Improving Sonar Image Patch Matching via Deep Learning

Matias Valdenegro-Toro[1]

Abstract— **Matching sonar images with high accuracy has been a problem for a long time, as sonar images are inherently hard to model due to reflections, noise and viewpoint dependence. Autonomous Underwater Vehicles require good sonar image matching capabilities for tasks such as tracking, simultaneous localization and mapping (SLAM) and some cases of object detection/recognition. We propose the use of Convolutional Neural Networks (CNN) to learn a matching function that can be trained from labeled sonar data, after pre-processing to generate matching and non-matching pairs. In a dataset of 39K training pairs, we obtain 0.91 Area under the ROC Curve (AUC) for a CNN that outputs a binary classification matching decision, and 0.89 AUC for another CNN that outputs a matching score. In comparison, classical keypoint matching methods like SIFT, SURF, ORB and AKAZE obtain AUC 0.61 to 0.68. Alternative learning methods obtain similar results, with a Random Forest Classifier obtaining AUC 0.79, and a Support Vector Machine resulting in AUC 0.66.**

I. Introduction

One of the basic problems in robotics is data association, where sensor readings have to be associated with previous measurements, as the combination of sensor data reduces noise and improves robot understanding of the world. Autonomous Underwater Vehicles (AUVs) constantly struggle with data association, as the underwater environment is very hostile for sensing. Some common robot tasks that require data association are tracking and simultaneous localization and mapping (SLAM). Object detection/recognition can also benefit from data association in the form of matching images if the task is to locate an object and only a single training sample is available.

Acoustic sensing (Sonar) is used in underwater environments as sound can travel large distances on water with little attenuation. Optical cameras are not an option as light is attenuated and absorbed by particles in the water column. Interpretation of acoustic images is not trivial as unwanted reflections, noise, and low signal-to-noise ratio (SNR) degrades the amount of information that the AUV can gather.

For sonar, matching image patches to known objects or landmarks in the environment is an important problem. Matching can also be formulated for other tasks such as mosaicing [1], where sonar images must be registered before being combined to improve SNR ratio and image resolution. Matching sonar images is difficult due to viewpoint dependence.

In this work, we propose the use of Convolutional Neural Networks (CNN) to learn a matcher for sonar images. Our objective is to produce a function that takes two sonar image patches and makes a binary decision: both images correspond to different views of the same object, or not. Matching should be possible even as insonification[1] varies due to AUV or sensor movement, different views, and object rotation or translation.

CNNs have obtained very good results [2] in different tasks that use optical color images, such as object recognition [3] and transfer learning [4]. We have previously evaluated CNNs for object recognition in sonar images and found that they also improve the state of the art [5]. CNNs have also been used to match patches from color images [6] with high accuracy. These results motivate us to use CNNs for sonar data, as the trained network can learn sonar-specific information directly from the data.

We show that we can build and train a CNN that matches Forward-Looking Sonar (FLS) image patches with high accuracy (AUC[2] 0.91), surpassing the state of the art keypoint matchers such as SIFT and SURF (with AUC in the range 0.61 - 0.68).

Our contributions are: we propose an algorithm to generate matching pairs from labeled objects for training, we learn the matching function directly from labeled data, without any manual feature engineering, we show that it is possible to match sonar images with relatively high accuracy.

II. Related Work

Matching sonar images with high accuracy has been a unsolved problem for a long time [7] [1] [8]. This is due to specifics issues in sonar imaging, such as viewpoint dependence, non-uniform insonification, low signal-to-noise ratio, low resolution, and low feature repeatability [9]. Most methods that are used for different kinds of matching in sonar imagery are not specifically designed for sonar (originally developed for optical images), and do not consider sonar-specific information.

Kim et al. [10] matches keypoints detected with the Harris corner detection to register general sonar images. Vandrish et al. [8] compares the use of SIFT [11] with different feature methods for sidescan sonar image registration, concluding that for this task SIFT performs best. Hurtos et al. [1] uses Fourier-based features for registration of Forward-Looking Sonar images, with great success. These kind of features could be used to make a matching decision, but they only work appropriately when rotation/translation between frames are small. Pham et al. [12] uses block-matching guided by

[1] Matias Valdenegro-Toro is with Ocean Systems Laboratory, School of Engineering & Physical Sciences, Heriot-Watt University, EH14 4AS, Edinburgh, UK m.valdenegro@hw.ac.uk

[1] Amount of acoustic signal that "illuminates" the target area by the sonar sensor.

