INDIVIDUAL SHIP DETECTION USING UNDERWATER ACOUSTICS[1]

Damianos Karakos Jan Silovsky Richard Schwartz William Hartmann John Makhoul

Raytheon BBN Technologies
Cambridge, MA, USA

{damianos.karakos,jan.silovsky,rich.schwartz,william.hartmann,john.makhoul}@raytheon.com

ABSTRACT

Individual ship detection from underwater audio is the task of deciding whether a specific ship is present, using sound captured by an underwater hydrophone. It is a task analogous to speaker identification (SID), in the sense that it is an open-class detection task; the ships present could be other irrelevant ("impostor") ships, never encountered in the training data. We present two methodologies for tackling this problem, both motivated by our work in speech-related technologies: (i) one based on neural networks, which follows, to a large extent, the approach of [1], and (ii) one based on i-vectors and PLDA [2]. To the best of our knowledge, this is the first time that the topic of individual ship detection is approached as an open-class detection problem.

Index Terms **—** Sonar, ship detection, neural networks, speaker identification.

1. INTRODUCTION

Automatically identifying ships in the ocean is an important undertaking, for both commercial and military applications. The sounds emitted by ships can be captured by underwater hydrophones and then used for such a task. The assumption is that these sounds, which are emitted by moving parts, such as engines and propellers, are correlated with the class label of interest. A number of papers (e.g., [1] [3]) have shown that it is possible to perform ship-type *classification* successfully using underwater sound captured with just a single hydrophone.

Ship-type classification assumes that we have a closed set of target labels. The task is to output one (and only one) of these labels which minimizes the error. In this paper, we focus on a different (albeit, related) task, that of individual ship *detection*, where the goal is to detect whether a specific ship of interest (from a predefined set of ships) is present in a piece of audio or not. Detection is of particular interest in many applications; its main advantage is that we don't have to assume that we have a closed set of ships. Specifically, if the audio is from another ship (an "impostor"), that ship could be anything, including one that was never seen in the training data. A successful detection system has to be able to reject the bulk of these instances without significantly affecting the detection of true occurrences of ships of interest. In this sense, individual ship detection is analogous to speaker detection (also known as speaker identification (SID) [2]), where the goal is to detect whether a speech sample is from a specific speaker or not. Note that the task tackled here is distinct from the (much easier) binary detection task of ship vs. no-ship, whose goal is to determine whether there is a ship (that is, *any* ship) present or not [1]. Having explained this distinction, and, to simplify our exposition, we will refer to individual ship detection as ship detection in the rest of the paper.

In this paper, we present details about our ship detection system, using data collected by the Scripps Institution of Oceanography, UCSD, with a single hydrophone setup. The feature extraction follows the pipeline in [1]. Modeling is done either (i) with a neural network, or (ii) an approach that uses i-vectors [4] and PLDA [5]. The scores generated from these approaches are then processed further, through a normalization process, which makes the scores more comparable across different ships and is shown to improve global performance.

The paper is organized as follows: Section 2 gives details about the corpus used in our experiments. Section 3 describes the various components of the detection system. Section 4 gives details about the score normalization algorithm. Section 5 presents experimental results from ship detection, and Section 6 presents concluding remarks.

2. CORPUS

All our experiments were conducted with the sonar data collected with a single omnidirectional hydrophone by the Scripps Institution of Oceanography, UCSD, San Diego, CA over a period of 9 years (part of this corpus was used in [6]). The hydrophone was located off the coast of Santa Barbara, at a depth of about 600 m. The data was originally sampled at 200 kHz and subsequently downsampled to 10 kHz before being delivered to us. Scripps also provided us with data from the Automatic Identification System (AIS) which contains information broadcast regularly by most commercial ships, such as time stamp, GPS coordinates, unique ship identification, speed, etc. This data set is described in more detail in [1]. The data selection and the final corpus preparation is,

[1] We would like to thank John Hildebrand at the Scripps Institution of Oceanography, UCSD, San Diego, CA for providing us with this large collection of underwater recordings. We would also like to thank our BBN colleagues George Shepard, Edin Insanic, and Stavros Tsakalidis for useful discussions. The research presented in this paper was sponsored by the Defense Advanced Research Projects Agency (DARPA) Information Innovation Office (I2O) under the Robust Automatic Transcription of Speech (RATS) program, issued by DARPA/CMO under Contract No. HR0011-15-C-0038. The views, opinions, and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations. Distribution Statement A: Approved for Public Release, Distribution Unlimited.

