Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 52-21 (2019) 156–162

Autonomous Underwater Vehicle Navigation Using Sonar Image Matching based on Convolutional Neural Network

Wenli Yang*, Shuangshuang Fan, Shuxiang Xu*, Peter King***, Byeong Kang*, Eonjoo Kim*****

** Information & Communication Technology, University of Tasmania, TAS, Australia (e-mail: yang.wenli@ utas.edu.au, shuxiang.xu@utas.edu.au,byeong.kang@utas.edu.au).*
*** School of Marine Sciences, Sun Yat-sen University, Zhuhai, Guangdong, 519082, China (email: fanshsh6@mail.sysu.edu.cn)*
**** National Centre for Maritime Engineering and Hydrodynamics, Australian Maritime College, University of Tasmania, Launceston, TAS, 7250, Australia (e-mail: pdking@utas.edu.au, eonjoo.kim@utas.edu.au)*

Abstract:

This paper presents an image matching algorithm based on convolutional neural network (CNN) to aid in the navigating of an Autonomous Underwater Vehicle (AUV) where external navigation aids are not available. We aim to solve the problem where traditional image feature representations and similarity learning are not learned jointly and to improve the matching accuracy of sonar images in deep ocean with dynamic backgrounds, low-intensity and high-noise scenes. In our work, the proposed CNN-based model can train the texture features of sonar images without any manually designed feature descriptors, which can jointly optimize the representation of the input data conditioned on the similarity measure being used. The validation studies show the feasibility and veracity of the proposed method for many general and offset cases using collected sonar images.

Keywords: Sonar Image matching, Convolutional neural network, feature extraction, AUV, Teach-and-Repeat path following.

1. INTRODUCTION

Nowadays, autonomous underwater vehicles (AUVs) are able to dive into the deep ocean to gather data on scientific missions. Paull (2014) and Kinsey (2006) found that AUVs are required to navigate long distances in underwater/under-ice environments where absolute navigation aid (GPS) are not available (such as Antarctic and Southern Ocean) due to the severe attenuation of high frequency radio signals through underwater.

Geophysical referencing allows an AUV to position itself using sensory feedback. King (2018) introduced that Teach-and-Repeat (TR) path following was an efficient geophysical method which was used to re-follow a path only based on the historical collected data. This proposed approach is beneficial to both AUV homing operation and area coverage mission.

According to King (2018), for long-range exploration missions, TR allows an AUV to venture into an unexplored area and return along the same path, regardless of its accumulated global position error. Image recognition and matching are the key processes to achieve sonar image interpreting and TR path following. Thus, this paper aims to develop an effective image matching algorithm for the TR path following control of an AUV moving in deep ocean.

At present, there are three main approaches for image matching (Remondino et al. 2014): shape matching, direct (appearance-based) registration and feature-based registration. For sonar images it is difficult to get exact shape from the original image, and there is a wide diversity of appearances between images. Thus, the most common methods used for sonar image matching is feature-based registration, which represents the image the encoding of various features. Some researchers have used feature extraction and matching for sonar images including corners, colour schemes or texture of image, such as HOG (Aulinas et al. 2011), SIFT (Kavukcuoglu et al. 2009), SURF (Prabhakar and Kumar 2012), etc. These traditional features can be represented with complex cells in human brain. However, due to the diversity of appearances, illumination conditions and backgrounds, it is difficult to manually design a robust feature descriptor to perfectly describe various features (Zhao et al. 2019). Another shortcoming of traditional image matching method is that the feature representation of the data and the similarity learning are not learned jointly. Furthermore, AUV is different to repeat exact following path; in many cases there are certain offsets between the path to go and return. Traditional methods always have lower matching accuracy for these offsets cases.

