Single- and Multi-channel Whistle Recognition with NAO Robots

Kyle Poore(✉), Saminda Abeyruwan, Andreas Seekircher, and Ubbo Visser

Department of Computer Science, University of Miami,
1365 Memorial Drive, Coral Gables, FL 33146, USA
{kyle,saminda,aseek,visser}@cs.miami.edu

Abstract. We propose two real-time sound recognition approaches that are able to distinguish a predefined whistle sound on a NAO robot in various noisy environments. The approaches use one, two, and four microphone channels of a NAO robot. The first approach is based on a frequency/band-pass filter whereas the second approach is based on logistic regression. We conducted experiments in six different settings varying the noise level of both the surrounding environment and the robot itself. The results show that the robot will be able to identify the whistle reliability even in very noisy environments.

1 Introduction

While much attention in autonomous robotics is focused on behavior, robots must also be able to interact and sense trigger-events in a human environment. Specifically, apart from direct interaction between humans and robots, it is appropriate for robots to sense audio signals in the surrounding environment such as whistles and alarms. Digital Audio Signal Processing (DASP) techniques are well established in the consumer electronics industry. Applications range from real-time signal processing to room simulation.

Roboticists also develop DASP techniques, only tailored for their needs on specific kind of robots. Literature shows a whole spectrum of techniques, starting with techniques that aim for the recognition of specific signals with one microphone on one end, e.g. [13], to complete systems that combine the entire bandwidth between single signals to microphones arrays combined with speech recognition and other tasks such as localization on the other end, e.g. [8]. The available literature reveals that there are many cases of audio processing/recognition situations as there are different robots and environments, including real-time processing, combining human speech and other audio signals etc.

A lot of research has been devoted to audio signals featuring humanoid robots, especially in the past decade. Audio signals can be important sensor information as they can be used for various purposes, e.g. for the improvement of the robot's self-localization, the communication between multiple robots, or using the audio signals as the only source for self-localization when an existing Wi-Fi network might be down. A demonstration within the SPL in 2013 in

R.A.C. Bianchi et al. (Eds.): RoboCup 2014, LNAI 8992, pp. 245–257, 2015.
DOI: 10.1007/978-3-319-18615-3_20

Eindhoven by the team RoboEireann revealed how difficult it is to communicate between NAOs on the soccer field in a noisy environment.

The technical committee of SPL announced a challenge where the robots have to recognize predefined static signals emitted by a global sound system. Similar to the horn-like audible alarms in ice hockey, where half-time starts and ends are signaled using the horn, future RoboCup tournaments could rely on this mechanism to signal GameController or referee messages. Teams are also required to bring one whistle that has to be recognized by the teams robots. This part of the challenge brings in a real soccer aspect to the SPL. In this paper, we focus on recognizing the sound of a whistle utilizing several NAO robots. We present two approaches, one general idea of a naive one-channel approach and one using a multi-channel learning approach.

The paper is organized as follows: we discuss relevant work in the next section and describe our approach in Sect. 3. Our experimental setup and the conducted robot tests is explained in Sect. 4. We discuss the pros and cons of our results in Sect. 5 and conclude and outline future work in the remaining Sect. 6.

2 Related Work

When consulting the literature one finds a number of research papers that relate to our work. Saxena and Ng [13] present a learning approach for the problem of estimating the incident angle of a sound using just one microphone, not connected to a mobile robot. The experimental results show that their approach is able to accurately localize a wide range of sounds, such as human speech, dog barking, or a waterfall. Sound source localization is an important function in robot audition. Most existing research investigates sound source localization using static microphone arrays. Hu et al. [4] propose a method that is able to simultaneously localize a mobile robot and in addition an unknown number of multiple sound sources in the vicinity. The method is based on a combinational algorithm of difference of arrival (DOA) estimation and bearing-only SLAM. Experimental results with an eight-channel microphone array on a wheeled robot show the effectiveness of the proposed method. Navigation is part of another study where the authors developed an audio-based robot navigation system for a rescue robot. It is developed using tetrahedral microphone array to guide a robot finding the target shouting for help in a rescue scenario [14]. The approach uses speech recognition technology and using a time DOA method (TDOA). The authors claim that the system meets the desired outcome.

ASIMO, the remarkable humanoid developed by HONDA also uses the auditory system for its tasks. An early paper from 2002 introduces the use of a commercial speech recognition and synthesis system on that robot. The authors state that the audio quality and intonation of voice need more work and that they are not yet satisfactory for use on the robot [12]. Okuno et al. [11] present a later version of ASIMO's ability to use the auditory system for tasks at hand. They use the HARK open-source robot audition software [9] and made experiments with speech and music. The authors claim that the active audition improves the localization of the robot with regard to the periphery.

