IMPROVED DEEP-LEARNING-BASED CLASSIFICATION OF MINE-LIKE CONTACTS IN SONAR IMAGES FROM AUTONOMOUS UNDERWATER VEHICLES

Abdesselam Bouzerdoum[a], Philip B. Chapple[b], Mark Dras[c], Yi Guo[d], Len Hamey[c], Tahereh Hassanzadeh[c], Thanh Hoang Le[a], Omid Mohamad Nezami[c], Mehmet Orgun[c], Son Lam Phung[a], Christian Ritz[a], Maryam Shahpasand[c]

[a] University of Wollongong, Wollongong, NSW, Australia
[b] Defence Science & Technology Group, Australia
[c] Macquarie University, Sydney, Australia
[d] Western Sydney University, Sydney, Australia

Abstract: *This paper describes recent work conducted by a team of researchers from three universities in partnership with defence researchers to investigate deep learning methods for automatic detection of mine-like objects from sidescan sonar images captured by autonomous underwater vehicles. While deep learning can produce state-of-the-art classification performances in several application domains, it often relies on a large amount of labelled training data, which is difficult to obtain in our application. To address this problem, we investigate the use of data augmentation, transfer learning, and compact neural networks. For data augmentation, approaches for increasing the size of the training data are investigated, including standard image processing and manual segmentation. For transfer learning, we use publicly available convolutional neural networks (CNNs) pre-trained on large image datasets, and replace later layers with classifiers trained on sonar image data. For compact neural networks, we train a custom small-sized CNN and also process only the region-of-interest in a sonar snapshot. The proposed techniques are evaluated on a data set consisting of three classes: mine-like objects, non mine-like objects, and false alarm objects. The experimental results indicate the feasibility of the proposed techniques, with a classification accuracy of 98.3%.*

Keywords: *Automatic target recognition, sonar image processing, mine-like object detection, convolutional neural network.*

1. INTRODUCTION

Automatic detection of mine-like objects from sidescan sonar images captured by autonomous underwater vehicles has been an active research topic [1-4]. Existing systems include a detection stage, which identifies regions-of-interest based on highlights, shadows and other image features [1, 3], followed by a classification stage to differentiate mine-like objects (MLOs), non mine-like objects (NMLOs), and false alarm objects (FAOs) (also known as false alarms [3]). However, accurate detection relies on optimising the algorithms using a large number of labelled data which are difficult and costly to collect. Furthermore, the available data may not cover the varying underwater environments where mines are found. Recently, researchers have investigated the use of deep learning for detection and classification of MLOs from sonar images [4-7]. In [7], a convolutional neural network (CNN) was used to classify snapshot images, produced by the first stage of the detection system, into three categories: MLO, NMLO and FAO. This CNN classifier is needed to improve the overall accuracy of the mine detection system.

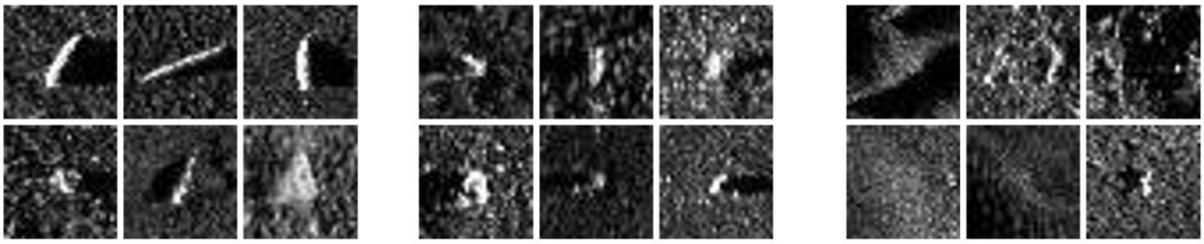
This paper describes several approaches to improve the performance of deep learning based techniques in underwater mine detection. First, we investigate data augmentation techniques to increase the size of the data set for training the CNN classifiers. Second, we explore the use of transfer learning, where the final layer of an existing pre-trained CNN is replaced by a more powerful classifier trained on the sonar data. Third, we design a compact CNN classifier to enhance the generalisation capability from limited training data. Lastly, we investigate the approach of using only the object-of-interest and shadow in classification.

The remaining of the paper is structured as follows. Section 2 of this paper provides an overview of the automatic target recognition system and sonar data collection. Section 3 presents four proposed techniques: data augmentation, transfer learning, designing a compact CNN, and processing only object-of-interest and shadow. Section 4 presents experimental methods and results of different classification methods. Section 5 provides the concluding remarks.

