

EECS 207

Bird Call Project Report Instructions and Format (ver. 2: April 29)

Due: Friday May 14 by 11:59pm.

How: As a **SINGLE** PDF file uploaded to CatCourses

Your report should contain three sections:

1. The results of your CBIR framework applied to the bird call dataset.
2. A discussion of your results including any unexpected observations.
3. Your code.

More details are now provided.

Section 1: Results

Assuming you are able to get your CBIR system to work and you are able to compute precision-recall curves and average precision, your results section should contain the following.

The quantitative results of performing retrievals with each of the 218 ROI spectrogram subimages:

- Consider each of the 218 subimages as a query.
- Perform a retrieval against the 218 image target dataset (leave the query image in the target dataset).
- For each set of ranked retrievals, compute the precision at recall values of 0, 0.1, 0.2, ..., 1.0. See my lecture notes on computing precision at specific recall values.
- Compute the average precision at these recall values over all 218 queries.
- Plot the average precision versus these recall values as shown below.
- Compute the overall average precision (a single value) by computing the average of the average precision over the recall values (11 values: 0, 0.1, 0.2, ..., 1.0).
- Report the overall average precision for different configurations in a table.

You should provide results from different configurations. At a **minimum**, you should present results for:

- Two different types of image features. Obvious candidates are mean/stddev features and Gabor texture features.
- Two different distance measures.

You can also explore other configurations:

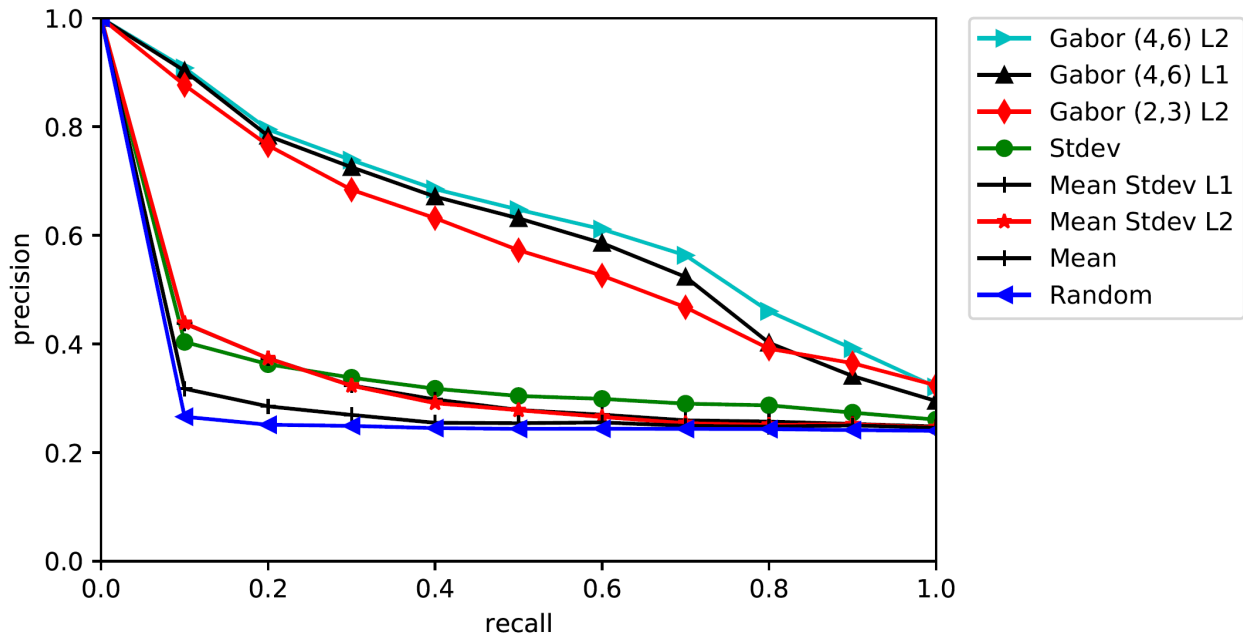
- Other image features and distance measures.
- Different types of feature normalizations.

- Different spectrogram computations (vary the window size, window overlap, nonlinear mapping of the values, etc.).
- Denoising of the spectrogram images and/or audio data.
- Anything else you can think of.

Your results section should contain:

- One or more average precision-call curves corresponding to the different CBIR configurations you tried. See the example below. These should be the average precision over 218 queries at recall values of 0, 0.1, ..., 1.0.
- A table with the average precision values.
- A short description of your different configurations. This can be in a table.