Speech/dialogue based approaches for the NAO also exist. Kruijff-Korbayová et al. [5], e.g., present a conversational system using an event-based approach for integrating a conversational Human-Robot-Interaction (HRI) system. The approach has been instantiated on a NAO robot and is used as a testbed for investigating child-robot interaction. The authors come to the conclusion that the fully autonomous system is not yet mature enough for end-to-end usability evaluation. Latest research such as the paper by Jayagopi et al. [15] suggest that significant background noise presented in a real HRI setting makes auditory tasks challenging. The authors introduced a conversational HRI dataset with a real-behaving robot inducing interactive behavior with and between humans. The paper however does not discuss the auditory methods used in detail. We assume that the authors use the standard auditory recognition that comes with the NAO.

Athanasopoulos et al. [1] present a TDOA-based sound source localization method that successfully addresses the influence of a robot's shape on the sound source localization. The evaluation is made with the humanoid robot NAO. The authors state that this approach allows to achieve reliable sound source location.

All mentioned approaches differ from our approach (a) in the method used, (b) in the purpose of the audio recognition, and (c) in us using the RoboCanes framework. Here, all audio modules have been implemented from scratch and run within the robot's system loop. We are synchronizing the audio signals with the update of the vision system of our NAO robots.

3 Approach

The recognition of whistle sounds will provide information that can be used by the behavior control of the robot to react to signals, which, for example, may be given by a referee. The behavior is mostly based on information gained from the camera images. Therefore, most behavior modules are running in a control loop synchronized with the camera (in our experiments 30 fps). To minimize the delay in reacting to whistle signals, we need to run the audio processing with the same rate. In every cycle of the decision making, the whistle detection needs to check the most recent audio data from the microphones. However, integrated in the behavior control the time between two executions of the audio processing module can vary slightly. Processing all audio data since the last cycle would result in a slightly varying amount of recorded audio samples to be processed, since the microphones of NAO provide a constant stream of audio samples with 48 kHz. To be independent of the exact execution frequency of the audio processing, we select the block of audio sample to process using a moving window. Every cycle we use the most recent 2,048 audio samples. The time between two executions of the whistle detection will be approximately 33 ms (30 fps), thus a window length of 42.67 ms on the audio data (2,048 samples at 48 kHz) is a sufficient size to not skip any samples. When multiple microphones are available, this process is done for each channel independently, such that we obtain new microphone measurements in the form of equally sized blocks of audio samples. The audio

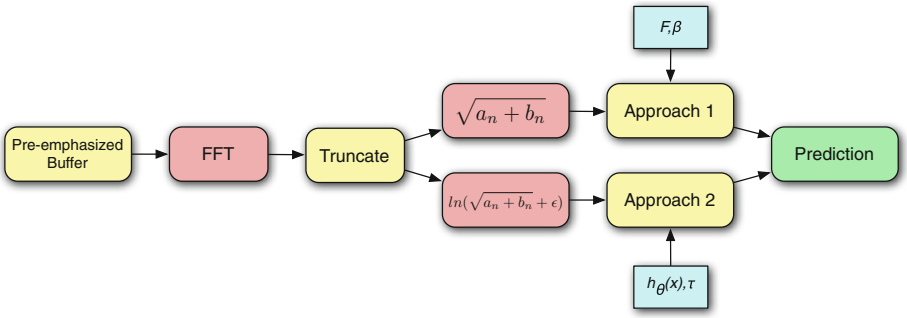


Fig. 1. The whistle identification framework for Sect. 3.

signal in the time domain can then be transformed to the frequency domain by using a Fast Fourier Transformation (FFT) on those blocks of 2,048 audio samples (Fig. 1).

If the size of the input to the FFT is N, then the output contains $\frac{N}{2}+1$ coefficients [7]. We have used these coefficients to generate the energy or log-energy profiles for each block of audio samples. These energy or log-energy profiles will be the input data for the whistle detection approaches. In the following, we will call a set of those coefficients a sample (as in sample input data or training sample, not audio sample). In our case, the output of the FFT is 1,025 coefficients. The preliminary analysis has shown that the majority of the energies or log-energies resides within the first 400 frequency components. Therefore, our samples contain feature vectors with 400 components, such that each feature consists of $\sqrt{a_n^2 + b_n^2}$ or $\ln(\sqrt{a_n^2 + b_n^2} + \epsilon)$, where $n = \{1, \ldots, 400\}$, a_n represents the real coefficients, b_n represents imaginary coefficients, and $\epsilon = 2.2204e^{-16}$ is a small positive number. We would also add a bias term to provide more expressivity to our learning models. We have collected positive and negative samples, and have annotated the target of each sample indicating the presence of the whistle. It is to be noted that we have collected our samples at the rate the system outputs the coefficients, which would amount to approximately 40–50 ms. For datasets containing multiple channels, for each sampling point, we have collected multiple samples proportional to the number of channels. We have tested two approaches, a simple approach using a frequency/band-pass filter to isolate the wanted frequency from the audio signal and another approach using logistic regression with l^2-norm regularization.