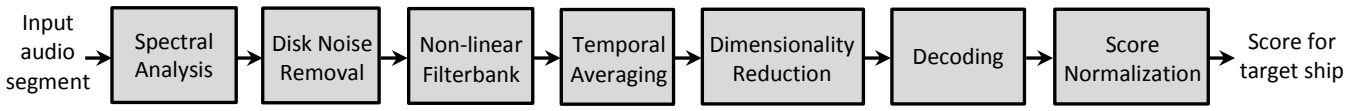

Fig. 1. Block diagram of the detection system pipeline.

however, done differently here. In the corpus of [1], for ship-type classification, we constrained the training and test ships for each ship type to be distinct in order to demonstrate the ability to do ship-type classification on *unseen* ships. In contrast, here, in order to detect individual ships, we had to ensure that the ships that we were trying to detect were in both the training and test data.

Similar to [1], we imposed a constraint that passages of different ships should not overlap in time (detecting ships in the presence of other ships is beyond the scope of this paper). For this reason, we kept portions of passages that were within d_1= 6 km from the hydrophone, only when no other ships were within a distance d_2 = 10 km at the same time. Thus, we implicitly assumed the acoustics of ships beyond the 10 km limit would be quiet enough so as not to affect our results – an assumption that might not always be true. We also imposed the constraint that there are at least 2 passages per ship.

We have labels for all of the ships, defined by the AIS data, but in order to come up with a realistic scenario, we divide the ships into three categories:

a) **Labeled ships:** these are ships that exist in the training data and for which we have the exact ship id. These are used as "target" ships that we would try to detect.
b) **Unlabeled ships:** these are ships that also exist in the training data, but for which (we pretend) we do not have a detailed label. These are all categorized with a single "other" class label.
c) **Unseen ships:** these are ships that do not exist in the training data.

We perform detection experiments using each of the target ships in category (a), by trying to correctly detect true instances of that ship (target trials). We also test each target with "impostor" trials using audio from ships in all three categories. So the impostor could be another ship for which we have an explicit model (a), or another ship that was in the "other" category (b), or a unseen ship (c). We compute the Miss and False Alarm rate for each target ship and then average each of the two error rates across all ships.

The ships were divided into the three categories based on the number of passages. The top 100 ships with the most passages were placed in category (a), the next 100 were placed in category (b) and the rest in category (c).

The data was split into train/test such that, for any given ship, all of the training data predates all of the test data, and the training data contains at least half of the passages from each ship. Furthermore, in order to leave as much data available for training as possible, we imposed an upper bound of 5 on the number of passages per ship in the test data.

A development (dev) set was carved out from the training set, making sure that all ships are represented in both sets and that the distribution of dev ship passages follows that of the test set. The purpose of the dev set is two-fold: (i) to tune parameters, such as the size of the neural network, and (ii) to estimate normalization maps (more details about such maps appear in a later section). To ensure that we would have a representative dev set without compromising much on the size of the training set, the dev set contained about six times fewer passages than the training set.

We provide several statistics for this corpus in Table 1. The bulk of the passages in the test data are from the unseen ships to mimic the realistic scenario of trying to detect only a limited number of ships of interest from among a potentially large set of ships that appear in the ocean and are not of interest. The training set statistics are for the final set (after excluding the dev set passages).

	TRAIN	DEV	TEST
# unique ships	200	200	611
# passages	1539	259	1764
# ships in (a)	100	100	100
# passages in (a)	1337	159	437
# ships in (b)	100	100	100
# passages in (b)	202	100	246
# ships in (c)	-	-	411
# passages in (c)	-	-	1081

Table 1. Statistics of the ship detection corpus.

3. DETECTION SYSTEM

The detection system consists of various components shown in Figure 1.