[2] Area under the ROC Curve

a segmented sonar image with a Self-Organizing Map for registration and mosaicing of sidescan sonar images.

A large portion of the research about matching sonar images is devoted to registration and mosaicing [10] [1]. Both processes require many assumptions on the kind of images and their content, specially when considering non-uniform insonification and simple transformations between images.

In comparison, CNNs [3] have been used to compare and match color image patches. Zagoruyko et al. [6] uses CNNs trained on a dataset of 500K matched patches with high accuracy in tasks such as stereo matching and descriptor evaluation. Raw matching performance is also good, but only possible due to the availability of large labeled datasets.

Zbontar and LeCun [13] also use CNNs for stereo matching, improving over the state of the art in several datasets. These recent results using CNNs motivate us to explore such algorithm for matching sonar images. CNNs have several advantages when applied to sonar imaging: they can learn sonar-specific information directly from raw data, they do not require feature engineering or specific data preprocessing, and they make little assumptions on input data.

III. Matching Sonar Image Patches with CNNs

A. Training Data Generation

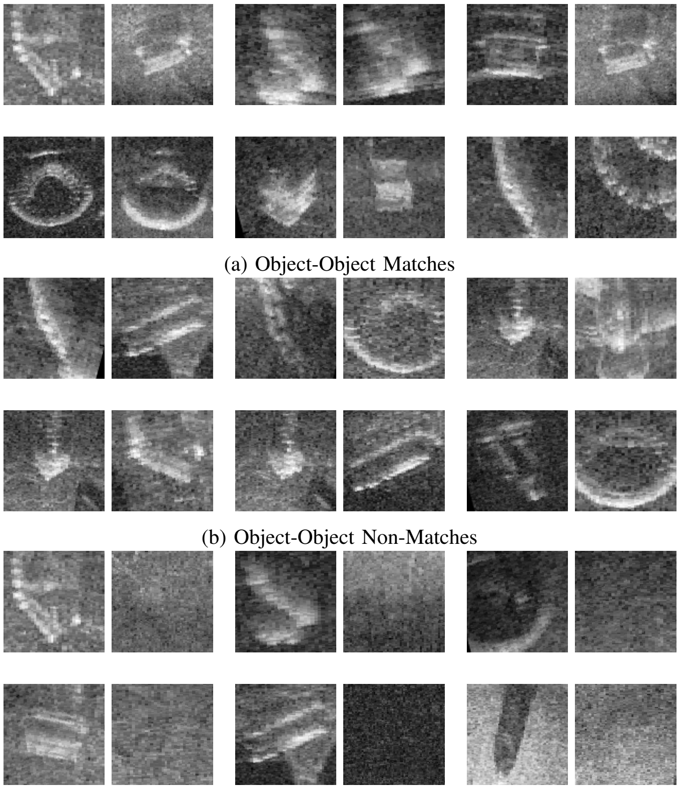
Given a dataset containing labeled bounding boxes (including object classes), we generate matching and non-matching image pairs that are sampled from the dataset. We do this by using object class information to generate matching image pairs, and we also produce non-matching pairs that contain objects versus background. The dataset that we used for this purpose was originally designed for object detection. We generate the following kinds of pairs:

- **Object vs Object, Same class**. A matching pair is generated from two objects of the same class. We sample two random image crops of objects in the same class and generate one pair, typically both crops corresponds to different perspectives of the same object, or different insonification levels from the sensor.
- **Object vs Object, Different class**. A non-matching pair is generated from two objects from different classes. This makes the assumption that objects in the dataset are not similar across different classes.
- **Object vs Background**. A non-matching pair is generated by sampling one background patch that has IoU score lower than 0.1 with the ground truth and generating a pair with a random object image crop.

As the number of possible non-matching pairs is very large, we balance matches (positive) and non-matches (negative) samples to be 1 : 1. This is done by sampling 10 matches per object, 5 non-matches between objects of different class, and 5 non-matches with background. The detailed algorithm is presented in Algorithm 1. We generate pairs of 96×96 image crops, as this is the most appropriate size for the objects in our dataset. A small sample of generated pairs is shown in Fig. 1.

