



Norges miljø- og
biovitenskapelige
universitet

DAT390 term paper

Detection and Classification of Acoustic Events using Machine Learning

Jon Nordby, December 2018

Abstract

A review of available literature on Acoustic Event Detection and Acoustic Event Classification was performed, and the theory of the important concepts for practical applications was summarized.

A baseline method was implemented using RandomForest on max-summarized mel-spectrograms, and tested on Bird Audio Classification using the DCASE2018 Task 3 dataset. At 77% AUC ROC on the testset in the matched case, it falls short of state-of-the-art, likely due to the limited feature extraction method that was chosen.

Contents

Introduction	3
Theory	4
Background	4
Digital sound	4
Time domain	5
Frequency domain	5
Time-Frequency domain	5
Problem formulations	6
Classification	6
Event Detection	6
Weak labeling	7
Feature representations	8
Frames	8
Low-level features	9
Spectrograms	9
Mel-spectrogram	10
MFCC	10
Summarizing features	11
Convolution	11
Feature learning	12
Learning on raw audio waveform	12
Other feature representations	12
Case study	14
Bird vocalization detection	14
Feature representation	14
Method	14
Results	15
Discussion	16
Conclusion	17
References	18

List of Figures

1	From acoustical sound to digital and back. Source: (Processing.org, n.d.) . .	4
2	Waveform (time domain) of a frog croaking repeatedly. Overall changes in signal intensity over time is visible, but what pitches are present (if any) is not.	5
3	Spectrum (frequency domain) of same frog sample. The frequency content can be seen, but temporal patterns are gone	5

4	STFT spectrogram of same frog sample. Both temporal pattern and frequency patterns can be seen.	6
5	Classification task with one class label per audio sample. Source: (Virtanen, Plumbley, and Ellis 2017, fig 6.1)	6
6	Event detection, with labels at precise start and end locations in time. Source: (Virtanen, Plumbley, and Ellis 2017, fig 6.3)	7
7	Computing frames from an audio signal, using windows functions. Based on image by (Driedger, n.d.)	8
8	Low-level features of birdsong. There are distinct peaks in the signals, but how well they correspond to birdsong events is not immediately clear.	9
9	Mel-spaced filterbank. Filters are set to to be unity-height. Mel-filters using unit-area filters also exist. Source: (Fayek 2016)	10
10	Mel-spectrogram of birdsong. The birdsong is clearly visible as up and down chirps at 3kHz and higher	10
11	MFCC of birdsong	11
12	Clip summarizations of frames. Source: (Bello, n.d., 27)	11
13	Convolution kernel as edge detector, applied to image. Source: (Dettmers 2015)	12
14	Results of hyperparameter search	15
15	Learning curve of chosen model	16

Introduction

Sound, and machine learning on sound, is a wide topic. Some areas have enough interest to be considered their own fields. This especially includes machine understanding of human speech (*Speech Recognition*) and the computational analysis of music (*Music Information Retrieval*).

This paper instead focuses on detection and classification of *Acoustic Events*. Such events can be any kind of sound, and in contrast to speech and music, are often not produced with an intent to communicate. Many of the techniques utilized for Acoustic Events can also be used within speech and music, and many methods were in fact adopted from these fields.

While not as popular as computer vision or natural language processing, the application of machine learning to acoustic events can be found across many fields of scientific study and in many industries.

(Bhattacharya et al. 2017) designed a portable unit for assisted with heart auscultation, a common acoustic procedure to screen for heart problems. In (Dixon et al. 2018) acoustic emission was used to create sensors for early warning of landslides. (al. 2018) explains how detection of acoustic events can assist security camera operators in spotting situations of interest. In (Sanders and J. Mennill 2014) bird migration patterns are studied using acoustic monitoring. In domestic animal care, (Chelotti et al. 2016) shows an acoustic method for monitoring cattle feeding activity. Acoustic events are used in industry for non-destructive quality control, predictive maintenance of machinery and in structural analysis of buildings.

In most application scenarios, analysis is done on audible sound transmitted via the air. However there are fluid-borne sound (hydrophony) and structure-borne sounds and vibrations, and sounds are below the audible frequency range for humans (infrasound) and some above (ultrasound). The techniques presented here should apply in general across these modalities.

While a basic understanding of the human auditory system can be beneficial when developing machine hearing, Psychoacoustics will not be covered here. Those interested in a detailed treatise on this subject can read (Lyon 2017).

The paper assumes some prerequisite knowledge on machine learning, in particular difference between supervised and unsupervised learning; common task formulations like Classification, Clustering and Regression; and common methods like Random Forests and Convolutional Neural Networks. A good primer is (Raschka 2015).

Basic familiarity with Digital Signal Processing is assumed. In particular knowledge of the Fourier Transform, digital filters and convolutions.

When it comes to sound, a brief background is provided. We will then cover the main theoretical aspects of Machine Learning for Acoustic Event Detection and Classification, and then use the task of detecting the presence of birds as a practical example.

In summary, the goal is that after reading this report: a machine learning practitioner, without prior knowledge about sound and digital audio processing, is be able to solve basic Acoustic Event Detection problems.