A. Feature extraction at the frame level

Similar to [1], a feature vector is computed at every frame of audio. The sequence of processing steps consists of (i) short-time spectral analysis, using a 3 s analysis window, computed every 1 s; (ii) disk noise removal, where some artifacts in the audio, caused by a disk drive while recording data, are removed; (iii) a non-linear filterbank, where the 3000 spectral values between zero and 1 kHz are summed into 400 triangular filters – the filters are roughly equally spaced up to 65 Hz and log-spaced above 65 Hz; (iv) temporal averaging, where the filter energies, computed every 1 s, are averaged over an interval of 3 s, with the averaging repeated every 1 s; (v) dimensionality reduction, where the 400-dimensional feature vector at each frame is reduced to a smaller dimension using PCA; and (optionally) (vi) i-vector computation [4], which reduces the ensemble of feature vectors of a whole segment of interest into a single vector (more details about this appear below).

B. Modeling

We have experimented with two modeling approaches: (i) Using a neural network that outputs posteriors over the classes (ships) of interest, and (ii) using i-vectors and PLDA.

Modeling with a neural network

We use a neural network (NN) with one hidden layer (more layers did not improve performance). The input to the neural network is one feature vector corresponding to one frame. The output layer has one output for each of the 100 labeled ships in category (a)

plus one output for the "other" category, resulting in 101 posteriors. (Note that these posteriors were used only for computing the detection score for each ship, and not for classification.) The training is done with the backpropagation algorithm and the cross-entropy optimization criterion. The network weights are initialized with RBM pretraining [7].

Modeling with i-vectors and PLDA

This approach is described in detail in [2], but we give a summary here. First, we use PCA to reduce the dimensionality of the feature vector from 400 to 60 dimensions. Then, we estimate a universal background Gaussian mixture model (GMM) from the spectral features. For any given segment of audio (sequence of frames), we compute per-Gaussian posteriors and perform maximum aposteriori (MAP) adaptation of the means of the GMM. These modified means are then concatenated together into a single vector (the so-called "supervector"). The supervector is subsequently projected into an "i-vector" space [4], of much reduced dimensionality (60). I-vectors contain not only the information relevant for ship classification, resulting in the useful between-class variability, but also all kinds of nuisance information, which results in within-class variability. We can refer to this nuisance information as segment variability in our case. PLDA [5] defines an i-vector generative model that models these variabilities with the aim of emphasizing the ship variability and neglecting the passage/segment variability. Given a pair of two sets of i-vectors – one set extracted for the ship enrollment data from a target ship and the other one extracted from a segment of test data (formed by a single i-vector in our case) – PLDA allows us to compute a score in the form of a log-likelihood ratio between the hypotheses that a) both sets of i-vectors belong to the same ship and b) they belong to different ships.

C. *Generating detection output at the segment level*

We consider segments of duration 5 seconds, 1 minute, and 5 minutes for detection. That is, we split each passage in the test data into these short segments and perform detection on each one separately from the others. This "memoryless" style of detection was done in order to mimic a scenario where the goal is to detect a target as accurately as possible within a prescribed amount of time, without using any prior information about it (e.g., as soon as the system receives a signal from a novel source). For the case of the NN pipeline, conversion of frame-level posteriors into segment posteriors is done by computing the arithmetic mean of the posteriors over the frames of each segment, as explained in [1]; this was found to be more robust than computing the geometric mean. For the case of the i-vector and PLDA pipeline, since each i-vector is generated from the whole segment of interest (5 seconds, 1 or 5 minutes), segment-based scores are generated directly and no conversion from frame scores is needed.

For each segment, if the target ship has a score above a certain threshold, then that target is hypothesized as being present. Clearly, a segment can have multiple ships hypothesized as being present. Segment scores (log-posteriors or log-likelihood ratios) are optionally converted into normalized segment scores using either the "pFA-normalization" approach of [8], [9] or the "linear-fit" approach of [10]. Score normalization makes the scores of different target labels commensurate, so that when we sweep a single threshold we obtain a better Receiver Operating Characteristic (ROC) curve. More details about this normalization scheme appear in the next section.