In this paper, a deep neural network is considered to train directly from sonar image data without any manually-designed features. In this way the resulting descriptors will be very efficient to compute even when dense. Moreover, due to its convolutional nature, deep neural networks can jointly optimize the representation of the input data conditioned on

10.1016/j.ifacol.2019.12.300

the similarity measure being used, also known as end-to-end learning, to achieve accurate TR path following control on the AUV. The main contributions of this work are: a deep neural network structure to extract region-based features and achieve feature matching between different sonar images; the proposed deep model is validated using the collected sonar images for many offset cases to show the feasibility and veracity of this approach.

2. RELATED WORK

To help AUV navigate in the deep ocean, researchers have widely used image recognition techniques including feature extraction and matching in their underwater missions. Traditional feature extraction methods just consider limited feature types which require expert knowledge to select the interesting features, which are generally not suited to sonar images. In addition, Russ (2016) have proposed that the influence of imaging conditions such as illumination is also one of the major difficulties in the context. Deep learning methods are an alternative which can overcome the above traditional drawbacks. Convolutional Neural Networks, as a specific deep learning method, mitigate the curse of high dimensionality inherent in fully connected networks but must be trained, unlike other feature extractors (Zhu et al. 2017, Cao et al. 2016 and Horn et al. 2017). This allows image features to be discovered and utilised. Once image features are identified, they can be used for image matching. This section we shortly review the related works and present some fundamental deep neural network concepts used for image matching.

From 2015, some researcher such as Zagoruyko et al. (2015) and Gatys et al. (2015) have proposed to compare and match images via Convolutional Neural Networks. Based on that idea, a multi-scale CNN structure was presented to improve capturing fine grained image similarities over traditional CNN's. Up to now, these are two of the main types of deep learning methods used in image matching.

The one stage method is the one that uses a single convolutional neural network as shown in Fig. 1 to extract object features directly without region proposal selection, such as YOLO (Redmon et al. 2016), SSD (Liu et al. 2016), etc. SSD and YOLO only need an input image and ground truth boxes for each object during training. However, the limitations of one stage algorithms is that they struggle with small objects within the image (Girshick et al. 2014), for example it might have difficulties in extracting and matching detailed features. This is due to the spatial constraints of the algorithms.

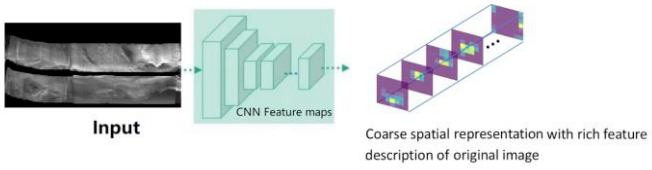


Fig. 1. One stage feature extraction processing

In the two stage method (see Fig. 2), the feature extraction is divided into two stages. The first stage is aimed at selecting region proposals to find a small set of Region of Interests (ROIs) which tightly cover as many useful features in the image as possible. The second stage is used to generated feature representations based on selected ROIs, such as R-CNN, Fast-RCNN (Girshick 2015), Faster-RCNN (Ren et al. 2015), R-FCN (Dai et al. 2016), etc.

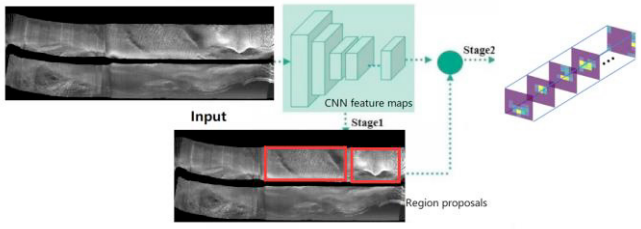


Fig. 2. Two stage feature extraction processing

In general, two stage methods have advantages in accuracy, while one stage methods are faster (Huang et al. 2017). In our research, the accuracy is considered to be more important than processing speed, thus two stage methods are used to achieve feature matching between different sonar images. Furthermore, to extract different features, we need to consider the different layered neural network structure. For our work we compare and summarize the layered structure of features in different target objects shown in Table 1.