2. AUTOMATIC TARGET RECOGNITION AND SONAR DATA COLLECTION

Sonar data used in this project were collected by the DST Group over the last 10 years using several types of AUVs and sonar scanners: i) a REMUS 100 vehicle and a Marine Sonic Technology (MST) sonar operating at 900 kHz and 1800 kHz; ii) a Gavia vehicle and an MST sonar operating at 600 kHz and 1200 kHz; iii) a REMUS 600 vehicle and a Kraken AquaPix Interferometric Synthetic Aperture Sonar.

The existing SonarDetect software tool, developed by the DST Group for processing the sonar data captured by these AUVs, is described in [7]. This software tool displays sonar imagery alongside the navigation chart. It also indicates objects such as MLOs, NMLOs and FAOs that are automatically found in the image. These objects are detected using several image pre-processing steps that include mitigation of the surface return, detection of highlight and shadow, spatial filtering, clustering of pixels into regions according to size and shape criteria. These pre-processing steps have several parameters that need to be selected to maximise the detection of MLOs while minimising the detection of FAOs (e.g. irrelevant objects such as fish and rocks that should be excluded).



(a) MLO (b) NMLO (c) FAO

Fig. 1: Examples of sonar snapshots in the three categories.

For this study, the sonar images were acquired from a naval mine-shape recovery mission in Jervis Bay, Australia using the above AUVs [3]. The sonar images were processed by the SonarDetect tool to generate snapshots of size 501 × 501 pixels. For experiments, the snapshots were manually labelled by a defence analyst into three categories, MLO, NMLO and FAO. Examples of these snapshots are shown in Fig. 1. Table 1 presents a summary of the snapshots that are used in this study to investigate deep learning-based approaches for sonar snapshot classification.

Table 1: A summary of the sonar snapshots provided by the DST Group.

Category	**MLO**	**NMLO**	**FAO**
Number of snapshots	176	40	196

3. SONAR SNAPSHOT CLASSIFICATION METHODS

3.1. Convolutional Neural Network Classifier

The various methods in this paper are based on the convolutional neural network [8, 9]. Hence, we present here a brief description of the main layers in this architecture. A CNN typically consists of several blocks of convolutional layer, batch normalization layer, rectified linear unit, and pooling layer, followed by a block of fully-connected layer and softmax layer for classification.

- *A convolutional layer* extracts features from its two-dimensional (2-D) input using a convolution operator which preserves the spatial relationship between pixels. The output of a convolution layer is called a feature map, and is computed as the sum of convolutions of the 2-D input with a set of trainable kernels. An adjustable scalar bias is typically added to the sum. Formally, let Y_c^l be the c-th feature map in the l-th convolutional layer, and $K_{c,k}^l$ be the k-th convolutional kernel for the c-th input. The k-th output feature map is computed as

$$Y_k^{l+1} = f(\sum_{c=1}^{C} Y_c^l \otimes K_{c,k}^l + b_k^l), \tag{1}$$

where $\otimes$ denotes the 2-D convolution operator, and f represents a non-linear activation function. Here, b_k^l is a scalar bias for the k-th output feature map.

- *A rectified linear unit* (ReLU) performs a nonlinear activation function to each element of its input. The ReLU function with a threshold of zero is defined as $f(x) = max(x; 0)$.
- *A batch normalization layer* is inserted between a convolutional layer and a ReLU layer to increase the stability of a neural network and accelerate network training [8]. This layer normalizes an output feature map by subtracting the mini-batch mean, and then dividing by

the mini-batch standard deviation. Effectively, this layer aims to provide inputs with a zero mean and unit variance for the subsequent layer in a neural network. Let $\boldsymbol{x} = \{x_1, x_2, \dots, x_m\}$ be the activations over a mini-batch of m samples. The normalized activations $\hat{\boldsymbol{x}}$ are defined as

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}, \tag{2}$$

where $\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i$ is the mini-batch mean, and $\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2$ is the mini-batch variance. Here, ε is a pre-defined positive scalar, added for numerical stability. Furthermore, to improve training, this layer also shifts and scales the normalized activations as

$$\hat{y}_i = \gamma \hat{x}_i + \beta, \tag{3}$$

where γ and β are the trainable parameters.

- *A pooling layer* divides its 2-D input into smaller rectangular regions, and computes the maximum value (i.e. max-pooling) or average value (i.e. average-pooling) of each region. A pooling layer reduces the spatial dimension of the input and hence the computational complexity of the subsequent layers; it also helps extracting image representations that are more stable to variations in the inputs [9].