D. *Evaluation*

The False Alarm (FA) and Miss rates for a target class c and a detection threshold t are computed as:

$$pFA(c,t) = \frac{\#\text{ segments falsely accepted as } c}{\#\text{ segments that do not contain } c}$$

$$pMiss(c,t) = \frac{\#\text{ segments of } c \text{ falsely rejected as not } c}{\#\text{ segments that truly contain } c}$$

For any choice of t, the average FA and Miss rates are defined as follows:

$$pFA(t) = \frac{1}{D}\sum_c pFA(c,t)\,,\;\; pMiss(t) = \frac{1}{D}\sum_c pMiss(c,t)$$

where D is the number of detailed labels (100). By sweeping the threshold t, we can trade off the *pFA(t)* against the *pMiss(t)*. In all of our experiments, we report the Equal-Error-Rate (EER), which is a single-number measure of the performance and is that point on the ROC curve where *pFA = pMiss*.

4. SCORE NORMALIZATION

Since our performance measure depends on a global ranking of the detections, raw posterior scores may not be optimal for ranking, as they are not necessarily commensurate across classes. For example, detections of a class may have low posterior in general (too conservative learner), or detections of another class may have high posterior in general (too confident learner). *Score normalization* was introduced in [8], [9] as a way to fix this problem: we learn to map raw posteriors to scores that are less class-dependent. As was shown in these papers, all score normalization methods investigated resulted in significant performance improvements in keyword spotting. Here, we mainly focus on two of the methods: (i) pFA normalization and (ii) linear-fit normalization.

A. *pFA normalization*

The goal of pFA normalization [8], [9] is to map the posterior of each ship to the corresponding pFA value that results when setting the decision threshold at that posterior. A dev set (described in Section 2) is used for estimating these ship-dependent maps. Note that the dev set does not need to contain any true samples of a ship to estimate this map; it only needs scores for other ships against the model for this target ship.

At test time, a posterior is mapped to the linear interpolation of the mapped values of the two closest training posteriors.

B. *Linear-fit normalization*

This type of normalization was described in [10]. It entails computing a ship-dependent linear mapping between log-posterior and log-rank. To better emphasize the fit at higher posteriors (lower ranks), the data used to estimate the mapping is sampled

more densely in that region. As with pFA normalization, the linear fit is estimated on the dev set and is then applied on the test set.

5. EXPERIMENTS

Ship detection results (EER) are now reported for the dataset that was specified in Section 2. All tuning of parameters (e.g., size of hidden layer in the NN) was done on the dev set. So, the reported results on the test set are fair.

Table 2 shows results obtained on 1-minute segments of the test set. The NN pipeline uses 300 nodes in the hidden layer, while the i-vector+PLDA pipeline uses a UBM with 16 Gaussians, and 60-dimensional i-vectors. Each column in the table corresponds to one of the normalization schemes, with "original" referring to the raw posteriors without normalization. The rows give the results for different subset trials, including "all ships", "labeled" ships only, "unlabeled", and "unseen". As can be seen, both normalization schemes perform better than the original, with pFA normalization performing better than linear fit in the case of the NN pipeline, while the opposite happens with the other pipeline. Interestingly, the NN pipeline performs almost equally well on segments of labeled ships and unlabeled ships, while the alternative pipeline performs best on the segments of unlabeled ships, possibly because it can model the unlabeled ships better as a group. Furthermore, the i-vector+PLDA pipeline is better than the NN pipeline on the unseen ships, while the opposite happens for the labeled ships.

	Original	pFA norm	Linear fit
NN pipeline			
All ships	21.1	**20.3**	20.6
Labeled	20.2	**19.4**	19.7
Unlabeled	20.1	**19.4**	19.7
Unseen	21.6	**20.8**	21.1
i-vector+PLDA pipeline			
All ships	21.5	20.6	**20.4**
Labeled	21.6	20.6	**20.5**
Unlabeled	21.2	20.2	**20.0**
Unseen	21.5	20.6	**20.5**

Table 2. Ship detection performance (EER) results with the NN and i-vector+PLDA pipelines for 1-minute segments. Results are shown for each subset of the impostors separately. The best result in each row is shown in **bold**.

Given the complementary strengths of the two modeling approaches, we show, in Table 3, the result of using a simple linear combination of the detection scores of the two methods:

$$s = w \cdot s_1 + (1 - w) \cdot s_2$$

where s_1, s_2 are the scores of the NN and the i-vector+PLDA methods for a particular test sample, respectively.