Algorithm 1. Training Data Generation

```
Input: Labeled Image dataset $I$ with bounding boxes $B_i$ and
    class labels $C_i$, number of positive samples $S_p$, number
    of negative samples $S_n$.
Output: List of matching pairs $L_m$ and list of non-matching
    pairs $L_{nm}$.
$L_m \leftarrow \emptyset, L_{nm} \leftarrow \emptyset$
for img $\in I$ do
    for object $o \in$ img do
        $OC \leftarrow$ crop $B_o$ from img.
        for $i = 0$ to $S_p$ do
            $MC \leftarrow$ sample random object $p$ of class $C_o$ and
            make an image crop.
            Append $(OC, MC)$ to $L_m$
        end for
        for $i = 0$ to $S_n$ do
            $NMC \leftarrow$ sample random object $p$ of class $C_p \neq$
            $C_o$, and make an image crop.
            Append $(OC, NMC)$ to $L_{nm}$
        end for
        for $i = 0$ to $S_n$ do
            $BC \leftarrow$ sample random background patch and
            make an image crop.
            Append $(OC, BC)$ to $L_{nm}$
        end for
    end for
end for
```



(a) Object-Object Matches

(b) Object-Object Non-Matches

(c) Object-Background Non-Matches

Fig. 1. A small sample sonar image patches labeled as matching or non-matching that were generated by our algorithm. These patches were captured with an ARIS Explorer 3000 Forward-Looking Sonar.

B. CNN Architecture

We base our architectural choices on the work of Zagoruyko et al. [6]. This paper introduced CNNs for matching image patches and propose three different architectures for that task: A siamese, pseudo-siamese and a two-channel architecture. We use the two-channel and siamese architectures. In the two-channel architecture, both input images (denoted as I_A and I_B) are merged to form a two-channel image, which is the input to our neural network.

The siamese architecture uses two branches that share weights, I_A is input to the left branch, while I_B is input to the right branch. The output of each branch is a feature vector of a fixed size. Both feature vectors from each branch are concatenated to form a single vector that is input to a decision network (formed only of fully connected layers). The idea of a siamese network is that both branches share weights and will learn patch invariant features that are useful for the decision network to produce a matching decision. We use two 96×96 input images to be matched.

We use the following notation for CNN layers: Conv(N_f, $F_w \times F_h$) is a convolutional layers with N_f filters of width F_w and height F_h. MP(P_w, P_h) is a max-pooling layer with sub-sampling size of $P_w \times P_h$, and FC(n) is a fully connected layers with n output neurons.

We designed four CNN architectures, two using a 2-Channel approach, and two using a Siamese architecture. We obtained these architectures by performing grid search over a defined set of variations, including depth, number of filters, and filter size. We now describe the 2-Channel architectures:

- **2-Channel CNN Class**. This network is designed to output a binary matching decision (match or non-match), with a two-element output probability distribution given by a softmax function. The full network architecture is Conv(16, 5×5)-MP(2, 2)-Conv(32, 5×5)-MP(2, 2)-Conv(32, 5×5)-MP(2, 2)-Conv(16, 5×5)-MP(2, 2)-FC(64)-FC(32)-FC(2). The network is trained using a categorical cross-entropy loss function, as the matching decision is formulated as a classification problem.
- **2-Channel CNN Score**. This network outputs a matching score in the range $[0, 1]$ with a sigmoid function. The full network architecture is Conv(16, 5×5)-MP(2, 2)-Conv(32, 5×5)-MP(2, 2)-Conv(32, 5×5)-MP(2, 2)-Conv(16, 5×5)-MP(2, 2)-FC(64)-FC(32)-FC(1). This network is trained using binary cross-entropy loss function, and the activation at the output is sigmoid.

The Siamese architectures are based on branches with configurations Conv(16, 5×5)-MP(2, 2)-Conv(32, 5×5)-MP(2, 2)-Conv(64, 5×5)-MP(2, 2)-Conv(32, 5×5)-MP(2, 2)-FC(96)-FC(96). The output feature vector contains 96 elements, and the output activation is sigmoid. From this branch architecture we derive the following architectures:

- **Siamese CNN Class**. Both branch outputs are concatenated to form a 192 element vector, that is passed through a decision network with configuration FC(64)-FC(2). The output activation in this case is softmax. This network is trained with a categorical cross-entropy loss.
- **Siamese CNN Score**. Same as the previous architecture, but the decision network has configuration FC(64)-FC(1), with a sigmoid output activation. This network is trained with a binary cross-entropy loss.