Theory

Background

Digital sound

Physically, sound is a variation in pressure over time. For machine learning, it must exist in a digital representation. The acoustic data is converted to analog electric signals by a microphone and then digitized using an Analog-to-Digital-Converter (ADC).

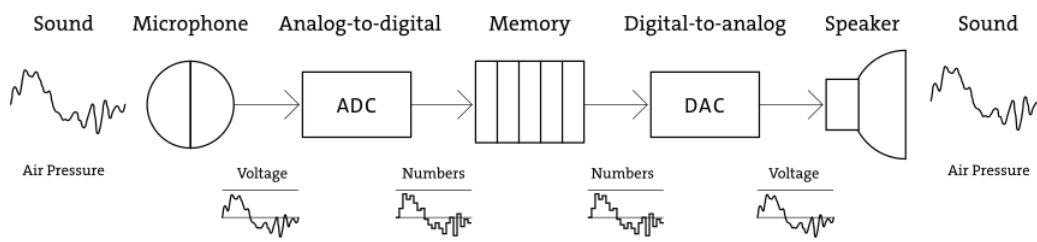


Figure 1: From acoustical sound to digital and back. Source: (Processing.org, n.d.)

In the digitization process, the signal is quantized in time at a certain sampling frequency, and the amplitude quantized at a certain bitdepth. A typical sampling frequency is 44100 Hz, and bitdepth 16 bit. With these parameters, the acoustic sound can be reconstructed without perceivable differences by humans.

Digital sound can be stored uncompressed (PCM .wav), using lossless compression (.flac) or using lossy compression (.mp3). Lossy compression removes information and may add artifacts, and is best avoided for machine learning tasks. Recordings can be multi-channel but for acoustic events single-channel (mono) data is still the most common.

Time domain

In the time domain an audio signal is seen as a continuously varying signal corresponding to the pressure variation of the acoustic signal.

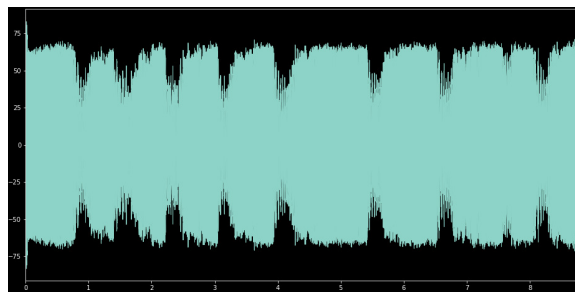


Figure 2: Waveform (time domain) of a frog croaking repeatedly. Overall changes in signal intensity over time is visible, but what pitches are present (if any) is not.

Frequency domain

Using the Fourier Transform, the audio can be losslessly converted to the frequency domain.

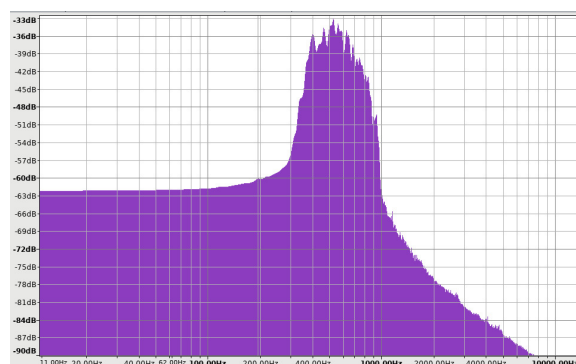


Figure 3: Spectrum (frequency domain) of same frog sample. The frequency content can be seen, but temporal patterns are gone

Time-Frequency domain

Using Short-Time-Fourier-Transform (STFT), the audio can be mapped to the Time-Frequency domain.

The STFT has a tradeoff in time/frequency resolution. Multi-resolution STFT can be used if high resolution in both time and frequency is needed.

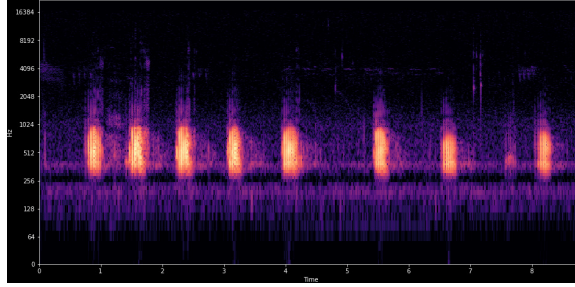


Figure 4: STFT spectrogram of same frog sample. Both temporal pattern and frequency patterns can be seen.

Problem formulations

Classification

In classification the goal is to determine the which acoustic event appears in an audio sample. Samples can be short relative to the acoustic event length, or long, possibly having many instances of an event. The number of events and their time is not returned in classification.

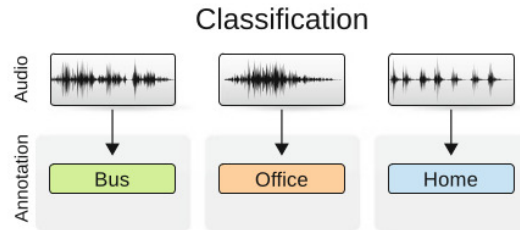


Figure 5: Classification task with one class label per audio sample. Source: (Virtanen, Plumbley, and Ellis 2017, fig 6.1)

The classification can be binary (is this birdsong?), or multi-class (which species is this?). Classification is often limited to a single label.