Data Set	Fusion EER
All	18.8
Labeled	18.1
Unlabeled	18.1
Unseen	19.2

Table 3. EER Results with a fusion of the NN and i-vector+PLDA

We used a single weight w for all target ships and all test segment trials and we tuned it on the dev set. The optimized weight w was 0.7. The EER decreased by 7.8% relative to the best method (the NN method).

The following three tables show the effect of different experimental conditions on the results for the NN method and pFA normalization. Table 4 shows EER for various segment sizes. As expected, EER is reduced with longer segments, as longer segment length provides more robust estimation of class posteriors.

Segment duration	EER
5 seconds	22.7
1 minute	20.3
5 minutes	**19.7**

Table 4. EER for various segment durations.

Table 5 shows EER as a function of the maximum number of passages per ship in the training data. As expected, performance improves with more training data.

Maximum # passages/ship	EER
1	35.4
2	28.8
3	26.8
4	26.1
5	24.6
20	**20.3**

Table 5. EER as a function of the maximum number of passages per ship in the training data.

Finally, Table 6 shows EER as a function of the distance of the ship from the hydrophone. To make results comparable, the EERs are all computed based on the ships present between 3-4 km (77 unique ships). Performance degrades with distance, which is an expected consequence of lower SNR with increased distance.

Distance (km)	EER
3-4	**16.8**
4-5	19.5
5-6	21.1

Table 6. EER as a function of the distance from the hydrophone.

6. CONCLUDING REMARKS

In this paper, we showed that two modeling approaches from the speaker identification field, a general one (neural networks) and a more specialized one (i-vectors and PLDA), can be used to tackle the individual ship detection task and result in almost identical performance. We found that the NN pipeline works best on trials with labeled ships seen in the training data and the i-vector+PLDA pipeline works best on (impostor) trials with unlabeled (or unseen) ships. A combination of these two gives gains on top of the best system.

REFERENCES

[1] D. Karakos, W. Hartmann, S. R., J. Makhoul, S. Tsakalidis, E. Insanic and G. Shepard, "Applying Speech Technology to the Ship-Type Classification Problem," in *OCEANS 2017*, Anchorage, AK, 2017.

[2] P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam and P. Dumouchel, "PLDA for speaker verification with utterances of arbitrary duration," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, 2013.

[3] J. B. O. S. Filho and J. M. de Seixas, "Class-modular multi-layer perceptron networks for supporting passive sonar signal classification," *IET Radar, Sonar and Navigation,* vol. 10, no. 2, pp. 311-317, 2016.

[4] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," *IEEE Transactions on Audio, Speech and Language Processing,* vol. 19, no. 4, pp. 788-798, 2011.

[5] S. J. D. Prince and J. H. Elder, "Probabilistic Linear Discriminant Analysis for Inferences About Identity," in *IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, 2007.

[6] M. McKenna, D. Ross, S. Wiggins and J. Hildebrand, "Underwater radiated noise from modern commercial ships," *Journal of the Acoustical Society of America,* vol. 131, no. 1, pp. 92-103, 2012.

[7] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science,* vol. 313, no. 5786, pp. 504-507, 2006.

[8] B. Zhang, R. Schwartz, S. Tsakalidis, N. L. and M. S., "White Listing and Score Normalization for Keyword Spotting of Noisy Speech," in *Interspeech*, Portland, OR, 2012.

[9] D. Karakos, R. Schwartz, S. Tsakalidis, L. Zhang, S. Ranjan, T. Ng, R. Hsiao, G. Saikumar, I. Bulyko, L. Nguyen, J. Makhoul, F. Grezl, M. Hannemann, M. Karafiat, S. I., K. Vesely, L. Lamel and V.-B. Le, "Score Normalization and System Combination for Improved Keyword Spotting," in *ASRU*, Olomouc, Czech Republic, 2013.

[10] D. Karakos, I. Bulyko, R. Schwartz, S. Tsakalidis, L. Nguyen and J. Makhoul, "Normalization of Phonetic Keyword Search Scores," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Florence, Italy, 2014.