3.1 Frequency/Band-Pass Filter

In frequency/band-pass filter approach, we investigate the recognition of a whistle given the energies of the frequency spectrum. The fundamental frequency is usually the dominant frequency in the sample. For this reason, an attempt was made to exploit this correlation to provide a fast, memory efficient algorithm for whistle recognition. The algorithm takes as input a sample $\mathbf{x}$ (the energy profile), the known frequency of the whistle F, and a frequency error parameter β.

We iterate over the elements in the sample and record the index of the element with the highest amplitude. The index of the maximum element is translated to a frequency value by multiplying by the sample rate to frame ratio, where the sample rate is the number of samples taken per second in the original audio signal and the number of frames is the number of time-domain samples used to compute the FFT. If the computed frequency is within the bounds defined by $F \pm \beta$, the sample is assumed to have been taken in the presence of a whistle.

The frequency F may be selected by analyzing several positive data samples and computing the average fundamental frequency across these samples. β may be selected by trial and error, although there are fundamental limits to its potential values; since the frequency granularity of the output of the FFT is the $\frac{S}{f}$, where S is the sample rate and f is the number of frames used to compute the FFT, β cannot be chosen to be less than half of $\frac{S}{f}$, as this will prevent any recognition at all. In practice, it is desirable for β to be much larger than $\frac{S}{f}$; as β increases, the recall of the set should increase to 1.0, while the precision may decrease due to the inclusion of an increased number of false-positives. The value of β should also not be chosen to be high either, as while this will ensure excellent recall, it will include far too many false positives to be a useful recognition system. This algorithm may be improved by averaging the calculations of the fundamental frequency across multiple channels of input before testing the frequency's inclusion in $F \pm \beta$.

3.2 Logistic Regression with l^2-norm Regularization

Our datasets contain log-energy profiles as well as indications of the availability of the whistle. Therefore, we can formulate our original goal mentioned in Sect. 1 as a binary classification problem using logistic regression [2]. The outcome or the target of the methods such as logistic regression is quite suitable for robotic hardware, as it consumes minimal computational and memory resources. We represent our training examples by the set $\{(\mathbf{x}_i, y_i)\}_{i=1}^{M}$, where, the feature vector $\mathbf{x}_i \in \mathbb{R}^{N+1}$ with bias term, $y_i \in \{0, 1\}$, $M \gg N$, and $M, N \in \mathbb{Z}_{>0}$. Hence, we define the design matrix $\mathbf{X}$ to be a $M \times (N + 1)$ matrix that contains training samples in its rows. We also define a target vector $\mathbf{y} \in \mathbb{R}^M$ that contains all the binary target values from the training set. Our hypotheses space consist of vector-to-scalar sigmoid functions, $h_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{1}{1+e^{-\boldsymbol{\theta}^{\mathrm{T}}\mathbf{x}}}$, with adjustable weights $\boldsymbol{\theta} \in \mathbb{R}^{N+1}$. Similarly, we define the matrix-to-vector function, $\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{X})$, which results in a column vector with i^{th} element $h_{\boldsymbol{\theta}}(\mathbf{X}_i)$, where, $\mathbf{X}_i$ is the i^{th} row of the design matrix $\mathbf{X}$. We use a cross-entropy cost function, $J(\boldsymbol{\theta}) = -\frac{1}{M}(\mathbf{y}^{\mathrm{T}} \ln(\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{X})) + (\mathbf{1}-\mathbf{y})^{\mathrm{T}} \ln(\mathbf{1}-\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{X}))) + \frac{\lambda}{2M}\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\theta}$, with l^2-norm regularization. Here, the natural logarithmic function, $\ln(.)$, is applied element wise and $\mathbf{1}$ is an M-dimensional column vector with all elements equal to one. It is a common practice to avoid regularizing of the bias parameter. We have regularized the bias weight in the cost function, in order to present the equations without too much clutter. In practice, we normally do not regularize the weight associated with bias term. Taking the gradient of the cost function with respect to $\boldsymbol{\theta}$, we obtain $\nabla_{\boldsymbol{\theta}} J = \frac{1}{M}\mathbf{X}^{\mathrm{T}}(\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{X}) - \mathbf{y}) + \frac{\lambda}{M}\boldsymbol{\theta}$.