When the classification is done on long samples consisting of many different events it is called Acoustic Scene Classification. Examples of a ‘scene’ in urban environments could be ‘restaurant’, ‘park’, ‘office’, each having a different composition of typical acoustic events.

Event Detection

In event detection (also known as onset detection) the goal is to find the time spans where a given acoustic event occurs. If the acoustic event is “frog croaking”, then for each instance of a frog croak the start time and end time of this event should be marked.

In monophonic event detection, only the most prominent event is returned. A classifier ran on short samples relative to the length of the acoustic event can be used a detector.

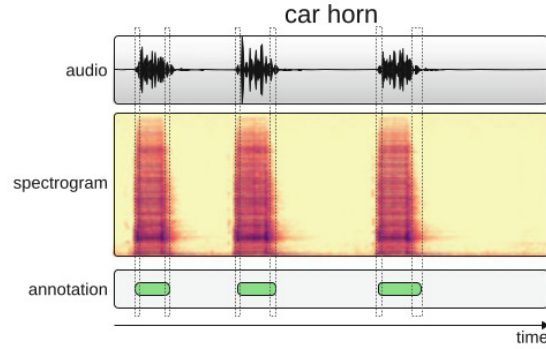


Figure 6: Event detection, with labels at precise start and end locations in time. Source: (Virtanen, Plumbley, and Ellis 2017, fig 6.3)

In polyphonic event detection, multiple events are allowed at the same time. This can be approached using separate classifiers per event type, or using a multi-label classifier as a joint model.

Weak labeling

Many acoustic events are short in duration, for instance a door slamming. Other acoustic events only happen intermittently, like the the frog vocalizations in previous example.

Under supervised learning, ideally each and every of the acoustic events instances in the training data would be labeled with their start and end time. This is called ‘strong labels’. However acquiring strong labels requires careful attention to detail by the annotator and is costly.

Therefore, often per-event time-based labels are not available. Instead fixed-length audio clips are only marked whether an event occurs at least once, or not at all. This is called a ‘weak label’ or ‘weakly annotated’ data.

When performing event detection, with small time windows (compared to annotated audio clips) as input, the missing time information means there is not a direct label for this input. This is known as a Multiple Instance Learning (MIL) problem. Under MIL input instances are grouped into a ‘bag’, and the label exists on the bag instead of the individual instances. MIL formulations exist for many common machine learning algorithms.

Feature representations

In a typical machine learning pipeline, raw data is transformed into features for the machine learning model via feature extraction. The typical feature extraction process and feature representation is discussed here.

Frames

To process and analyze the audio data it is often represented as *frames*, small groups of samples across time. Frames can be produced in real-time, by collecting N samples at a time, or by splitting up audio files after they have been recorded.

In a frame based approach, a frame is the smallest unit of time processed by the machine learning algorithm. Therefore the frame length must be set long enough to contain enough relevant information, but not so long that temporal variations disappear. For speech, a typical choice of frame length is 25ms. Similar frame lengths are often adopted for acoustic events, unless there are specific concerns.

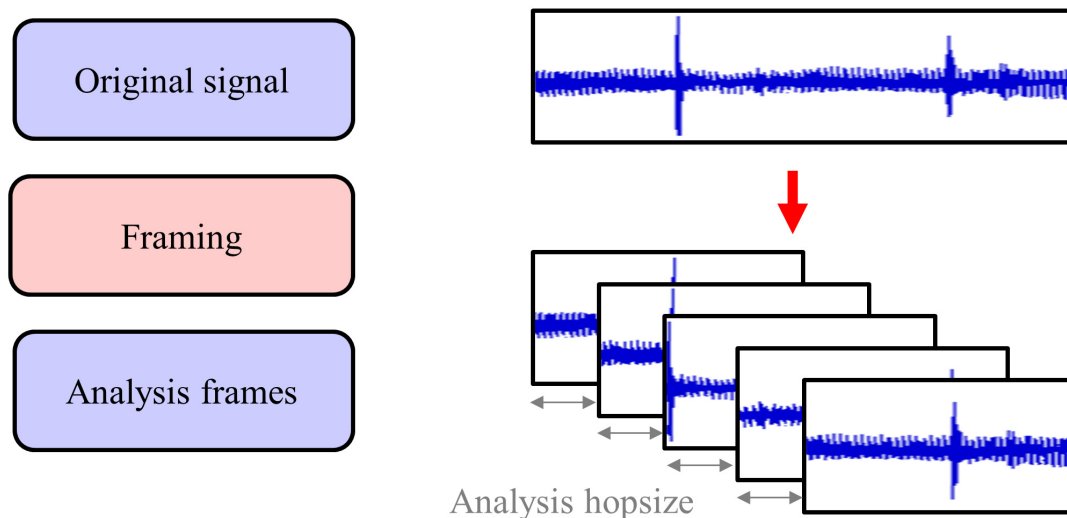


Figure 7: Computing frames from an audio signal, using windows functions. Based on image by (Driedger, n.d.)