3.2. Data Augmentation Techniques

We investigate two approaches to data augmentation. The first approach applies simple image processing techniques on the sonar snapshots: reflections, scaling, shear, rotation and translation. These techniques are implemented as run-time randomised augmentations in Keras [10], so that each time a training image is used, a random combined augmentation is applied. We consider random rotations between [-10, +10] degrees, translations up to 10% of the image size, scaling by factor between [1, 1.5], and vertical and horizontal reflections. Among these techniques, only reflections and scaling are found to be useful in improving classification accuracy.

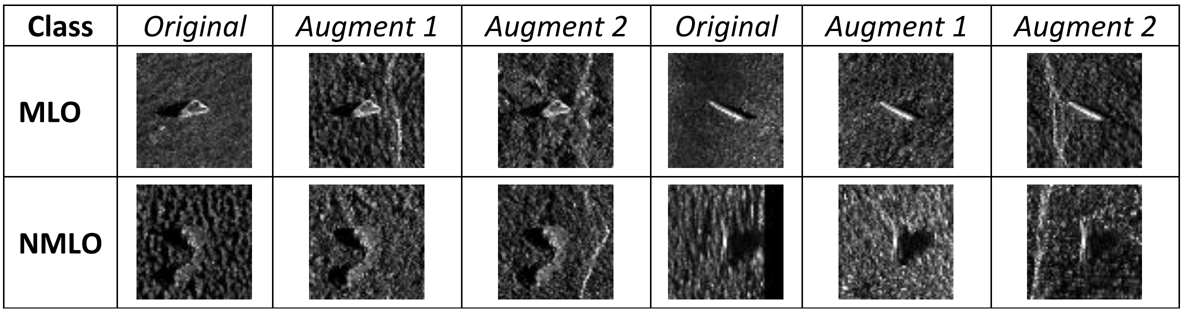


Fig. 2: Examples of snapshots and the corresponding augmented images.

The second approach uses masking to place objects on different backgrounds. To this end, the snapshots in the original data set are manually segmented to mark MLO and NMLO regions. Each marked region is then overlaid on random seabed backgrounds to generate additional training snapshots. During overlaying, the direction of the shadow cast by the object (left or right) is matched to the shadow direction in the background image. Figure 2 shows examples of MLO and NMLO snapshots and the corresponding augmented images. Additional snapshots for the FAO class are generated by adjusting the pre-processing parameters of the SonarDetect tool.

3.3. Transfer Learning with SVM

For transfer learning, we consider several pre-trained CNN architectures including the VGG16 and VGG19. For each network, two approaches are investigated. The first approach replaces the last layers with a fully-connected layer, a softmax layer and a classification output layer. Then, the entire network is trained with snapshot images using a small learning rate. The second approach replaces the final layers of VGG with an support vector machine (SVM) classifier. That is, the SVM is used to classify the feature vector extracted by a pre-trained CNN. In our application, the second approach is found to be more effective than the first, so it is analyzed in more detail in this paper (Section 4).

3.4. Designing a Small-Sized CNN

Large CNNs (such as AlexNet, VGG16, VGG19, Inception) with millions of trainable parameters are prone to overfitting when training with limited data. To overcome this problem, we design a compact CNN with much fewer learnable weights.

The small-sized classifier is designed as a typical CNN comprising two convolutional blocks followed by a classification block, see Fig. 3. The kernel sizes used in the first and second convolutional layers are 5×5 and 3×3 pixels, respectively. The convolutional layers have a total of 33,782 trainable parameters. To stabilize the training, batch normalization layers are applied to the convolutional layers. A max-pooling layer of size 2×2 pixels with stride of 2 and no zero padding is employed after each convolutional unit for spatial dimensionality reduction.

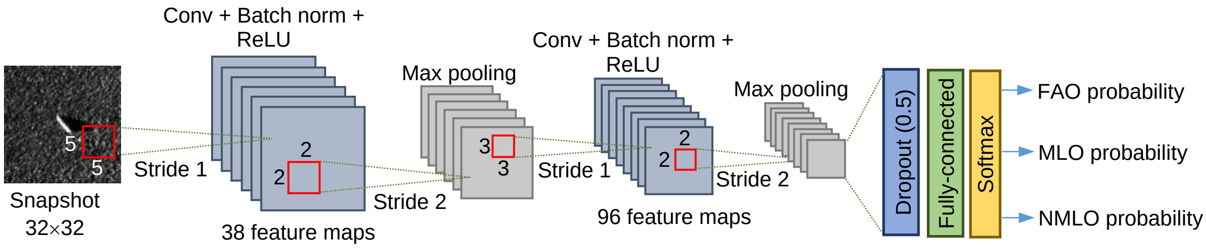


Fig. 3: A custom small-sized CNN for snapshot classification.