Table 1. Dataset description for different environments, and robot activities. Table shows the number of positive and negative samples collected for each environment and robot combinations, and the number of channels active on the robot.

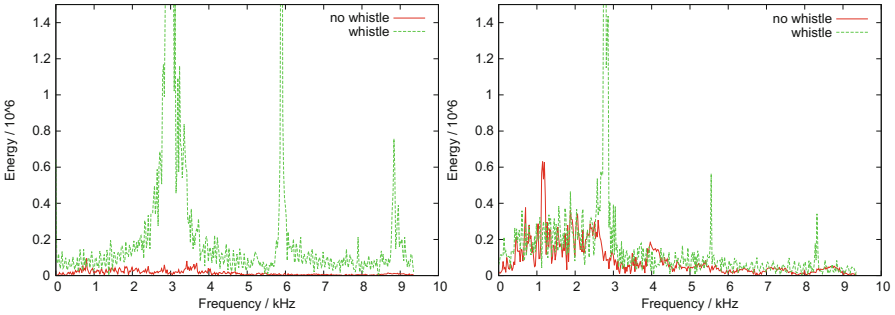
Set	Description	Positive	Negative	Channels
1	Indoor silent environment; silent robot	1005	1000	1
2	Indoor silent environment; active robot	2328	2032	1 – 2
3	Outdoor mildly noisy environment; silent robot	2112	2154	1 – 2
4	Outdoor noisy environment; silent robot	2030	2010	1 – 2
5	Indoor noisy environment (1); silent robot	4022	4170	1 – 2
6	Indoor noisy environment (2); silent robot	8024	8000	1 – 2
7	Indoor noisy environment (3); silent robot	8324	7996	1 – 4
8	Indoor noisy environment (3); active robot	8272	8000	1 – 4

We have trained our logistic regression classifiers in batch mode with the state of art L-BFGS quasi-Newton method [10] to find the best $\boldsymbol{\theta}$. We predict the availability of the whistle if and only if $h_{\boldsymbol{\theta}}(\mathbf{x}) \geq \tau$, where, $0 < \tau \leq 1$. Therefore, λ and τ would be the hyper-parameters that we need to modify to find the best solution. We have used standard parameter sweeping techniques to find the λ that provides the best trade off between the bias and the variance, while precision, recall, and F_1-score have been used to obtain the suitable τ value. As a preprocessing step, the features, except the bias, have been subjected to feature standardization. We have independently set each dimension of the sample to have zero-mean and unit-variance. We achieved this by first computing the mean of each dimension across the dataset and subtracting this from each dimension. Then each dimension is divided by its standard deviation.

4 Experiments and Results

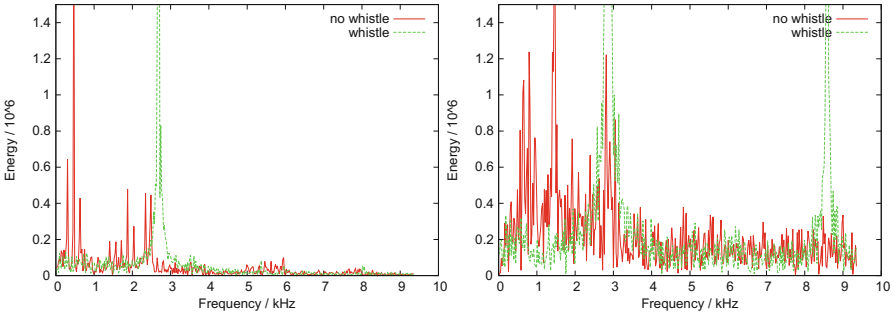
We have conducted all experiments using audio data recorded on NAO robots. We have used several different setups to evaluate the performance of the different approaches on a range of recorded data with different characteristics and different amounts of noise. Each recorded sample contains the log-energy profile of a the captured audio signal of one time step. During the recording, the samples were manually marked as positive, $y = 1$, or negative, $y = 0$, samples.

The whistle identification methods, that we will be describing in this paper, have used the datasets shown in Table 1 and Fig. 2. The samples in the datasets 1, 2, 5, 6, and 7 were collected from indoor environments, while the samples in the datasets 3, and 4 were collected from outdoor environments. The datasets 5, 6, and 7 contain samples from noisy RoboCup environments simulated though speakers. We have simulated three different noisy environments with a combinations of silent and active robots to collect samples. The datasets 2–6 have used channels 1 and 2 to collect samples, the datasets 7 and 8 have used all four channels, and the first dataset have used only the first channel.