All four architectures were obtained by doing grid search over a varying number of layers, convolutional filters and fully connected neurons. The categorical and binary cross-entropy loss functions are the same for the case of binary classification, with the only difference being the application to a single output or to a two-element vector (see Eq 1).

$$L(y, \hat{y}) = -\sum_i y_i \log(\hat{y}_i) = -y_0 \log(\hat{y}_0) - (1 - y_0) \log(1 - \hat{y}_0) \tag{1}$$

All architectures use ReLU activations, except at the output layers, and are trained on the same dataset, and learn to discriminate sonar image patches. Dropout [14] is used after every fully connected layer (except at outputs). We also evaluated the use of Batch Normalization [15] but Dropout is superior in achieving good generalization performance. Our original design was the class output networks, posed as a classification problem, which works well but interpretation of the output is not trivial as the scores are correlated with the classification outputs. This motivated us to explore the scoring architecture that outputs a score directly that can be separated to obtain a matching decision by a simple threshold . Note that [6] uses networks that only output a score, while we both evaluate continuous score and discrete classification outputs.

C. Training

Our networks are trained using stochastic gradient descent from random initialized weights. We train using mini-batch gradient descent using a batch size of $b = 128$ images. We adopt the ADAM optimizer [16] for accelerated training and learning rate decay. The initial learning rate is $\alpha = 0.1$. All networks are trained for 5 epochs. We tuned this value on a validation set with early stopping.

In order to prevent the network from learning patterns in the order images are presented, we augment the dataset and for each training pair (A, B), we add the pair (B, A) to the augmented training set. No other data augmentation was used.

IV. Experimental Evaluation

A. Data

We have captured a dataset of marine debris objects in our water tank with an ARIS Explorer 3000 (FLS). This dataset consists of 2072 images with ~ 2500 total object instances labeled in 9 classes (Metal Cans, Bottles, Drink Carton, Metal Chain, Propeller, Tire, Hook, Valve, Background). On this dataset we ran our matching pair generation algorithm (Algorithm 1). In order to evaluate generalization performance of our networks, we split the dataset according to object class, before obtaining train and test splits. This generated two datasets:

- **Dataset D**: In this dataset the train and testing sets are generated with different objects. Classes 0-5 are used to generate the training set, while classes 6-9 are used to generate the test set. All of our matching networks are trained on this training set.
- **Dataset S**: This dataset is generated using all classes, and split into a training and testing set. From this split we only use the testing set to evaluate performance.

The training set (from Dataset D) contains 39840 matching and non-matching pairs (50% each). Both testing sets (D and S) contain 7440 matching and non-matching pairs, also balanced. All reported metrics are evaluated on the test set.

B. Matching Performance

In this section we evaluate raw matching performance. We plot the ROC curve, and report the Area Under the Curve (AUC). We also obtained accuracy scores for the test set. Accuracy for the matching networks was obtained by considering the raw match probability p (second element of the softmax output, or the sigmoid output score) and taking the class with maximum probability (Eq 2).

$$c = \arg\max\{1 - p, p\} \tag{2}$$

We compare both our matching networks with the state of the art keypoint detectors and feature extractors, namely SIFT [11], SURF [17], ORB [18] and AKAZE [19]. SIFT and SURF represent the best keypoint detectors for optical images, while ORB was chosen to evaluate its binary features. While it is known [1] that these algorithms do not perform well in sonar, there is no other comparison point, as no keypoint detectors have been developed specifically for sonar images. We also compare with Machine Learning (ML) based methods, namely a Support Vector Machine (SVM) as classifier, a Support Vector Regressor (SVR) to regress a score, and Random Forest (RF) classifier and regressor.

Accuracy for keypoint algorithms is obtained by considering a positive match when the ratio test [11] gives at least 1 good match. If there are no good matches, then we output a negative match. This threshold is low on purpose to evaluate the best performance of a keypoint matching system.

As our dataset is generated using one type of matching pairs and two different types of non-matching pairs, we also compute the accuracy over each kind of match. This is reported as ”Obj-Obj $+$” for Object-Object matching pairs, ”Obj-Obj $-$” for Object-Object non-matching pairs, and ”Obj-Bg $-$” for Object-Background non-matching pairs.

Table I displays the main results, only considering the best performing matching networks. Our methods have considerably higher AUC and mean accuracy, which shows that using neural networks for matching sonar images does have a considerable improvement over the state of the art. The 2-Chan CNN Class network has an advantage of 22.4 AUC percentage points over ORB, with a corresponding increase of 31.3 accuracy percentage points.