3.5. Classifying only Object-of-Interest and Shadow

The existing SonarDetect tool produces, after several pre-processing steps, each snapshot together with information (bounding box) about the object of interest and its shadow (OIS), which can be utilised for classification. We explore a dedicated CNN that considers only the OIS for three reasons. First, if appearance features of sonar snapshots are indeed significant for classification, then background distraction should be minimised. Second, by focusing only on the object of interest and its shadow, we can reduce the input size to the network, and consequently the network structure and the amount of required training data. Third, the dedicated CNN can be modified to perform object detection directly via window scanning on the full image.

In our approach, the tightest patches containing OIS from the snapshot images are extracted. Although the OIS vary in width and height, they are resized to a patch size of 27 × 58 pixels, which we find suitable for most of the snapshots. This resized patch is used as input to the dedicated CNN network, called ObjectNet. Figure 4 shows the extracted OIS of various sizes before rescaling.

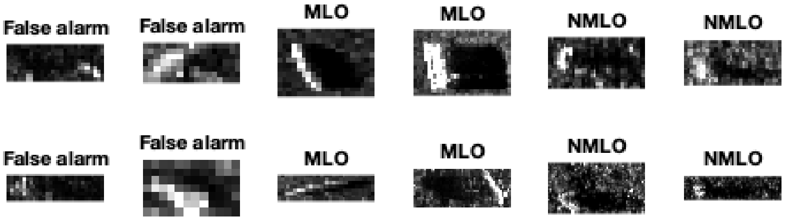


Fig. 4: Examples patches of objects of interest and shadows.

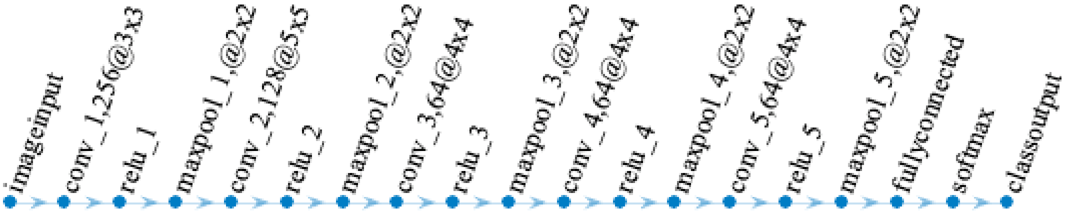


Fig. 5: ObjectNet structure for OIS classification. The notation **conv_1,256@3×3** *means the first convolutional layer has 256 feature maps produced using a kernel size of 3×3 pixels.*

The structure of the ObjectNet is shown in Fig. 5. It consists of a feature extraction block with 5 convolutional layers, ReLU layers, and max-pooling layers, followed by a classification block with a fully connected layer and a softmax layer. The max-pooling layers operate on non-overlapping local regions of size 2 × 2 pixels. Here, batch normalization is not used so that the network can be extended for object detection via window scanning. ObjectNet is trained with the stochastic gradient descent [9], using the augmented data created in Section 3.2. Several rounds of optimisation are performed with batch sizes varying from 50 to 800. For each batch size, the number of training epochs is 800. The best network found at each batch size is used as the initialised network for the next round of optimisation, which uses a larger batch size.

4. EXPERIMENTS AND RESULTS

4.1. Experimental Methods

The five-fold cross-validation technique is used to evaluate the different classification methods. The original snapshots in Table 1 are divided into five approximately equal partitions. For each fold, a partition is used for testing, and the remaining partitions are used for training. This step is performed five times for different choices of the test partition. The classification rates are averaged over the five folds to obtain the estimated classification rate (CR). For training, the data augmentation techniques described in Section 3.2 are applied to generate 11 additional snapshots for each MLO training snapshot, and 47 additional snapshots for each NMLO training snapshot.

The CR for each class and the overall CR are estimated through five-fold cross validation. We also report an additional classification measure: the *balanced* overall CR. As evident in Table 1, the snapshot data are unbalanced among the three classes. This is partially addressed through augmentation of the training data, but testing is still performed only on original snapshots. Therefore, the test data remain unbalanced, and the classification accuracy is biased toward the class with the highest number of test samples. The *balanced* overall CR measure addresses this problem by assigning a weight to each test sample so that all classes have an equal weighted number of test samples.