Table 1. Layered structure of features in different target objects

Target	Original input	Shallow levels	Middle levels	High levels
Speech	Sample	Frequency, voice	Tone, phoneme	word
Image	Pixel	Lines, texture	Pattern, local	object
Text	letter	Word, phrase	Sentence, paragraph	article

According to Table 1, for the sonar image feature extraction and matching, we need to extract the texture feature from original image, thus the most useful features are extracted from shallow levels rather than high levels.

3. PROPOSED SOLUTION

In order to design and develop a sonar image matching model based on convolutional neural network, this section describes the overview architecture of our proposed method and detailed methodologies used in the design of the system's sub components.

The problem has been modelled as sonar image matching to calculate the similarities between two different images. The overall architecture of our proposed system is shown in Fig. 3.

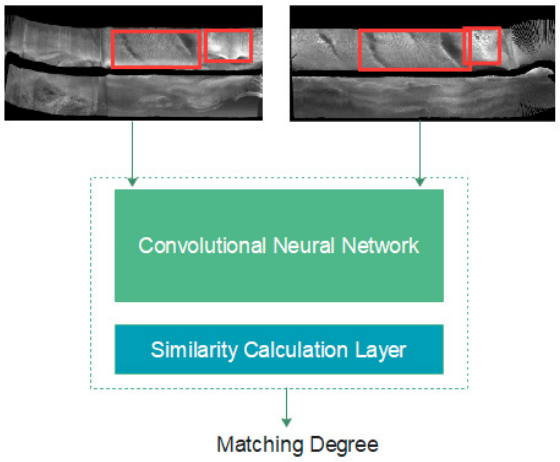


Fig. 3. The overall architecture of image matching using convolutional neural network.

For this a convolutional neural network architecture is used to extract and concatenate the feature descriptions of input ROIs and pass to a top network that consists of linear fully connected, batch normalization layer and ReLU layers to try to minimize a loss function. There are four main components shown in Fig. 4 to extract texture features from sonar image and build a deep model to compare different images and determine the matching degree they both cover the same location from one orientated image to other repeat orientated images.

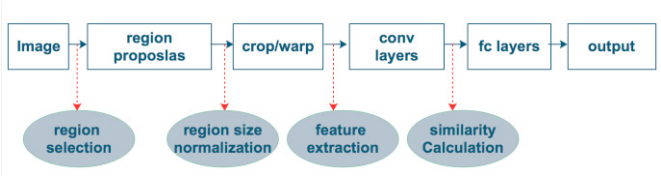


Fig. 4. The main workflow of sonar image matching using Convolutional Neural Network.

3.1 Region selection

For the original sonar images, there are many useless features from the global level, thus region selection method is used to select local-features to train the model. Among the region selection methods, Uijlings et al. (2013) proposed the selective searching method, which works by clustering image pixels into segments, and then performing hierarchical clustering to combine segments from the same feature regions into useful features proposals.

Since there are few training images, we utilized pre-processing methods to generate more training samples, including rotating images, flipping the image both horizontally and vertically and enhanced images. Thus, each selected ROI will be expanded to 10 local-featured images shown in Fig. 5.

The goal of this step is to find a small set of ROIs which tightly cover as many useful features in the image as possible.