Frames often use overlapping samples at the start and end. Without overlap, acoustic event that happens partially in one frame results in different signals than when appearing in the middle. The overlap can be specified as percentage of the frame length (overlap percentage), or as a number of samples (hop length). Overlap can for instance be 50%. A window function is applied to ensure that the signal level stays constant also in overlapping sections.

Low-level features

From the samples in each frame, features can be calculated.

(Breebaart 2004, Ch 2.1.1) defined a Low-Level feature set for audio classification as:

1. root-mean-square (RMS) level
2. spectral centroid
3. spectral bandwidth
4. zero-crossing rate
5. spectral roll-off frequency,
6. band energy ratio,
7. delta spectrum magnitude,
8. pitch and (9) pitch strength

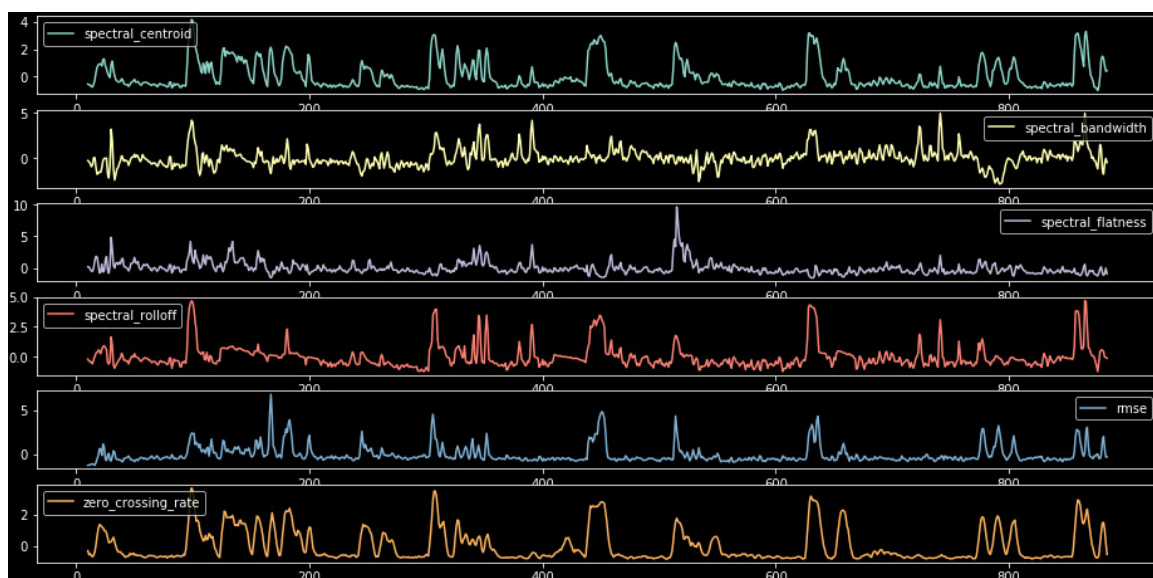


Figure 8: Low-level features of birdsong. There are distinct peaks in the signals, but how well they correspond to birdsong events is not immediately clear.

Of these, root-mean-square (RMS) and zero-crossing rate (ZCR) are calculated directly on the samples in the time-domain.

The spectral features are computed using the Fourier Transform on a frame, and then summarizing with standard statistics. See (Li 2001) for the definitions.

Low-level spectral features, being univariate statistics on the spectrogram, removes a lot of the information present.

Spectrograms

A raw Short Time Fourier Transform can contain 1024 or more bins, often with strong correlation across multiple bins. To reduce dimensionality, the STFT spectrogram is often

processed with a filter-bank of 40-128 frequency bands. Some filter-bank alternatives are 1/3 octave bands, the Bark scale, Constant-Q transform and the Mel scale. All these have filters spacing that increase with frequency, mimicking the human auditory system.

Mel-spectrogram

A spectrogram processed with triangular filters evenly spaced on the Mel scale is called a Mel-spectrogram.

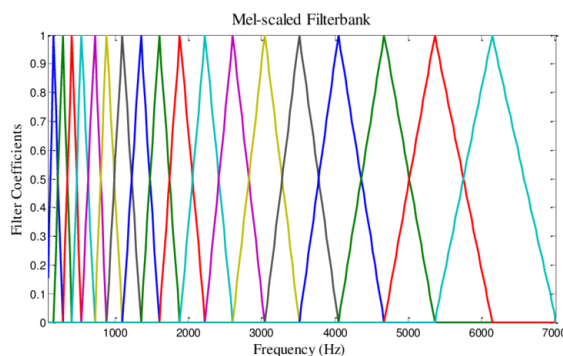


Figure 9: Mel-spaced filterbank. Filters are set to to be unity-height. Mel-filters using unit-area filters also exist. Source: (Fayek 2016)

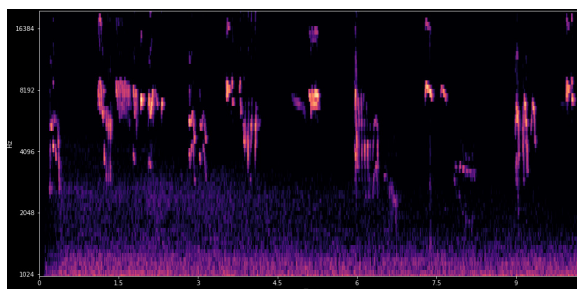


Figure 10: Mel-spectrogram of birdsong. The birdsong is clearly visible as up and down chirps at 3kHz and higher

A mel-spectrogram can still have significant correlation between bands.