Classical keypoint detectors match sonar images with a chance that is slightly better than chance, with the best classical method being ORB with 0.682 AUC. Both our matching CNNs outperform classic matches by a considerable margin. Our class matcher also outperforms the scoring matcher. This is due to the fact that scoring matcher is considerably harder to train because the sigmoid output easily saturates. This also contrasts with the results from [6] as they use $\{+1, -1\}$ scoring with a hinge loss. Our results show that a softmax output with cross-entropy loss can outperform a saturating non-linearity.

Machine Learning methods also perform poorly, but better than keypoint matching. A RF classifier has the highest non-CNN AUC at 0.795, but their Obj-Obj positive accuracy is considerably poor than the alternatives. SVM and SVE have AUC that is close to keypoint matching.

Classic matchers have a higher Obj-Obj positive accuracy, and this can be explained by overconfident predictions that classify too many pairs as positive matches. This can easily produce high accuracy for positive matches, but will hurt the performance of negative matches. Our matching networks seem to be more balanced, but still their lowest accuracy is when they need to predict a positive match. Both results show the difficulty of matching sonar images. ML methods suffer from the opposite, where they are very accurate for negative matches, but suffer in accuracy for positive matches.

Fig. 2 shows the corresponding ROC curves for the classic matchers, ML-based methods, and our best performing networks. The positive class probability p of Class matching networks and SVM/RF-Class is used to construct the ROC curve, while for Score matching networks and SVR/RF-score the raw score is considered to produce the curve. For keypoint detectors we vary the minimum number of good keypoint matches to declare a positive match. Keypoint matchers have a curve that is very close to random chance, while our methods are closer to a perfect matcher. There is still a considerable room for improvement in the sonar image matching problem. All tested methods produce results that are better than random chance, and RF-based methods are superior when compared to keypoint matching, but still our matching networks are considerably better.

Tables III and II show a comparison of all our matching networks on the **S** and **D** datasets. Corresponding ROC curves are shown in Fig. 3a and Fig. 3b.

Looking at Fig. 3a, we can see that network 2-Chan CNN Class performs the best when compared to the scoring network, but this trend reverses when looking at Siamese networks (Fig. 3b), as the highest AUC is obtained by Siamese CNN Score. When comparing performance on **S** and **D** datasets, we show that performance slightly increases when evaluating on the same objects as the training set (Test set **S**). This is expected and shows a slight amount of overfit to the training set objects. But generalization performance to unseen objects (Test set **D**) is still good.

Finally, Table IV offers a breakdown of accuracy in our matching networks over the three different match pairs on both datasets **S** and **D**. When evaluated in different objects, all networks have decreased performance on Object to Object positive matches, while at the same time having adequate

Method	AUC	Mean Accuracy	Obj-Obj + Acc	Obj-Obj − Acc	Obj-Bg − Acc
SIFT	0.610	54.0%	74.5%	43.6%	44.0%
SURF	0.679	48.1%	89.9%	18.6%	35.9%
ORB	0.682	54.9%	72.3%	41.9%	60.5%
AKAZE	0.634	52.2%	95.1%	4.8%	56.8%
RF-Score	0.741	57.6%	22.5%	88.2%	97.2%
RF-Class	0.795	69.9%	12.5%	97.7%	99.7%
SVR-Score	0.663	70.5%	57.2%	66.6%	87.5%
SVM-Class	0.652	67.1%	54.4%	69.1%	90.5%
2-Chan CNN Class	0.910	86.2%	67.3%	95.2%	96.1%
2-Chan CNN Score	0.894	82.9%	68.0%	96.1%	84.5%

TABLE I. Comparison of classic keypoint algorithms for matching versus our two best performing matching networks. Area Under the ROC Curve (AUC), Accuracy at match threshold zero, and Accuracy for each match type is reported for Test Set **D**.