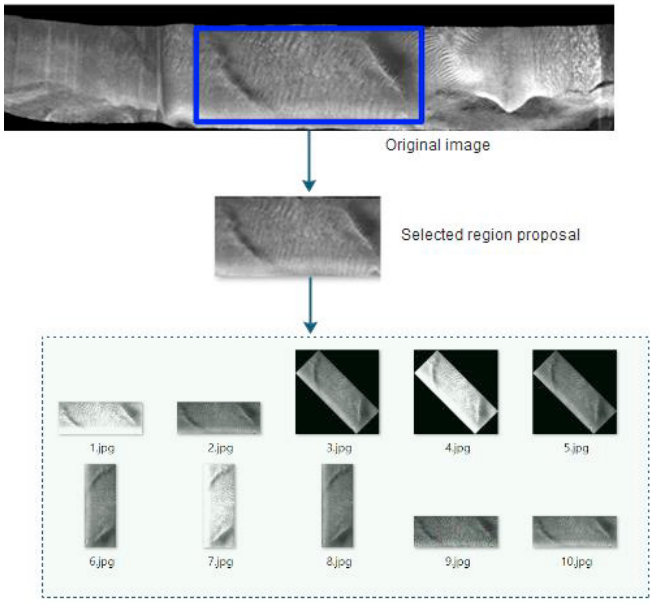


Fig. 5. 10 extending local-featured images (including brighter, darker, rotating, adding salt-and-pepper noise and gaussian noise.)

3.2 Feature extraction

The most common sizes of feature maps are 3x3, 5x5 and 7x7, where the 7x7 always gets the best features, in regards to classification research problems (Cheng et al. 2016). As we know, the convolutional network is used to reduce the resolution of images, with the resolution reduction factor related to the stride, which is equal to 2n. Our training datasets are 300 dpi which is close to 7x25=224 (compared to 7x24=112 and 7x26=448). Thus, during training, each warped region proposal is resized to the same size 224x224. These region proposals are then sent through the network structure which outputs a vector of e.g. 512 floating point values for each ROI.

Furthermore, the greedy layer-wise pre-training which was proposed by Bengio et al. (2007) tis used to guide how many layers for convolutional neural network layers. Firstly, we set the original network layers as two convolutional layers, and then each time we added one more convolutional layer to test the generated model. The above the operation was repeated until we get the following evaluation results shown in Table 2.

Table 2. The comparisons of the matching degree accuracy using different convolutional layers

Layers number	Test loss	Test accuracy

2 Conv layers+2 pooling layers	5.47	0.33
3 Conv layers+3 pooling layers	4.63	0.33
4 Conv layers+4 pooling layers	3.55	0.5
5 Conv layers+5 pooling layers	1.05	0.85
6 Conv layers+6 pooling layers	1.68	0.63

Based on the above evaluation results and explanation, we choose five convolutional layers to train the feature extraction model. The network structure of our proposed system is shown as Fig. 6.