In this application, the end-user is also very interested in differentiating FAOs from the other two classes combined, MLOs and NMLOs. The main reason is that after an MLO *or* NMLO snapshot is detected, further processing can be performed, even by a human operator, to separate an MLO from an NMLO. Therefore, we also assess the performances of different classification methods for the two-class problem: MLO+ NMLO *versus* FAO.

4.2. Snapshot Classification Results

Table 2 presents the three-class classification results for each of the three deep learning approaches investigated in this study: i) transfer learning using a pre-trained VGG network and the SVM classifier (Section 3.3); ii) custom small-sized CNN (Section 3.4); iii) ObjectNet focusing on object of interest and shadow only (Section 3.5). For the transfer learning approach, the CR of the three-class classification is 76.2%, but this CR is affected by the imbalance in the dataset as the *balanced* CR is only 62.9%. For the custom small-sized CNN, the overall CR and the *balanced* overall CR are 98.3% and 98.7%, respectively. For the ObjectNet, the overall CR is 69.9%. The results indicate that a small-sized CNN can achieve quite high accuracy in sonar snapshot classification.

Table 2: Classification rates (%) produced by the transfer learning, the small-sized CNN, and the ObjectNet for the three-class problem.

Method	MLO	NMLO	FAO	Overall	Overall (balanced)
Transfer learning with SVM	84.7	25.0	79.0	76.2	62.9
Small-sized CNN	99.0	97.2	100	98.3	98.7
ObjectNet	82.4	65.0	59.7	69.9	69.0

Table 3: Classification rates (%) produced by transfer learning, the small-sized CNN and the ObjectNet for the two-class problem, MLO+NMLO versus FAO.

Method	MLO+NMLO	FAO	Overall	Overall (balanced)
Transfer learning with SVM	89.3	79.1	84.4	84.2
Small-sized CNN	99.0	98.2	98.5	98.6
ObjectNet	86.6	59.7	73.8	73.2

Table 3 presents the two-class classification results for each of the three deep learning approaches. For this two-class problem, there are 216 NMLO/MLO targets and 196 FAO targets, so the two classes have balanced data. Therefore, the overall CR and the balanced overall CR are quite similar. The overall CRs for the transfer learning VGG+SVM, the small-sized CNN, and the ObjectNet are 84.2%, 98.5%, and 69.9%, respectively.

5. CONCLUSION

This paper has described the application of deep learning to the classification of snapshots in sidescan sonar images collected by an AUV. To address the problem of limited training data and to improve the generalisation capability of the classifier, the paper explores the use of data augmentation, a small-sized network, and object-of-interest and shadow. The results indicate the feasibility of the proposed techniques, with a classification accuracy of 98.3%. Further work will combine the three proposed techniques and explore more advanced data augmentation methods.

REFERENCES

[1] G. J. Dobeck, J. C. Hyland, and L. Smedley, "Automated detection/classification of sea mines in sonar imagery," *Proc. SPIE 3079*, pp. 90-110, 1997.

[2] J.A. Fawcett, "Automatic target recognition methods for sidescan sonar images: The advances and the challenges," *Proc. Int. Conf. on Detection and Classification of Underwater Targets,* pp. 2-18, 2012.

[3] P. B. Chapple, "Unsupervised detection of mine-like objects in seabed imagery from autonomous underwater vehicles," *Proc. IEEE OCEANS Conf.*, pp. 1–6, 2009.

[4] D. P. Williams, "Underwater target classification in synthetic aperture sonar imagery using deep convolutional neural networks," *Proc. Int. Conf. Pattern Recognition,* pp. 2497–2502, 2016.

[5] P. Zhu, J. Isaacs, B. Fu, and S. Ferrari, "Deep learning feature extraction for target recognition and classification in underwater sonar images," *Proc. IEEE Annual Conf. Decision and Control*, pp. 2724–2731, 2017.

[6] K. Denos, M. Ravaut, A. Fagette, and H. Lim, "Deep learning applied to underwater mine warfare," *Proc. IEEE OCEANS Conf.*, pp. 1–7, 2017.

[7] P. B. Chapple, T. Dell, and D. Bongiorno, "Enhanced detection and classification of mine-like objects using situational awareness and deep learning," *Proc. Underwater Acoustics Conference and Exhibition (UACE)*, pp. 529-536, 2017.

[8] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Proc. Int. Conf. on Int. Conf. Machine Learning*, pp. 448-456, 2015.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature,* 521(7553), pp. 436-444, 2015.

[10] https://keras.io