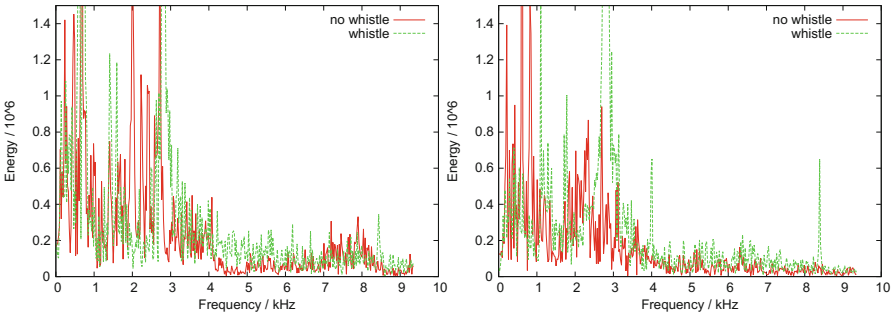
(a) Indoor, silent env., silent robot

(b) Indoor, silent env., active robot



(c) Outdoor, midly noisy env., silent robot

(d) Outdoor, noisy env., silent robot



(e) Indoor, noisy env., silent robot

(f) Indoor, noisy env., active robot

Fig. 2. Example frequencies for the different setups. Each figure shows one example for a positive sample (green) and one example for a negative sample (red) (Color figure online).

4.1 Frequency/Band-Pass Filter

We have analyzed the data using the maximum frequency technique, and for each dataset, we found best values for β such that the F_1-score was maximized. For each tuning of β and for each dataset, a random 70 % of the data was chosen as a training set, while the remaining 30 % served as a cross-validation set.

Table 2. Positive percentage, negative percentage, accuracy, precision, recall, F_1-score, and β for all datasets with all samples independently.

Dataset	Positive %	Negative %	Accuracy	Precision	Recall	F_1	β
1	100.00	99.66	99.83	1.00	1.00	1.00	271
2	99.86	99.18	99.54	0.99	1.00	0.99	154
3	100.00	97.82	98.90	0.98	1.00	0.99	130
4	99.84	99.67	99.75	1.00	1.00	1.00	247
5	89.28	98.10	93.82	0.98	0.89	0.93	457
6	94.46	98.32	96.38	0.98	0.94	0.96	226
7	84.49	98.18	91.12	0.98	0.84	0.91	154
8	93.72	98.85	96.19	0.99	0.94	0.96	247
1-8	92.00	97.89	94.96	0.98	0.92	0.95	319

Table 2 shows the performance on all datasets on the samples independently; each channel is considered a separate sample as well as the results for all of the data as a single set. The values for β were selected by performing a parameter sweep from 50 to 800 in increments of 1 and choosing the value which maximizes the F_1-score.

4.2 Logistic Regression with l^2-norm Regularization

We have conducted several analyses on our datasets to obtain the best outcome on the predictions. In all our experiments, we have used hold-out cross validation with 70 % data on the training set and 30 % data on the cross-validation set. In order to eliminate the bias, we have randomized the datasets before the split. We report here the results based on the minimum cost that have been observed on the cross-validation set after 30 independent runs, and the results are rounded up to two decimal points. In order to vary cost, we have used λ values from the set $\{0.01, 0.02, 0.04, 0.08, 0.16, 0.32, 0.64, 1.28, 2.56, 5.12, 10.24, 20.48, 40.96, 81.92, 163.84, 327.68\}$. Table 3 shows the results for positive percentage, negative percentage, accuracy, precision, recall, F_1-score, and τ for all datasets taking all

Table 3. Positive percentage, negative percentage, accuracy, precision, recall, F_1-score, and τ for all datasets with all samples independently.

Dataset	Positive %	Negative %	Accuracy	Precision	Recall	F_1	τ
1	100.00	100.00	100.00	1.00	1.00	1.00	0.5
2	99.86	100.00	99.92	1.00	0.99	0.99	0.5
3	99.53	100.00	99.77	1.00	0.99	0.99	0.5
4	99.51	99.67	99.59	0.99	0.99	0.99	0.4
5	94.70	94.97	94.84	0.95	0.95	0.95	0.5
6	97.88	98.38	98.13	0.98	0.98	0.98	0.5
7	96.68	97.08	96.88	0.97	0.97	0.97	0.5
8	95.57	97.92	96.72	0.98	0.96	0.97	0.7

Table 4. Positive percentage, negative percentage, accuracy, precision, recall, F_1-score, and τ for all datasets dependently (averaging).