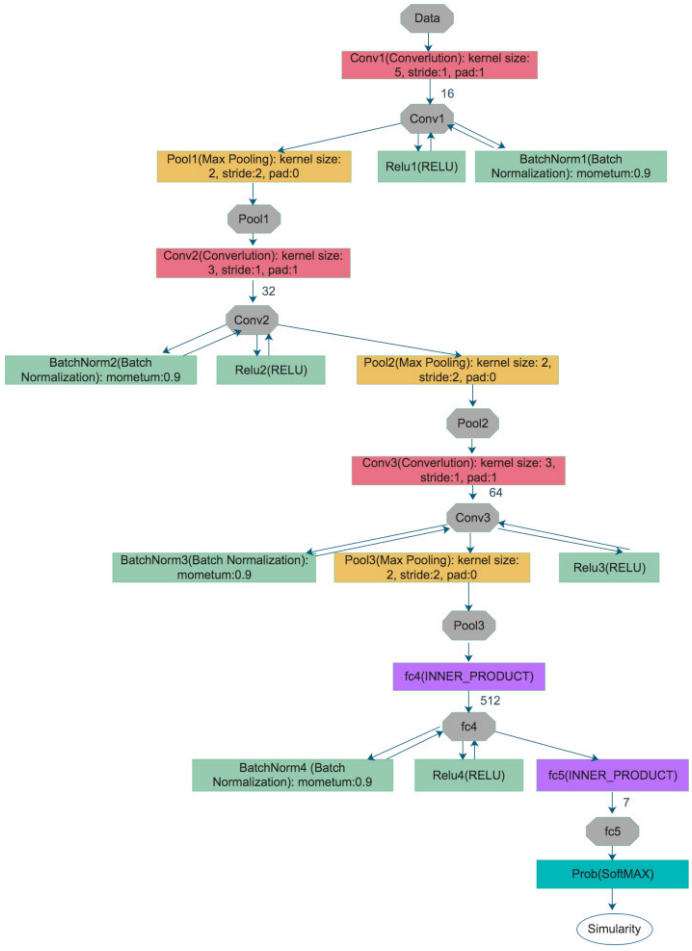


Fig. 6. The deep neural network structure for feature extraction

3.3 Similarity calculation

Based on the extracted feature map of the sonar images, we add two full-connected layers to calculate the similarity between feature maps, which takes the 512 float ROI representation as input, and we select the highest similarity as final matching results.

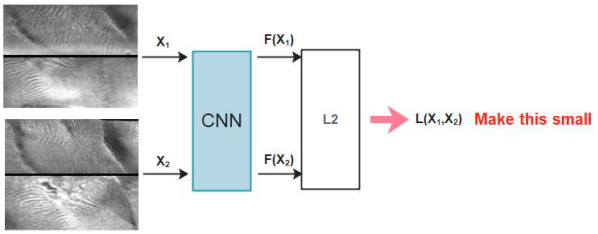


Fig. 7. The loss function used in similarity prediction

In this part, cross-entropy loss function as (1) is used to calculate the similarity between the prediction similarity and true similarity. Our training aim is to make loss function as small as possible.

$$(x_1, x_2) = \begin{cases} \|F(x_1) - F(x_2)\|_2, & p_1 = p_2 \\ \max(0, c - \|F(x_1) - F(x_2)\|_2), & p_1 \neq p_2 \end{cases} \quad (1)$$

Based on the above core steps, in order to increase the test accuracy as well as the training speed between different sonar images we implemented the following improvements:

- batch normalization to decrease over-fitting;
- early stop when the training accuracy increases (positive slope) while the validation accuracy steadily decreases (negative slope) to stop training ahead of time.

3.4 AUV Navigation

Based on the similarity results, we can get the node-to-node relative positioning to repeat the path. The Fig.8. shows the flowchart of how AUV is navigated.

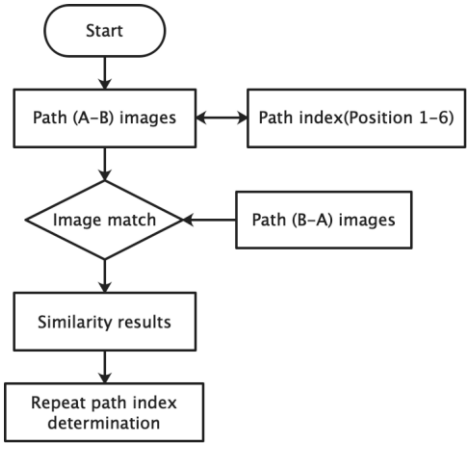


Fig. 8. Flowchart of AUV navigation processing

As shown in Fig. 8, each images from path A-B will be marked as an index vector, such as {position 1,position 2, position 3, position 4, position 5, position 6}. Then when the AUV repeats the path from B-A, the images from B-A are used to match with the images from A-B, and based on the similarity calculation results, each image from B-A will be

localized and determined as existing index. In this way, AUV can be navigated based on matching index vector.

4. EVALUATION AND DISCUSSION

The main aim of our evaluations is to check if our proposed deep model can match the correct image with same location from two different orientations. To evaluate the proposed model, we separate the original sonar images with two different orientations into training data and test data. Fig. 9 shows the matching results with different locations.