Example results section:



Configuration	Average Precision
Gabor (4,6) L2	0.6473
Gabor(4,6) L1	0.6237
Gabor (2,3) L2	0.6003
Stdev	0.3759
Mean Stdev L1	0.3634
Mean Stdev L2	0.3610
Mean	0.3298
Random	0.3151

Configuration	Features	Dist.	Norm.	Spectrograms		
				Win. Size	Win. Over.	Mapping
Gabor(4,6) L2	Gabor 4 scales, 6 orientations. Frequencies: (0.05, 0.1, 0.20, 0.4)	L2	None	1024	512	Log10
Gabor(4,6) L1	Gabor 4 scales, 6 orientations. Frequencies: (0.05, 0.1, 0.20, 0.4)	L1	None	1024	512	Log10
Gabor(2,3) L2	Gabor 2 scales, 3 orientations. Frequencies: (0.05, 0.25)	L2	None	1024	512	Log10
Stdev	Standard deviation of pixel values	L1	None	1024	512	Log10
Mean Stdev L1	Standard deviation and mean of pixel values	L1	None	1024	512	Log10
Mean Stdev L2	Standard deviation and mean of pixel values	L2	None	1024	512	Log10
Mean	Mean of pixel values	L1	None	1024	512	Log10
Random	N/A	N/A	N/A	N/A	N/A	N/A

Random means that the retrieval results were randomly ordered. This is, in some sense, the worst you can do.

Note, in the table above, the column labels indicate:

- Features: What features were used to perform the retrievals.
- Dist.: What distance measure was used to perform the retrievals.
- Norm.: Feature normalization if any.
- Win. Size: The window size when computing the spectrograms.
- Win. Over.: The window overlap when computing the spectrograms.
- Mapping: The nonlinear mapping of the spectrogram values.

Note, also, that you might need some text to describe aspects of your configurations. For example, if you performed feature normalization, you need to briefly state which kind. Or, if you performed denoising, how did you do it.

For your reference, I am copying here the MATLAB code I used to create the plot above:

```
import numpy as np
import matplotlib.pyplot as plt

# These are the reference recall values.
ref_recall = np.linspace(0,1.0,11)

# The following vectors are the precision values averaged over 218 queries at the
# reference recall values.
prec_mean_stdev_L2 = np.array([1., 0.43848573, 0.37336158, 0.32296442, 0.2908912,
                               0.27784999, 0.26553185, 0.25372586, 0.25071506,
                               0.2518069, 0.24625397])
prec_mean_stdev_L1 = np.array([1., 0.43766941, 0.37360812, 0.32361425, 0.29782827,
                               0.27843056, 0.26967821, 0.25932444, 0.25705279,
                               0.25268582, 0.24797991])
prec_mean = np.array([1., 0.31751072, 0.28509208, 0.26932032, 0.25484683,
                      0.25400531, 0.25547819, 0.24917323, 0.24766006,
                      0.24991321, 0.2455363])
prec_stdev = np.array([1., 0.40364162, 0.3626716, 0.33780847, 0.317542,
                      0.30420624, 0.29881685, 0.29001502, 0.28691787,
                      0.27353051, 0.26064867])
prec_Gabor_4_6_L2 = np.array([1., 0.90814209, 0.79487006, 0.73845427,
                              0.68512132, 0.64734454, 0.61128216, 0.56316666,
                              0.46007599, 0.39126469, 0.32067775])
prec_Gabor_4_6_L1 = np.array([1., 0.90293918, 0.78308272, 0.72537057,
                              0.67128851, 0.63133813, 0.58573478, 0.52338053,
                              0.40201358, 0.34095592, 0.2949108])
prec_Gabor_2_3_L2 = np.array([1., 0.87662676, 0.76559073, 0.68394105,
                              0.6316988, 0.57258549, 0.52587783, 0.46745005,
                              0.39124763, 0.36455554, 0.32375759])
prec_random = np.array([1., 0.26591098, 0.2511851, 0.24907526, 0.24497533,
                       0.24364291, 0.24396356, 0.24349314, 0.24314503,
                       0.24136042, 0.23994372])

fig, ax = plt.subplots()

# The following plots the PR curves with a legend. I plot them in order of
# decreasing performance (this makes it easier to interpret the legend).
ax.plot(ref_recall, prec_Gabor_4_6_L2 , '>c-', label='Gabor (4,6) L2')
ax.plot(ref_recall, prec_Gabor_4_6_L1 , '^k-', label='Gabor (4,6) L1')
ax.plot(ref_recall, prec_Gabor_2_3_L2 , 'dr-', label='Gabor (2,3) L2')
ax.plot(ref_recall, prec_stdev, 'og-', label='Stdev')
ax.plot(ref_recall, prec_mean_stdev_L1, '+k-', label='Mean Stdev L1')
ax.plot(ref_recall, prec_mean_stdev_L2, '*r-', label='Mean Stdev L2')
ax.plot(ref_recall, prec_mean, '+k-', label='Mean')
ax.plot(ref_recall, prec_random, '<b-', label='Random')
ax.set_xlim(0, 1)
ax.set_ylim(0, 1)
ax.set_xlabel('recall')
ax.set_ylabel('precision')
ax.legend(bbox_to_anchor=(1.04,1), borderaxespad=0)
fig.savefig('PR_Curves.pdf', bbox_inches = 'tight')
```

Section 2: Discussion

In this section, you should briefly discuss your results in terms of what configurations resulted in the best performance. Also note any unexpected results or anything else you would like to convey.

Section 3: Code

Append your code here. You can either copy-and-paste into your document or create separate PDF files for your report and code which you then combine into a single PDF file.

At a *minimum*, this should include your code for:

- Computing the precision and recall values.

You can include other code if you tried other things like denoising, etc.