Dataset	Positive %	Negative %	Accuracy	Precision	Recall	F_1	τ
2	100.00	100.00	100.00	1.00	1.00	1.00	0.5
3	100.00	100.00	100.00	1.00	1.00	1.00	0.5
4	99.67	100.00	99.84	1.00	0.99	0.92	0.5
5	96.69	95.85	96.26	0.96	0.97	0.96	0.5
6	98.67	98.83	98.75	0.99	0.99	0.99	0.5
7	98.56	98.50	98.53	0.99	0.99	0.99	0.5
8	96.30	99.33	97.79	0.99	0.96	0.98	0.6

Table 5. Overall performance on the combined dataset. The datasets 2–8 have 1–2 channels in common. The combined dataset have been tested on 400 + 1 features independently and averaging. We have performed analysis on combining the adjacent two channels to generate 800 + 1 features and tested the performance independently. Finally, we have analyzed the performance independently and dependently on all channels for datasets 2–8 on 400 + 1 features.

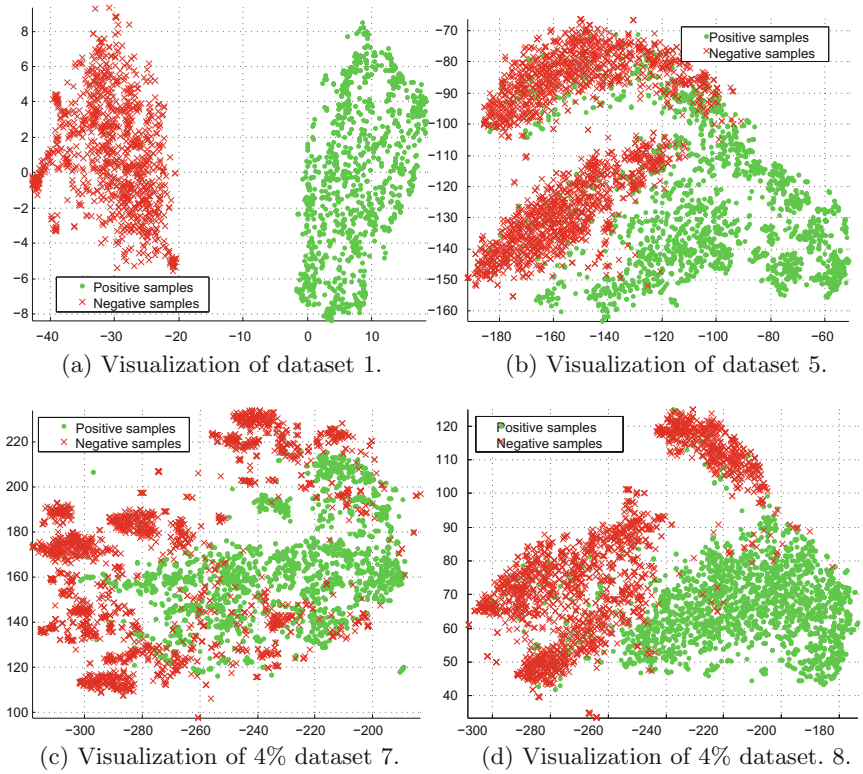
Dataset	Ch.	Method	Feat.	Pos.	Neg.	Acc.	Prec.	Recall	F_1	τ
2–8	1–2	Independently	400 + 1	94.26	96.78	95.51	0.97	0.94	0.96	0.5
2–8	**1–2**	**Dependently**	**400 + 1**	**96.47**	**97.60**	**97.03**	**0.98**	**0.97**	**0.97**	**0.5**
2–8	1–2	Independently	800 + 1	95.68	97.95	96.80	0.98	0.96	0.97	0.6
2–8	1–4	Independently	400 + 1	94.58	96.90	95.73	0.97	0.95	0.96	0.5
2–8	**1–4**	**Dependently**	**400 + 1**	**96.57**	**98.21**	**97.38**	**0.98**	**0.97**	**0.97**	**0.5**

samples independently, i.e., we have assumed that the samples from each channel is independent in the cross-validation set. Therefore, a sample is predicted positive if and only if $h_{\boldsymbol{\theta}}(\mathbf{x}) \geq \tau$. We have conducted a parameter sweep for τ from the set $\{0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$ and selected the value with the highest F_1-score.