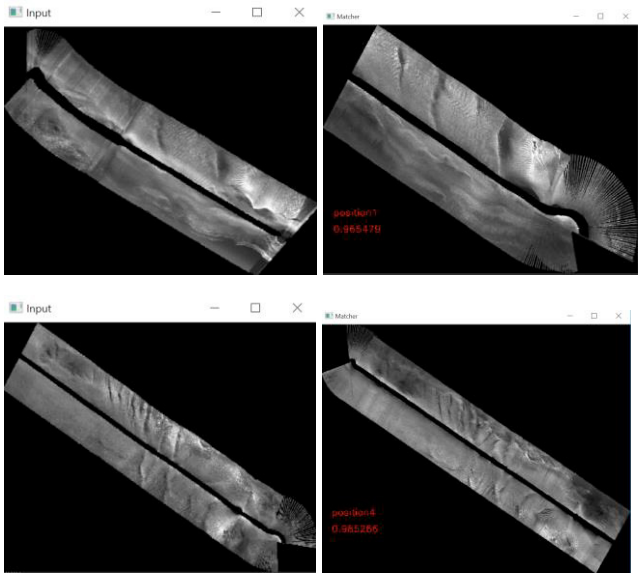


Fig. 9. Examples of sonar image matching results using the proposed CNN model(position 1 and 4).

Specifically, we used 6 sonar images from location A-B of the AUV to generate the matching model and different images from location B-A to test the model. In addition, we also repeat the evaluation using 6 images from location B-A to generate the model and use images from location A-B to test the model. We used the similarity to concisely describe the accuracy and quality of matching degree. Table 3 and 4 shows the evaluation results of these two evaluations respectively.

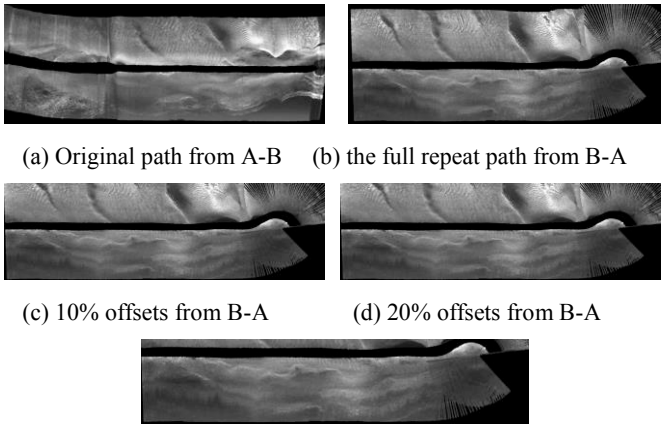
Table 3. Matching results: location A-B to generate model, and location B-A to test model

Location	Matching Similarity					
	1	2	3	4	5	6
1	1.000	0.000	0.000	0.000	0.000	0.000
2	0.018	0.743	0.105	0.016	0.028	0.090
3	0.000	0.000	0.990	0.000	0.010	0.000
4	0.000	0.000	0.019	0.978	0.003	0.000
5	0.000	0.001	0.019	0.007	0.887	0.070
6	0.123	0.036	0.004	0.017	0.001	0.819

Table 4. Matching results: location B-A to generate model, and location A-B to test model

Location	Matching Similarity					
	1	2	3	4	5	6
1	0.615	0.002	0.000	0.000	0.000	0.383
2	0.046	0.302	0.257	0.057	0.049	0.288
3	0.001	0.005	0.948	0.020	0.020	0.006
4	0.000	0.001	0.000	0.998	0.000	0.001
5	0.000	0.000	0.989	0.000	0.010	0.000
6	0.003	0.014	0.001	0.004	0.002	0.976

We also evaluate the matching accuracy when the repeat path has a certain offset compared with original path. We tested the matching accuracy when the UAV has 10%, 20% and 40% offsets, which means the test images only include some parts of features.



(a) Original path from A-B (b) the full repeat path from B-A

(c) 10% offsets from B-A (d) 20% offsets from B-A

(e) 40% offsets from B-A

Fig. 10. The 10%, 20% and 40% offsets of test images

Table 5, 6 and 7 shows the evaluation results of these three offsets evaluation results respectively (the positions marked as orange colour are incorrect matching).