Network Type	Output	Test Objects	AUC	Mean Accuracy
2-Chan CNN	2 Class	Different	0.910	86.2%
2-Chan CNN	Score	Different	0.894	82.9%
Siamese CNN	2 Class	Different	0.855	82.9%
Siamese CNN	Score	Different	0.826	77.0%

TABLE II. Accuracy and Area Under the ROC Curve (AUC) metrics for Test Set **D**

Network Type	Output	Test Objects	AUC	Mean Accuracy
2-Chan CNN	2 Class	Same	0.944	86.7%
2-Chan CNN	Score	Same	0.934	85.4%
Siamese CNN	2 Class	Same	0.864	75.8%
Siamese CNN	Score	Same	0.895	80.6%

TABLE III. Accuracy and Area Under the ROC Curve (AUC) metrics for Test Set **S**

performance in both negative cases. For evaluation in the same objects as the training set, this trend reverses (as expected) and the largest accuracies are reported in the Object to Object positive matches. This indicates that the networks are slightly overfitting the objects in the training set, but still they do provide an improvement over keypoint matching and ML-based methods in unseen objects. Matching an object to a different view of the same object is also a hard problem, and discarding matches from different objects or to background is easier.

C. *Discussion*

Our 2-Channel CNN Class matching networks performs well, with an AUC of 0.91 when evaluated on unseen data, and AUC of 0.94 on a dataset that shares objects with the train set. We have not seen any previous work claiming to match sonars images with such performance. Classic methods (SIFT/SURF and ORB) are known to work poorly in sonar, but still they are being used for registration in a large part of the literature [10] [7] [8]. We have not seen quantitative results of keypoint matching in sonar images, and our extensive evaluation is useful for comparison.

One clear limitation of our evaluation is the small size of our datasets (39K for training and 7K for testing). We believe our networks could greatly benefit from a bigger dataset, containing more variability in objects as well as more object views. A dataset captured in real underwater conditions is ideal but hard to produce.

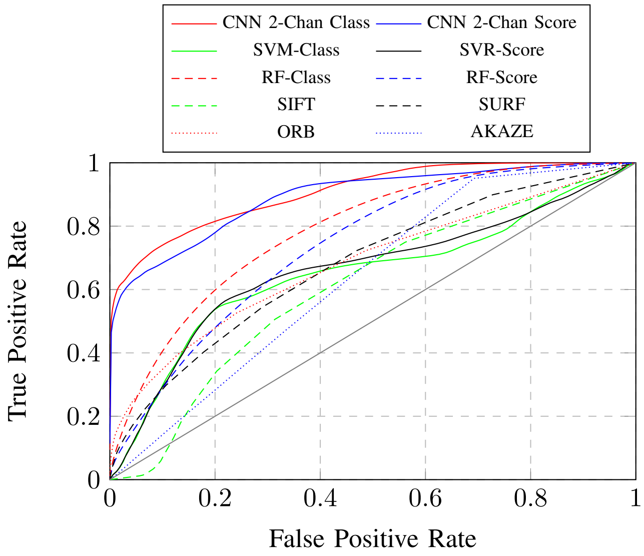


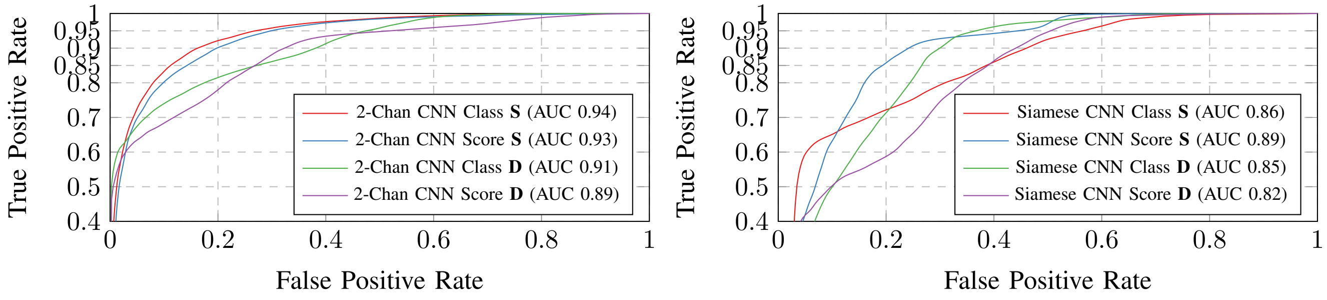
Fig. 2. ROC curve comparing our matching networks with ML-based methods (SVM/SVR and Random Forest, RF) and keypoint matching methods (SIFT, SURF, ORB and AKAZE). Our methods clearly outperform other methods when used to match sonar images. These curves were evaluated on the **D** dataset. The grey line represents random chance limit.

V. Conclusions

We have proposed the use of CNNs to learn a matching function for sonar images. Our results show that our 2-Chan CNN Class network can match FLS image patches with high accuracy (AUC 0.91), while classic keypoint matching methods can do it only with low accuracy, slightly better than random chance (AUC 0.61 to 0.68). ML-based methods (SVM and Random Forests) are also inferior (AUC 0.65 to 0.80).