Table 4 shows the results for all datasets dependently, i.e., during cross validation, we select hyper-parameters based on the average values of the channels active while the samples were collected. For example, when we collect samples for dataset eight, every time we would collect a sample, there are four active channels. During the cross validation phase, in order to determine the number of correctly classified positive samples, we have summed-up the probabilities of the samples of the adjacent four channels above the given threshold and divided by four. Therefore, when there are k-channels ($k \in \{1, 2, 3, 4\}$) active in a dataset, at every sampling point, we would collect k samples. Therefore, when we calculate the scores in Table 4 for cross-validation set, we have used the averaging formula (hence, dependently), $f(\mathbf{x}_1, \ldots, \mathbf{x}_k) = \frac{1}{k} \sum_{i=1}^{k} h_{\boldsymbol{\theta}}(\mathbf{x}_i) \geq \tau$, where $\{\mathbf{x}\}_1^k$ are features of the adjacent samples, and $f(\mathbf{x}_1, \ldots, \mathbf{x}_k) : \{\mathbb{R}^{N+1}\}_1^k \mapsto [0, 1]$, to predict a positive sample. When $k = 1$, independent and dependent scores will be similar. It is clearly evident from the Table 4 that the averaging has improved the prediction capabilities. We have not used the first dataset in Table 4 as it contains samples only from channel 1.

Table 6. Performance of the channels 1, 2, 3, and 4 separately and independently for the combined datasets (1–8).

Channel	Positive	Negative	Accuracy	Precision	Recall	F_1	τ
1	91.31	91.09	91.20	0.91	0.91	0.91	0.6
2	91.78	86.38	89.10	0.87	0.91	0.89	0.6
3	94.94	94.33	94.64	0.95	0.95	0.95	0.5
4	95.34	96.75	96.03	0.97	0.95	0.96	0.7



(a) Visualization of dataset 1. (b) Visualization of dataset 5.

(c) Visualization of 4% dataset 7. (d) Visualization of 4% dataset. 8.

Fig. 3. Visualization of datasets 1, 5, 7 (4 %), and 8 (4 %) using t-SNE.

Table 5 shows the overall performance on the combined dataset. We have conducted several analyses on the combined dataset. Firstly, Table 1 shows that the channels 1–2 are common to all datasets. Therefore, we have extracted all samples from channels 1–2 and analyzed the performance on 400 + 1 features independently and dependently (averaging). Secondly, we have expanded the adjacent two channels to create a feature vector of 800 + 1 features and analyzed the performance independently. Finally, we have analyzed the performance independently and dependently for all channels from the combined datasets 2–8

on 400 + 1 features. Table 5 concludes that for both robots with only two active channels (1–2) and for robots with all active channels (1–4) it is best to use weights learned from averaging for 400 + 1 features. Finally, we have observed the performance of the channels 1, 2, 3, and 4 separately and independently for the combined datasets (1–8), which is given in Table 6.

We have concluded from our findings that performance on averaging provides best results for our datasets. Once we have decided the hyper-parameters, we have learned the weights from the complete datasets. On the robot, we have used the average history of 16 decision points to decide the availability of a whistle. We have used a threshold of 0.8 for these averaging, and the robot has detected a whistle 100 % on a separate test set.

5 Discussion

When working with audio signals, it is a common practice to use Mel-frequency cepstral coefficients (MFCCs) [3] as features. In our work, we have used a truncated power or log-power spectrum of the signal as features. The main reason behind this choice is motivated by (1) the shape of the distribution of the samples in the high-dimensional space; and (2) the detection of a whistle signal at every sampling point. If we were to change the problem to identify particular patterns of whistle signals, then MFCCs would have been our primary choice as the feature extractor. Figure 3 shows the distribution of the samples in 2D for datasets 1, 5, 7, and 8 using t-Distributed Stochastic Neighbor Embedding (t-SNE) [6]. The distribution of the samples in dataset 1 (Fig. 3a) is clearly linearly separable, therefore, we have obtained 100 % accuracy in the first row of Tables 2, 3, and 4. Figure 3b shows the distribution of the samples of the dataset 5. The approach 2 has found solutions with 94.84 % and 96.26 % (Tables 3 and 4) accuracies, but the frequency/band-pass filter approach has shown slightly inferior (Table 2 fifth row) performance. The main reason behind the drop of performance for this approach is that it uses the frequency of the highest magnitude. When we collected samples for the dataset 5, we had explicitly whistled with less strength. Therefore, the energies of the whistle signal may not have enough strength to overcome the energies of the ambient sounds. Our second approach has managed to learn a statistically significant classifier for dataset 5. Figure 3c and d show the distribution of 4 % (approximately 4000) of the samples in the datasets 7 and 8. These were the hardest datasets that we had collected. Tables 3 and 4 show that approach 2 has found better solutions than approach 1 (Table 2 last row). Both approaches are fast enough to be executed in real-time on the NAO (Intel Atom Z530 1.6 GHz). The audio capture and FFT takes 2.4 ms. The whistle detection using approach 1 adds 0.1 ms, approach 2 adds 0.27 ms. Overall, our findings conclude that approach 2 has outperformed approach 1, and is suitable for practical usage.