MFCC

To reduce correlation one can compute the Discrete Cosine Transform (DCT-2) on the mel-spectrogram. The result is known as the Mel Filter Cepstrum Coefficients (MFCC).

Another advantage of the DCT is that important information in the signal tends to end up in the lower coefficients. Therefore it can be compressed efficiently (but lossily) by dropping the higher coefficients. 13-20 coefficients is common.

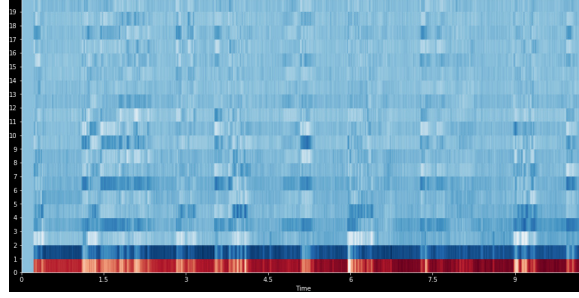


Figure 11: MFCC of birdsong

The band in the cepstrum are the coefficient of cosines that when summed together reconstruct the input (using the inverse DCT). This means that events that are close in frequency domain are no longer next to each other in a MFCC matrix, making local pattern-matching algorithms less effective. Therefore they are not so popular for use with Convolutional Neural Networks, but mostly used with linear models such as Gaussian Mixture Models (GMM) and Hidden Markov Models (HMM).

Summarizing features

When classifying longer segments of labeled audio it can be desirable to come up with a summary that represents the entire clip. This can be done by computing statistics across the per-frame features for all frames in the clip.

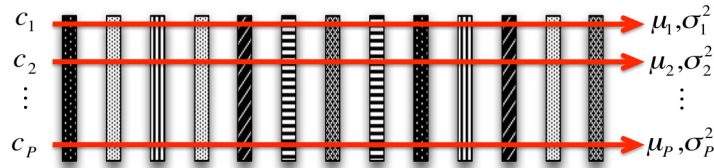


Figure 12: Clip summarizations of frames. Source: (Bello, n.d., 27)

Any statistical aggregate functions can be used, such as **min**, **max**, **median**, **mean**, **skew**, **Kurtosis**, etc. In a summary the order of frames is ignored, and we have a Bag-of-Frames feature model.

Convolution

A convolution filter (also called kernel) allows to express many common transformations on 1d or 2d data, like edge detection (horizontal/vertical) or smoothing filters (median). But kernels can be seen as parametric local feature detector can express more complex problem-specific patterns, like a upward or downward diagonal from a bird chirp when applied to a spectrogram. Using a set of kernels in combination can detect many pattern variations.

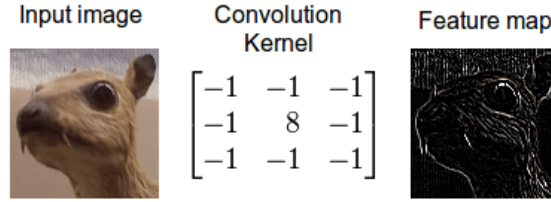


Figure 13: Convolution kernel as edge detector, applied to image. Source: (Dettmers 2015)

Feature learning

To avoid having to manually design or choose convolution kernels, they can be learned from data. One approach is to randomly select small N, M sized patches from the spectrogram and to reduce this into a representative codebook using vector quantization. In (Stowell and Plumbley 2014), this was done for detection of birdsong using *spherical k-means* clustering, a variation of k-means using the cosine (dis)similarity as the distance metric. In (Bisot et al. 2017), Sparse Non-negative Matrix Factorization was used for Acoustic Scene Classification.

Other unsupervised methods include and Convolutional AutoEncoders (CAE). In (Masci et al. 2011), CAEs were stacked to obtain a ‘deep’ model of hierarchical features.

If large amount of labeled data are available, a Convolutional Neural Network (CNN) can be used for supervised learning of deep network of convolution operators. CNN-based models using log mel-spectrogram features are the best performing models in many audio classification problems, such as those in (“DCASE2018 Challenge Results,” n.d.).

Learning on raw audio waveform

Deep neural networks can also use the raw audio waveform as input features, skipping the spectrogram computation step. This learning a model without dedicated feature extraction, is sometimes called ‘end 2 end’ learning.

(Dai et al. 2016) was able to train a Convolutional Neural Network on raw waveform data to match log-mel spectrogram features in an audio classification task. The network also needed to jointly learn bandpass filters in the first layers to replace the transformation normally performed by the spectrogram.

Such models can also be used in fusion with spectrogram model, as in (Lederle and Wilhelm 2018).