We believe that our results are appropriate for the small (39K samples) training dataset that we possess, and they could improve if more data and object/background variability is available. Our dataset was captured under laboratory conditions in a small water tank containing household garbage objects, and our work can surely be improved with a dataset captured in real underwater scenes.

Our method is limited by available number of images and objects in them. Our network architecture could fail to generalize with other objects, specially if their shape is radically different. Background is also a concern, as the



(a) 2-Channel CNN

(b) Siamese CNN

Fig. 3. ROC curves comparing 2 Channel and Siamese matching networks on test sets **S** and **D**. This comparison shows the difference in performance Test set **S** was generated with the same objects as in the training set. Test set **D** was generated with different objects than in the training set.

		Different Objects			Same Objects		
Network Type	Output	Obj-Obj + Acc	Obj-Obj − Acc	Obj-Bg − Acc	Obj-Obj + Acc	Obj-Obj − Acc	Obj-Bg − Acc
2-Chan CNN	2 Class	67.3%	95.2%	96.1%	86.6%	75.7%	97.8%
2-Chan CNN	Score	68.0%	96.1%	84.5%	85.0%	77.5%	93.7%
Siamese CNN	2 Class	62.9%	89.9%	96.0%	92.2%	39.4%	95.8%
Siamese CNN	Score	49.2%	84.7%	97.0%	89.1%	55.3%	97.3%

TABLE IV. Accuracy by matching type, for the **S** and **D** datasets

background in our water tank is not representative of a typical underwater environment.

We expect that AUV perception will benefit from our work and open possibilities of advanced CNN-based matching and registration methods. Our work can be applied to any kind of sonar, as we did not make assumptions on the kind of image produced by the sonar.

As future work, we would like to build a similarity function for sonar images instead of making a binary decision and learn from image patches in a unsupervised way. The best case is to learn from a dataset that contains automatically matched scenes with another sensor (like depth in optical images) as [13] and [6] do.

ACKNOWLEDGMENTS

This work has been partially supported by the FP7-PEOPLE-2013-ITN project ROBOCADEMY (Ref 608096) funded by the European Commission. The author would like to thank Leonard McLean for his help in capturing data used in this paper.

REFERENCES

[1] N. Hurtós, Y. Petillot, J. Salvi *et al.*, "Fourier-based registrations for two-dimensional forward-looking sonar image mosaicing," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5298–5305.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[4] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 806–813.

[5] M. Valdenegro-Toro, "Object recognition in forward-looking sonar images with convolutional neural networks," in *OCEANS 2016 MTS/IEEE Monterey*. IEEE, 2016, pp. 1–6.

[6] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4353–4361.

[7] S. Negahdaripour, M. Aykin, and S. Sinnarajah, "Dynamic scene analysis and mosaicing of benthic habitats by fs sonar imaging-issues and complexities," in *OCEANS'11 MTS/IEEE KONA*. IEEE, 2011, pp. 1–7.

[8] P. Vandrish, A. Vardy, D. Walker, and O. Dobre, "Side-scan sonar image registration for auv navigation," in *Underwater Technology (UT), 2011 IEEE Symposium on and 2011 Workshop on Scientific Use of Submarine Cables and Related Technologies (SSC)*. IEEE, 2011, pp. 1–7.

[9] N. Hurtós, N. Palomeras, S. Nagappa, and J. Salvi, "Automatic detection of underwater chain links using a forward-looking sonar," in *OCEANS-Bergen, 2013 MTS/IEEE*. IEEE, 2013, pp. 1–7.

[10] K. Kim, N. Neretti, and N. Intrator, "Mosaicing of acoustic camera images," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 152, no. 4, pp. 263–270, 2005.

[11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[12] M. T. Pham and D. Guériot, "Guided block-matching for sonar image registration using unsupervised kohonen neural networks," in *2013 OCEANS-San Diego*. IEEE, 2013, pp. 1–5.

[13] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *Journal of Machine Learning Research*, vol. 17, pp. 1–32, 2016.

[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[16] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[17] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[18] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. IEEE, 2011, pp. 2564–2571.

[19] P. F. Alcantarilla and T. Solutions, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *IEEE Trans. Patt. Anal. Mach. Intell*, vol. 34, no. 7, pp. 1281–1298, 2011.