Table 5. Matching results: location A-B to generate model, and 10% offset of location B-A to test model

Location	Matching Similarity					
	1	2	3	4	5	6
1	1.000	0.000	0.000	0.000	0.000	0.000
2	0.045	0.693	0.081	0.023	0.086	0.073
3	0.000	0.000	0.997	0.000	0.002	0.000
4	0.000	0.000	0.000	0.770	0.212	0.000
5	0.000	0.462	0.283	0.045	0.001	0.210
6	0.080	0.060	0.006	0.007	0.002	0.845

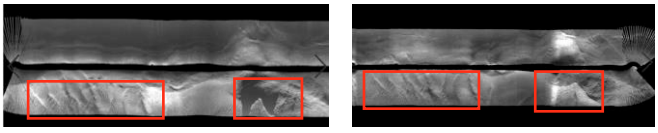
Table 6. Matching results: location A-B to generate model, and 20% offset of location B-A to test model

Location	Matching Similarity					
	1	2	3	4	5	6
1	1.000	0.000	0.000	0.000	0.000	0.000
2	0.013	0.896	0.059	0.007	0.015	0.010
3	0.000	0.005	0.937	0.000	0.000	0.051
4	0.000	0.000	0.031	0.045	0.961	0.000
5	0.000	0.462	0.072	0.045	0.000	0.824
6	0.153	0.072	0.008	0.005	0.003	0.759

Table 7. Matching results: location A-B to generate model, and 40% offset of location B-A to test model

Location	Matching Similarity					
	1	2	3	4	5	6
1	1.000	0.000	0.000	0.000	0.000	0.000
2	0.006	0.012	0.034	0.001	0.871	0.076
3	0.017	0.337	0.342	0.013	0.002	0.289
4	0.000	0.000	0.000	0.000	1.000	0.000
5	0.001	0.000	0.001	0.002	0.000	0.996
6	0.117	0.048	0.001	0.008	0.004	0.823

From the above evaluation results, we can find the matching accuracy of the AUV being located is an exact match (100% similarity) show in Table 3, while using the opposite path, the position 5 is matched incorrectly. The main reason about the difference of test accuracy between two orientations is because the difference between training samples based on the selected region proposals. For example, for the images of position 5.



(a) Position 5: B-A (b) Position 5: A-B

Fig.11. The selected region proposals about two orientations

As shown in Fig. 11, for the two orientations (A-B and B-A), we train and test the model based on the different selected region proposals rather than the whole image. In this way, we can ignore the useless noise in the global image to improve the test accuracy. However, this still has some shortcomings if the selected region proposals are similar between different positions.

In addition, along with the increasing of offset percentage, the matching accuracy will be also decreased. It is because if some important features are excluded, it will affect the matching accuracy. The way to improve the matching accuracy under this case is to use both full ROI and partial ROI images together to train the model.

6. CONCLUSION

In this paper we have shown how to help an AUV navigate using raw sonar image pixels matching based on a CNN-based model. We reviewed the two main types of deep learning methods used in image matching and choose the two-stage method to find a small set of ROIs which tightly cover as many useful features in the sonar image as possible to improve the matching accuracy.

Firstly, we used selective searching method to select a small set of ROIs to input into the model, then a 5-layered CNN-based model with batch normalization and early stopping is used to train the region proposals and generate the matching model. At the end, we evaluate our proposed model with different AUV offsets (0%, 10%, 20% and 40%) to compare the matching accuracy.

Our future work will try to improve the matching accuracy in the case of larger offsets using both full ROIs and partial ROIs to generate the matching model. Furthermore, we will also widen the collected datasets for other complex scenes in deep ocean to capture all the possible cases, such as offsets with rotation to evaluate our proposed model and improve the performance.

