or the unsupervised approach. It can also be observed that both in our proposal and in DLC better results are obtained when using LF instead of LGID. This is particularly interesting since, LF being the encoder output, computing it is faster than computing the LGID and less parameters have to be stored.

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 4 summarizes these results and show that all the tested methods greatly improve the direct use of GID. This suggests that the NN is able to learn good representations of loops even with the weak information provided by the global image descriptor. The results in the table are grouped in order to ease the comparison between supervised and unsupervised approaches. Overall, the supervised approach leads to better results, though they are quite similar to their unsupervised counterparts. In other words, our proposal to unsupervised training is almost as accurate as training the system with a hand labelled ground truth.

## 8. CONCLUSION

A novel deep neural network aimed at robust and fast visual loop detection has been presented. The proposal is based on an autoencoder architecture, the decoder part being replaced by three fully connected layers. In order to help the proposed network to learn the features that define loop closings, a global image descriptor based on clusters of SIFT descriptors has been defined and used. Also, a method allowing unsupervised training has been presented.

Once trained, the NN builds two descriptors of an image that can be easily compared to descriptors of other images in order to ascertain if the images close a loop: the LGID and the LF, which are the outputs of the decoder and the encoder parts of the NN respectively. The former is the target output during training whilst the latter represents the learned latent representation.

The ability of the NN to detect loops has been experimentally tested and compared to previously existing methods. The results have shown that our proposal surpasses the previously existing methods and that the NN greatly improves the ability of the global image descriptor alone. Also, results show that the unsupervised approach leads to results similar to a classical supervised training, thus making it an interesting method in underwater scenarios in which hand-labelling loops is a tedious and error prone task.

Our proposal being lightweight and unsupervised, it is an interesting replacement for larger and slower NN, especially when it comes to AUV where computational capabilities are limited.

We are now working on a loop confirmation method able to robustly filter the candidate set so that loops can be properly used to perform underwater visual SLAM.

## 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. doi:10.1109/TPAMI.2017.2711011.

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*, 23–47. doi: 10.1002/9783527675418.ch2.

Burguera, A., Bonin-Font, F., and Oliver, G. (2015). Trajectory-based visual localization in underwater surveying missions. *Sensors (Switzerland)*, 15(1), 1708–1735. doi:10.3390/s150101708.

Davison, A., Calway, A., and Mayol, W. (2007). Visual SLAM. *IEEE Transactions on Robotics*, 24(5), 1088–1093. doi:10.1109/TRO.2008.2004521.

Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping (SLAM): part I The Essential Algorithms. *Robotics & Automation Magazine*, 2, 99–110. doi:10.1109/MRA.2006.1638022.

Gálvez-López, D. and Tardós, J.D. (2012). Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5), 1188–1197. doi: 10.1109/TRO.2012.2197158.

Hong, S., Kim, J., Pyo, J., , and Yu, S. (2016). A robust loop-closure method for visual slam in unstructured seafloor environments. *Autonomous Robots*, 6(40), 1095–1109.

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*, 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*, 3304–3311. doi:10.1109/CVPR.2010.5540039.

Jimenez-Rezende, D., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In E.P. Xing and T. Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, 1278–1286. PMLR, Bejing, China.

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. doi:10.1109/TRO.2017.2705103.

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. doi: 10.1007/s10514-015-9522-4.

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*. doi: 10.1109/CVPR.2007.383266.

Razavian, A.S., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). CNN features off-the-shelf: An astounding baseline for recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 512–519. doi:10.1109/CVPRW.2014.131.

Sünderhauf, N., Shirazi, S., Dayoub, F., Upcroft, B., and Milford, M. (2015). On the performance of ConvNet features for place recognition. In *IEEE International Conference on Intelligent Robots and Systems*, volume 2015-Decem, 4297–4304. doi:10.1109/IROS.2015.7353986.