Other feature representations

A large amount of alternative feature representations have been proposed. Some have explored wavelet filterbanks and the Scattering transform (Salamon and Bello 2015).

Prior to efficient convolutional neural networks, image-based features like Scale Invariant Feature Transform (SIFT) and Histogram of Oriented Gradients (HOG) have also been applied to spectrograms (Behun and Hradis 2018), (Muroi et al. 2009).

Feature extraction models that more closely mimics the human hearing system include Cascade of Asymmetric Resonators with Fast-Acting Compression (CARFAC) (Lyon 2011).

Case study

To apply the above theory, we look at the Bird Audio Detection task from the DCASE2018 machine-learning challenge.

Some code will be provided inline for illustration, while the full code is available at <https://github.com/jonnor/birddetect> as executable Jupyter Notebooks.

Bird vocalization detection

The problem formulation is binary classification: the system should output whether audio sample has bird(s) or not.

- 10 second audio clips
- One label for the clip (weakly annotated)
- 3 training sets, 3 test sets
- Total 48'000 samples

The 2018 edition of this challenge is focused on generalizable performance. Therefore 2 of the test sets are recorded at different locations with different equipment than the training samples (mismatched conditions).

Feature representation

As a basis for feature representations, the mel-spectrogram is used. The mel-spectrogram is calculated with 64 mel-filter bands, from 500Hz to 15000 Hz. Exploration of the dataset showed that many recordings had a lot of static white-noise below approximately 1000Hz, and not all recordings had data above approximately 16KHz. The implicit assumption is that most or all of the bird vocalization is inside the preserved range.

Method

A baseline method of RandomForest on max-summarized mel-spectrogram was implemented first.

An attempt was also made to use transfer learning of a CNN model trained on ImageNet, however this failed to give good results. Even a trivial LogisticRegression on top of the max or mean pooled output failed to progress higher than 60% accuracy. Problem occurred with or without data augmentation. Fine tuning the network also did not improve the results. A bug was discovered in the pre-processing of the spectrograms used by this model, which may have been the root cause.

The RandomForest model with 100 trees was tested on a dataset of 3000 files, 1000 for each of the 3 labeled datasets, sampled randomly. AUC ROC was used as the score metric. 33% of the data was left out as a testing set.

The code is available as a Jupyter notebook at <https://github.com/jonnor/birddetect/blob/0.1.0/Baseline.ipynb>

Results

A grid-search with 5 cross-validation was performed on the training set to determine appropriate hyperparameters. The `min_samples_leaf` parameter was used to limit the depth of the RandomForest trees. A range of 1% to 20% of trainingset size was tested.

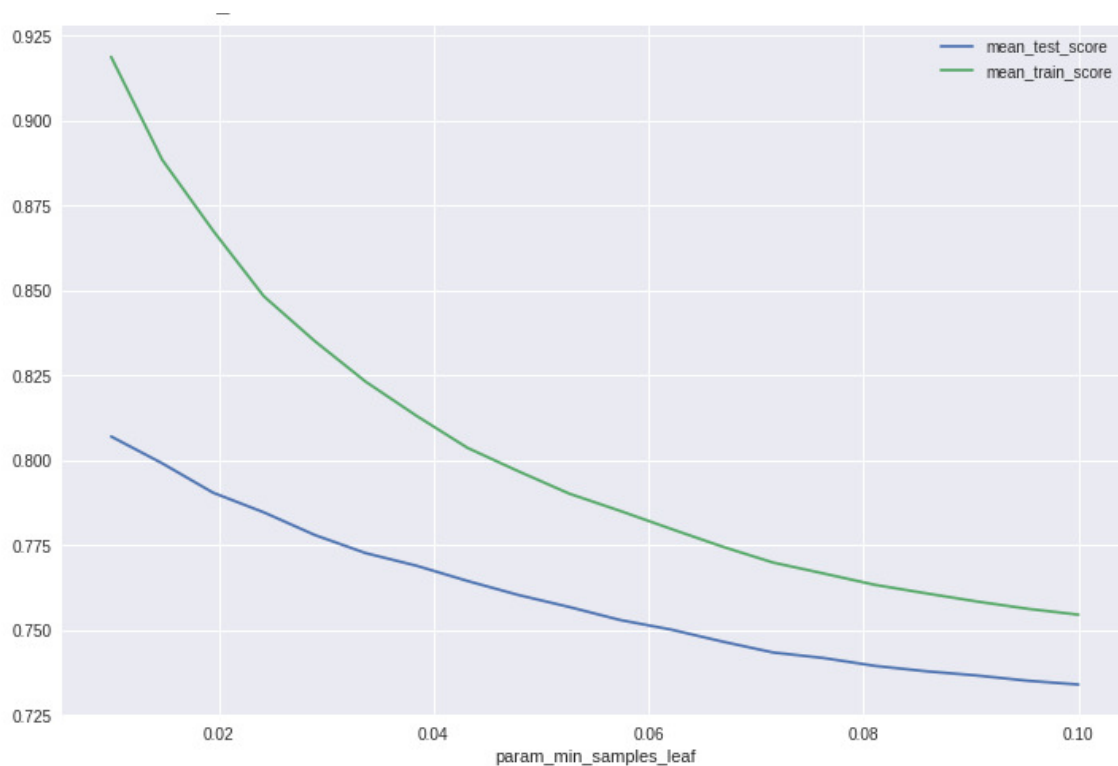


Figure 14: Results of hyperparameter search

Accepting some variance, a `min_samples_leaf` of 0.04 was chosen.