REFERENCES

L. Paull, S. Saeedi, M. Seto, and H. Li. (2014). AUV navigation and localization: A review, IEEE Journal of Oceanic Engineering, vol. 39, no. 1, pp. 131-149.

J. C. Kinsey, R. M. Eustice, and L. L. Whitcomb. (2006). A survey of underwater vehicle navigation: Recent advances and new challenges, in IFAC Conference of Manoeuvering and Control of Marine Craft, vol. 88, pp. 1-12.

P. King, A. Vardy, and A. L. Forrest. (2018). Teach - and - repeat path following for an autonomous underwater vehicle, Journal of Field Robotics, vol. 35, no. 5, pp. 748-763.

F. Remondino, M. G. Spera, E. Nocerino, F. Menna, and F. Nex. (2014). State of the art in high density image matching, The photogrammetric record, vol. 29, no. 146, pp. 144-166.

J. Aulinas et al. (2011). Feature extraction for underwater visual SLAM, in OCEANS 2011 IEEE, pp. 1-7.

K. Kavukcuoglu, R. Fergus, and Y. LeCun. (2009). Learning invariant features through topographic filter maps, in IEEE Conference on Computer Vision and Pattern Recognition, pp. 1605-1612.

C. Prabhakar and P. Kumar. (2012). LBP-SURF descriptor with color invariant and texture based features for underwater images, in Proceedings of the eighth Indian conference on computer vision, graphics and image processing, p. 23.

Zhongqui Zhao, Peng Zheng, Shoutao Xu, Xindong Wu. (2019). Object Detection With Deep Learning: A Review, IEEE Transactions on Neural Networks and Learning Systems, pp. 1-21.

Russ, John C. (2016). The image processing handbook, CRC press.

P. Zhu, J. Isaacs, B. Fu, and S. Ferrari. (2017). Deep learning feature extraction for target recognition and classification in underwater sonar images, IEEE 56th Annual Conference on Decision and Control (CDC), pp. 2724-2731.

X. Cao, X. Zhang, Y. Yu, and L. Niu. (2016). Deep learning-based recognition of underwater target, in IEEE International Conference on Digital Signal Processing (DSP), pp. 89-93.

Z.C. Horn, L. Auret, J.T. McCoy, C. Aldrich, B.M. Herbst. (2017). Performance of Convolutional Neural Networks for Feature Extraction in Froth Flotation Sensing, International Federation of Automatic Control, vol. 50, no. 2, pp. 13-18..

S. Zagoruyko and N. Komodakis. (2015). Learning to compare image patches via convolutional neural networks, in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4353-4361.

L. Gatys, A. S. Ecker, and M. Bethge. (2015). Texture synthesis using convolutional neural networks, in Advances in neural information processing systems, pp. 262-270.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. (2016). You only look once: Unified, real-time object detection, in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788.

W. Liu et al. (2016). Ssd: Single shot multibox detector, in European conference on computer vision, pp. 21-37.

R. Girshick, J. Donahue, T. Darrell, and J. Malik. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation, in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580-587.

R. Girshick. (2015). Fast r-cnn, in Proceedings of the IEEE international conference on computer vision, pp. 1440-1448.

S. Ren, K. He, R. Girshick, and J. Sun. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks, in Advances in neural information processing systems, pp. 91-99.

J. Dai, Y. Li, K. He, and J. Sun. (2016). R-fcn: Object detection via region-based fully convolutional networks, in Advances in neural information processing systems, pp. 379-387.

J. Huang et al. (2017). Speed/accuracy trade-offs for modern convolutional object detectors, in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7310-7311.

J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. (2013). Selective search for object recognition, International journal of computer vision, vol. 104, no. 2, pp. 154-171.

G. Cheng, P. Zhou, and J. Han. (2016). Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images, IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 12, pp. 7405-7415.

Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. (2007). Greedy layer-wise training of deep networks, in Advances in neural information processing systems, pp. 153-160.