

THE NINTH ANNUAL MLSP COMPETITION: FIRST PLACE

Gábor Fodor

Budapest University of Technology and Economics
fodgabor@math.bme.hu

ABSTRACT

The goal of the 2013 MLSP Competition is to predict the set of bird species present in audio recordings, collected in field conditions. Real-world audio data presents special difficulties such as simultaneously vocalizing birds, other animal sounds, and background noise. Although the task can be considered as a multi-instance multi-label learning problem, I propose a Binary Relevance approach with Random Forest. The proposed solution achieves 0.956 AUC and ranks 1st place on the Kaggle private leaderboard.

Index Terms— spectrogram, template matching, random forest

1. INTRODUCTION

The full dataset consists of 645 ten-second audio recordings in uncompressed WAV format. There are 19 bird species in the dataset. Each recording is paired with a subset of species that are present in the recording. In addition to the raw audio, the competition organizers provide additional pre-computed data (spectrograms, filtered spectrograms, segmentation, segment features, and a histogram-of-segment features for each recording).

The method presented in this paper builds on the results of the previous Marinexplore and Cornell University Whale Detection Challenge, where template matching gave the best result. My implementation is available at https://github.com/gaborfodor/MLSP_2013.

2. FEATURE ENGINEERING

This section describes the feature extraction steps. Although pre-processed spectrograms are available, I created new spectrograms using similar parameters (window size 512, Hamming window, 75% overlap). Beside the absolute spectrograms, logarithmic scale representation is considered too.

While the lower frequencies can be extremely useful in whale sound classification, I removed them for this challenge; only the highest 200 pixels of the spectrograms are kept.

The organizers describe a method to find relevant segments (“syllables”) with supervised classification [1]. I pro-

pose a different, unsupervised segmentation method based on image processing techniques.

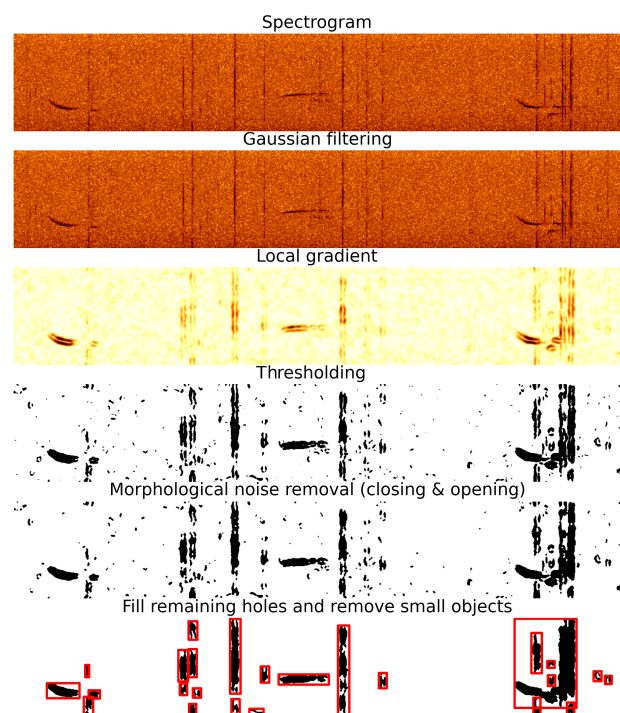


Fig. 1. Image Processing on a Hermit Thrush spectrogram.

To filter undesirable noise and separate different sounds, the following image processing steps are used (see Fig. 1):

- Gaussian filtering ($\sigma = 3$)
- Local gradient (maximum - minimum in 3 pixel radius)
- Thresholding (90%)
- Morphological noise removal (binary opening then closing)
- Fill the remaining holes and remove small segments (area less than 100 pixels)

The scikit-image image processing library is used for data preparation. After these steps, the remaining patterns (red boxes in Fig. 1) are stored. Many of the recordings in the training data are paired with multiple species labels. However, in our method, templates are collected only from the single-labeled 81 training recordings. These few recordings give 2959 patterns in the absolute-intensity spectrograms, and 3019 patterns in the logarithmic transformed spectrograms.

For every spectrogram, scikit-image’s template matching function is used to compute the similarity at the maximum value of the normalized cross-correlation map with each template. This gives a 5978 dimensional feature vector for each recording. Note that similar bird sound should appear in similar frequencies (but anywhere in the time domain) so we can look for matching templates in a smaller part of the spectrogram (5 pixel up or down shift is allowed). Despite this restriction, the template matching is the most time-consuming part of the whole process.

After extracting template matching features, the 100 dimensional histogram-of-segments features provided by the organizers, and the location code (mapped to an integer), are appended to the feature vector.

3. MODELING

Due to the small training size (322 recordings) and large dimension (6079 features) aggressive feature selection is used for each bird species. The feature selection is based on univariate linear regression tests. The number of selected features is between 30 and 70, and is set manually, with higher dimensions if the class is more balanced. For each species, there is a binary model using Random Forest Regressor [2]. For both the feature selection and modeling, the scikit-learn library is used [3].

My best uploaded submission is a simple average of 23 Random Forests. However the gain compared to my other selected submission (a single Random Forest) is not significant (+0.001 AUC). I compared several parameter settings for the Random Forest using cross-validation. The best parameters found are:

- 500 trees
- 4 features to consider when looking for the best split
- Minimum 2 samples required to split an internal node
- Mean squared error to measure the quality of a split

The Random Forest can be used to examine feature importance. Because the features come from our extracted templates, the variable importance tells us about which templates are most important for identifying each species. Fig. 2 shows the most important 31 features/templates to classify the Hermit Warbler.

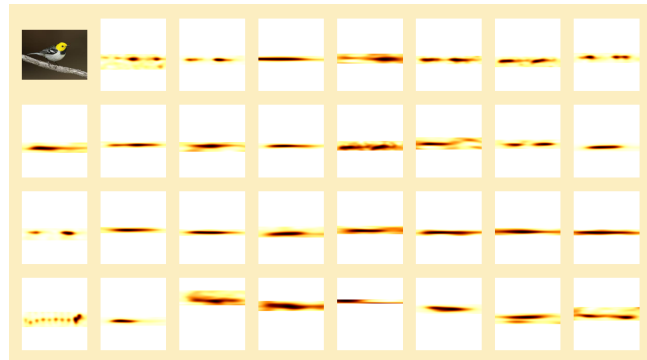


Fig. 2. Most important templates to catch a Hermit Warbler.

4. CONCLUSION

This paper describes the winning solution to the MLSP 2013 Bird Classification Competition. In the proposed method, robust features are extracted from the audio recording using template matching, then an ensemble of Random Forest classifiers is applied. Additionally, we present a simple way to visualize the most relevant patterns for detecting each species.

5. ACKNOWLEDGMENT

I would like to thank the organizers for collecting, labeling and preparing the dataset, and Forrest Briggs and Raviv Raich for their help in editing this writeup.

6. REFERENCES

- [1] F. Briggs, B. Lakshminarayanan, L. Neal, X. Z. Fern, R. Raich, S. J. K. Hadley, A. S. Hadley, and M. G. Betts. Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach. In *The Journal of the Acoustical Society of America*, 131:4640, 2012.
- [2] L. Breiman. Random forests. *Machine Learning*, 45(1):532, 2001.
- [3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825-2830, 2011.