A learning curve was computed to evaluate whether the model was limited by the dataset size.

When computing the AUC ROC score on the left-out testing set, it scored 76.9%.

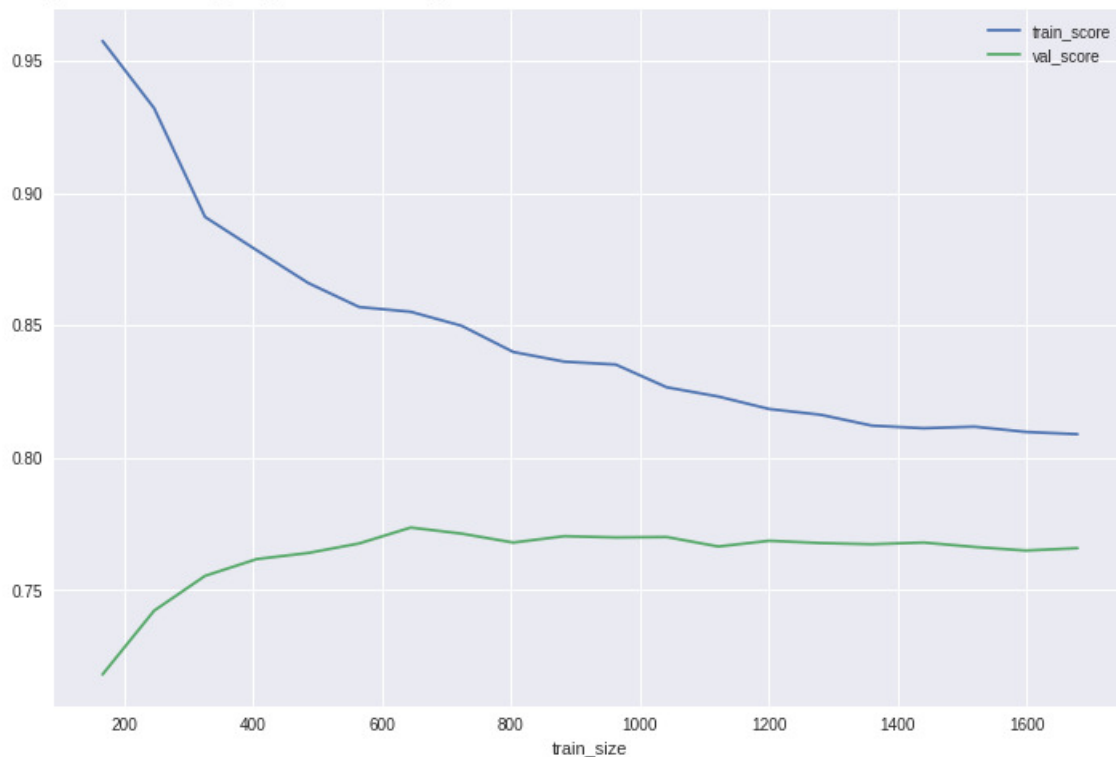


Figure 15: Learning curve of chosen model

Discussion

By default sklearn RandomForest has no depth limit. With the lowest tested limit, `min_sample_leaf=0.01`, the model performs 93% on training and 81% on test, suggesting serious overfitting. So hyperparameter tuning was critical to avoid overfitting.

The score of 76.9% on the testing set matches the 76–77% in validation in (from gridsearch and learning curve). This suggests that with the tuned hyperparameters, the final model is not overfitting significantly.

The learning curve is more or less flat from 1400 to 1700 (full set), indicating that the model would not improve much with more data, at least with these hyperparameters.

The feature representation is a statistical summarization of the entire 10 second clip, which is likely to be a limiting factor. (“Results Bird Audio Detection Task Dcase2018,” n.d.) showed that for in the *mismatched* case, which is significantly more challenging, Convolutional Neural Networks are able to reach AUC ROC of 89%, albeit with the full 48k dataset and much data augmentation.

Conclusion

RandomForest on max-summarized mel-spectrograms was a quick model to develop, train and execute, but has performance that falls short of state-the-art methods. Richer feature extraction is likely needed in order to get better results, and we hope to explore this in the future.