For approach 1, as a future work, we have considered attempts to learn the frequency profile of the noise in the signal. The method takes the ordering of the samples into account, and rather than computing the frequency of maximum amplitude, computes the frequency with the highest impulse; a characteristic

of most whistles is that they usually cause a large difference in a particular frequency in a short period of time. This method accomplishes this by computing a normalization vector $\mathbf{v}$ such that $\mathbf{v}_t \odot \mathbf{x}_t = \mathbf{1}$. The frequency impulse is then obtained by computing $\mathbf{v}_{t-1} \odot \mathbf{x}_t = \mathbf{w}$. The vector $\mathbf{v}$ is then adjusted such that $\mathbf{v}_t = \alpha \mathbf{v}_{t-1} + (1 - \alpha)\frac{1}{\mathbf{x}_{t-1}}$, where α is a resistance factor that determines how easily $\mathbf{v}$ conforms to the new environment. We can then determine if frequencies within the range $F \pm \beta$ have experienced a sufficient impulse between $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$.

6 Conclusion

We have presented two approaches to identify an existence of a whistle sound on a NAO robot in various noisy environments using one, two, and four microphone channels. The first approach is based on a frequency/band-pass filter, whereas the second approach is based on logistic regression. The results show that the robot will be able to identify the whistle reliability even in very noisy environments. Even though both approaches are real-time compatible on predictions, the second approach has outperformed the first approach in all datasets and combined datasets and it is the most suitable method for practical usage. In future, we are planning to conduct classification using a multi-layer perceptron and support vector machines [2], and to extend our work to recognize different whistle patterns. We also plan to use the approach to improve robot localization.

References

1. Athanasopoulos, G., Brouckxon, H., Verhelst, W.: Sound source localization for real-world humanoid robots. In: Proceedings of the SIP, vol. 12, pp. 131–136 (2012)
2. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York Inc., Secaucus (2006)
3. Davis, S., Mermelstein, P.: Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans. Acoust. Speech Signal Proc. **28**(4), 357–366 (1980)
4. Hu, J.S., Chan, C.Y., Wang, C.K., Lee, M.T., Kuo, C.Y.: Simultaneous localization of a mobile robot and multiple sound sources using a microphone array. Adv. Robot. **25**(1–2), 135–152 (2011)
5. Kruijff-Korbayová, I., Athanasopoulos, G., Beck, A., Cosi, P., Cuayáhuitl, H., Dekens, T., Enescu, V., Hiolle, A., Kiefer, B., Sahli, H., et al.: An event-based conversational system for the NAO robot. In: Proceedings of the Paralinguistic Information and its Integration in Spoken Dialogue Systems Workshop, pp. 125–132. Springer (2011)
6. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(11), 2579–2605 (2008)
7. Mitra, S.: Digital Signal Processing: A Computer-based Approach. McGraw-Hill Companies, New York (2010)
8. Mokhov, S.A., Sinclair, S., Clement, I., Nicolacopoulos, D.: Modular Audio Recognition Framework and its Applications. The MARF Research and Development Group, Montréal, Québec, Canada, v. 0.3.0.6 edn., December 2007

9. Nakadai, K., Takahashi, T., Okuno, H.G., Nakajima, H., Hasegawa, Y., Tsujino, H.: Design and implementation of robot audition system 'hark'–open source software for listening to three simultaneous speakers. Adv. Robot. **24**(5–6), 739–761 (2010)
10. Nocedal, J., Wright, S.: Numerical Optimization. Springer Series in Operations Research and Financial Engineering, 2nd edn. Springer, New York (2006)
11. Okuno, H.G., Nakadai, K., Kim, H.-D.: Robot audition: missing feature theory approach and active audition. In: Pradalier, C., Siegwart, R., Hirzinger, G. (eds.) Robotics Research. STAR, vol. 70, pp. 227–244. Springer, Heidelberg (2011)
12. Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N., Fujimura, K.: The intelligent asimo: system overview and integration. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3, pp. 2478–2483. IEEE (2002)
13. Saxena, A., Ng, A.Y.: Learning sound location from a single microphone. In: IEEE International Conference on Robotics and Automation, ICRA 2009, pp. 1737–1742. IEEE (2009)
14. Sun, H., Yang, P., Liu, Z., Zu, L., Xu, Q.: Microphone array based auditory localization for rescue robot. In: 2011 Chinese Control and Decision Conference (CCDC), pp. 606–609. IEEE (2011)
15. Wrede, S., Klotz, D., Sheikhi, S., Jayagopi, D.B., Khalidov, V., Wrede, B., Odobez, J.M., Wienke, J., Nguyen, L.S., Gatica-Perez, D.: The vernissage corpus: a conversational human-robot-interaction dataset. In: Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction. No. EPFL-CONF-192462 (2013)