References

- al., Supriya. A. M et. 2018. *International Journal of Electronics, Electrical and Computational System* 7 (4). <https://ijltemas.in/DigitalLibrary/Vol.7Issue3/169-173.pdf>.
- Behun, Kamil, and Michal Hradis. 2018. “Image Features in Music Style Recognition,” December.
- Bello, Juan Pablo. n.d. “Low-Level Features and Timbre: MPATE-Ge 2623 Music Information Retrieval Class.” New York University. http://www.nyu.edu/classes/bello/MIR_files/timbre.pdf.
- Bhattacharya, A., M. Mishra, A. Singh, and M. K. Dutta. 2017. “Machine Learning Based Portable Device for Detection of Cardiac Abnormality.” In *2017 International Conference on Emerging Trends in Computing and Communication Technologies (Icetcct)*, 1–4.
- Bisot, V., R. Serizel, S. Essid, and G. Richard. 2017. “Feature Learning with Matrix Factorization Applied to Acoustic Scene Classification.” *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25 (6): 1216–29. <https://doi.org/10.1109/TASLP.2017.2690570>.
- Breebaart, Jeroen. 2004. “Features for Audio Classification.” <http://www.jeroenbreebaart.com/papers/soia/soia2004.pdf>.
- Chelotti, José O., Sebastián R. Vanrell, Diego H. Milone, Santiago A. Utsumi, Julio R. Galli, H. Leonardo Rufiner, and Leonardo L. Giovanini. 2016. “A Real-Time Algorithm for Acoustic Monitoring of Ingestive Behavior of Grazing Cattle.” *Computers and Electronics in Agriculture* 127: 64–75. <https://doi.org/https://doi.org/10.1016/j.compag.2016.05.015>.
- Dai, Wei, Chia Dai, Shuhui Qu, Juncheng Li, and Samarjit Das. 2016. “Very Deep Convolutional Neural Networks for Raw Waveforms,” October.
- “DCASE2018 Challenge Results.” n.d. <http://dcase.community/challenge2018/index>.
- Dettmers, Tim. 2015. “Understanding Convolution in Deep Learning.” 2015. <http://timdettmers.com/2015/03/26/convolution-deep-learning>.
- Dixon, N., A. Smith, J. A. Flint, R. Khanna, B. Clark, and M. Andjelkovic. 2018. “An Acoustic Emission Landslide Early Warning System for Communities in Low-Income and Middle-Income Countries.” *Landslides* 15 (8): 1631–44. <https://doi.org/10.1007/s10346-018-0977-1>.
- Driedger, Jonathan. n.d. <https://commons.wikimedia.org/w/index.php?curid=47678736>.
- Fayek, Haytham. 2016. “Speech Processing for Machine Learning: Filter Banks, Mel-Frequency Cepstral Coefficients (Mfccs) and What’s in-Between.” 2016. <https://lucidworks.com/2016/08/22/17382/>.
- Lederle, Marcel, and Benjamin Wilhelm. 2018. “Combining High-Level Features of Raw Audio Waves and Mel-Spectrograms for Audio Tagging.” In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (Dcase2018)*, 207–11.

- Li, Dongge. 2001. “Classification of General Audio Data for Content-Based Retrieval.” <https://www.sciencedirect.com/science/article/pii/S0167865500001197>.
- Lyon, Richard F. 2011. “Using a Cascade of Asymmetric Resonators with Fast-Acting Compression as a Cochlear Model for Machine-Hearing Applications.” In *Autumn Meeting of the Acoustical Society of Japan*, 509–12. <https://ai.google/research/pubs/pub37215>.
- . 2017. *Human and Machine Hearing: Extracting Meaning from Sound*.
- Masci, Jonathan, Ueli Meier, Dan Ciresan, and Jürgen Schmidhuber. 2011. “Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction.” In, 52–59. https://doi.org/10.1007/978-3-642-21735-7_7.
- Muroi, Takashi, Ryoichi Takashima, Tetsuya Takiguchi, and Yasuo Ariki. 2009. “Gradient-Based Acoustic Features for Speech Recognition.” In *ISPACS 2009 - 2009 International Symposium on Intelligent Signal Processing and Communication Systems, Proceedings*, 445–48. <https://doi.org/10.1109/ISPACS.2009.5383805>.
- Processing.org. n.d. <https://processing.org/tutorials/sound/>.
- Raschka, Sebastian. 2015. *Python Machine Learning*. Birmingham, UK: Packt Publishing.
- “Results Bird Audio Detection Task Dcase2018.” n.d. <http://dcase.community/challenge2018/task-bird-audio-detection-results>.
- Salamon, Justin, and Juan Bello. 2015. “Feature Learning with Deep Scattering for Urban Sound Analysis.” In *Proceeding 2015 European Signal Processing Conference (EUSIPCO)*. <https://doi.org/10.1109/EUSIPCO.2015.7362478>.
- Sanders, Claire, and J. MennillDaniel. 2014. “Acoustic Monitoring of Nocturnally Migrating Birds Accurately Assesses the Timing and Magnitude of Migration Through the Great Lakes.” *The Condor* 116 (August): 371–83. <https://doi.org/10.1650/CONDOR-13-098.1>.
- Stowell, Dan, and Mark Plumbley. 2014. “Automatic Large-Scale Classification of Bird Sounds Is Strongly Improved by Unsupervised Feature Learning.” *PeerJ* 2 (July): e488. <https://doi.org/10.7717/peerj.488>.
- Virtanen, Tuomas, Mark Plumbley, and Dan Ellis. 2017. *Computational Analysis of Sound Scenes and Events. Computational Analysis of Sound Scenes and Events*. <https://doi.org/10.1007/978-3-319-63450-0>.



Norges miljø- og biovitenskapelig universitet
Noregs miljø- og biovitenskapelige universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway