TAMING HORSES IN SINGING VOICE DETECTION

Jan Schlüter

Vienna DL Meetup

December 2, 2019

JOHANNES KEPLER UNIVERSITY LINZ

Austrian Research Institute for Artificial Intelligence

OFAI

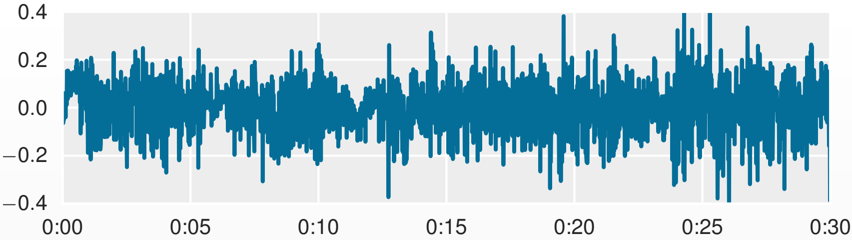
TAMING HORSES IN SINGING VOICE DETECTION

Jan Schlüter Vienna DL Meetup December 2, 2019

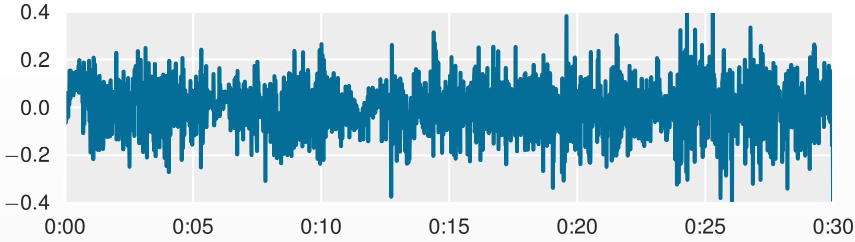
JOHANNES KEPLER UNIVERSITY LINZ

OFAI Austrian Research Institute for Artificial Intelligence

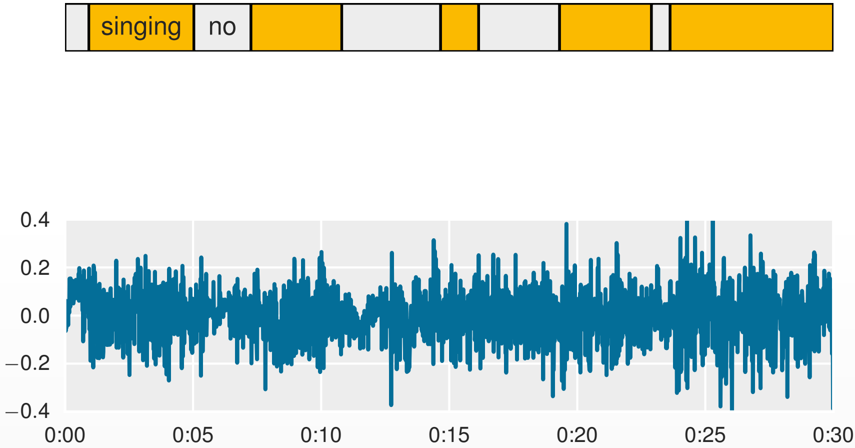
Task



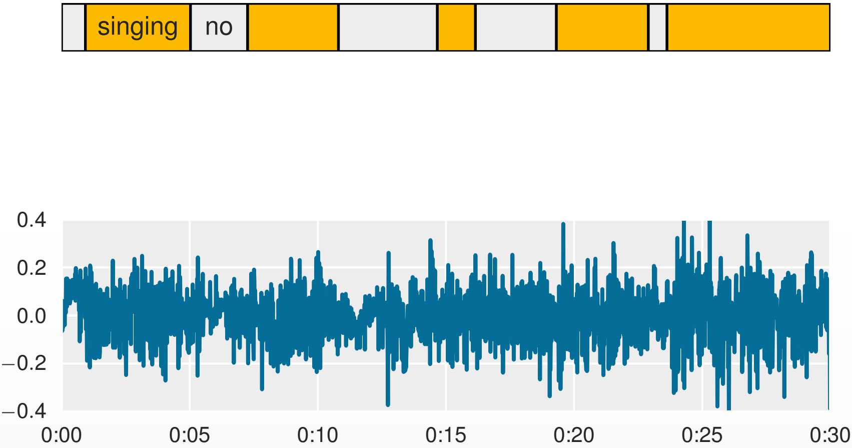
Task



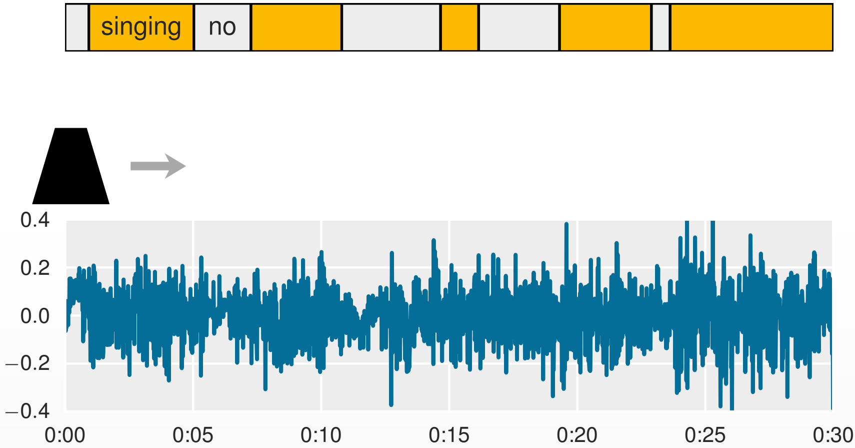
Task



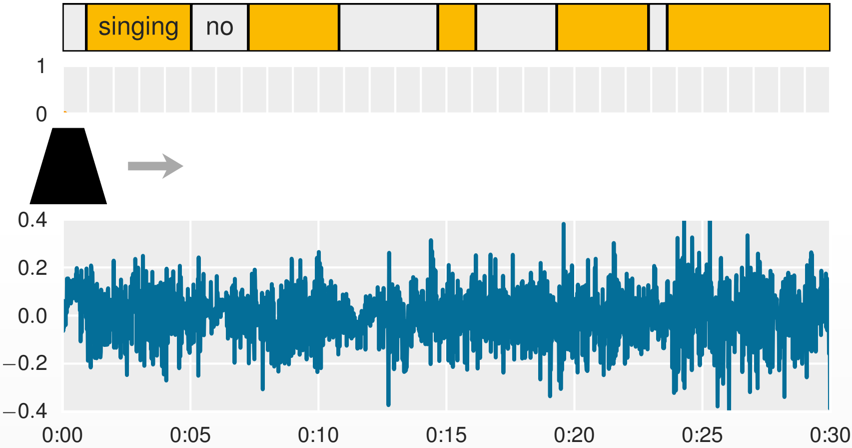
Generic Approach



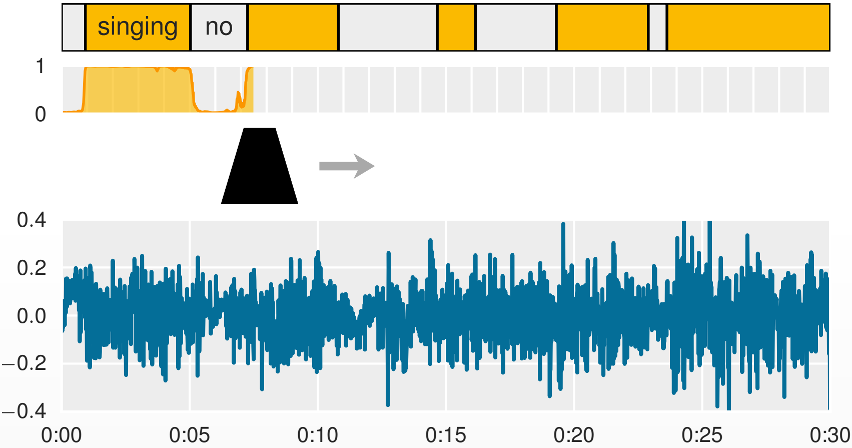
Generic Approach



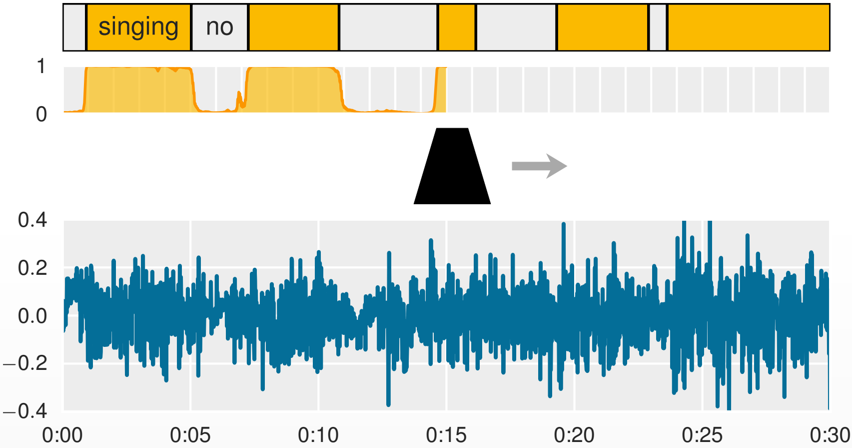
Generic Approach



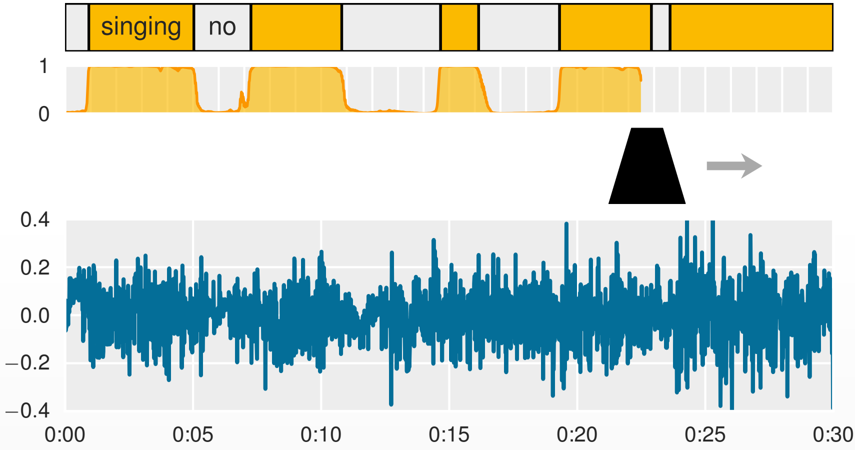
Generic Approach



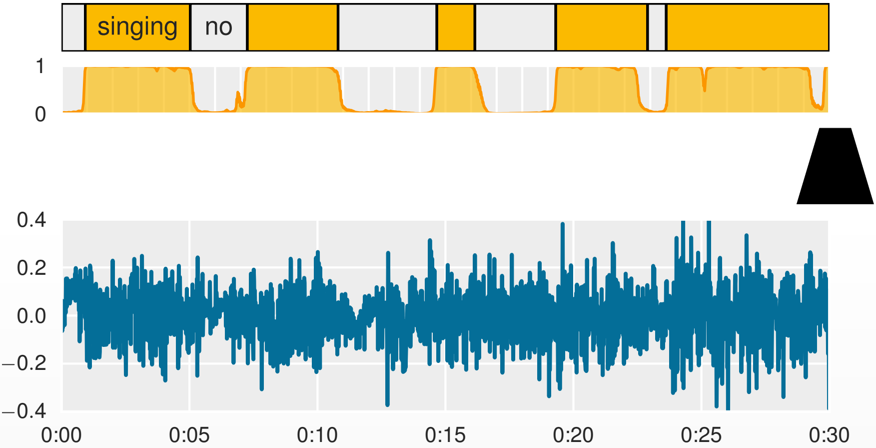
Generic Approach



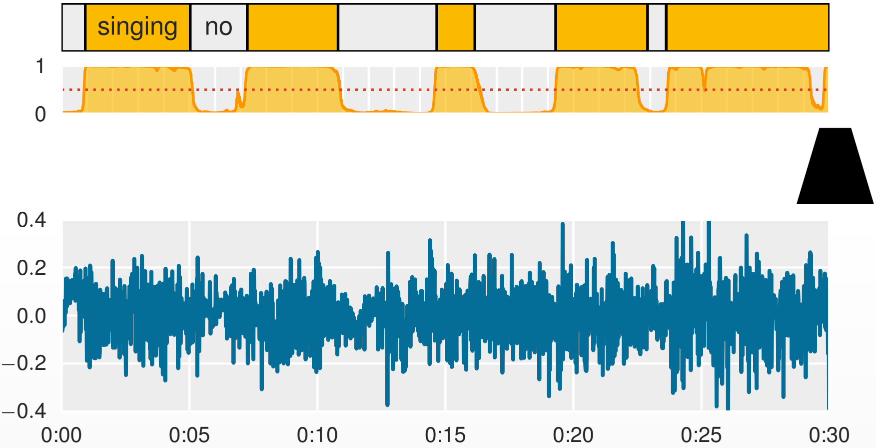
Generic Approach



Generic Approach

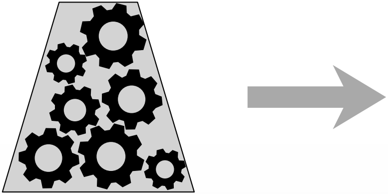


Generic Approach

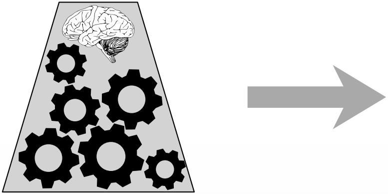


Design vs. Learning

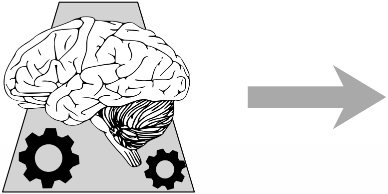
Design vs. Learning



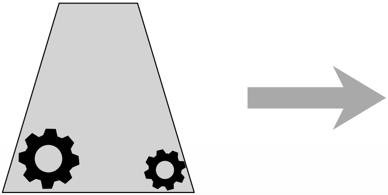
Design vs. Learning



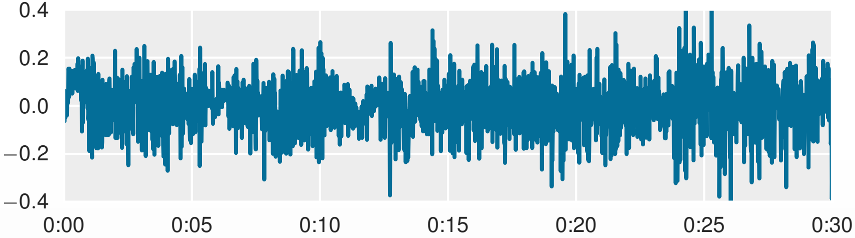
Design vs. Learning



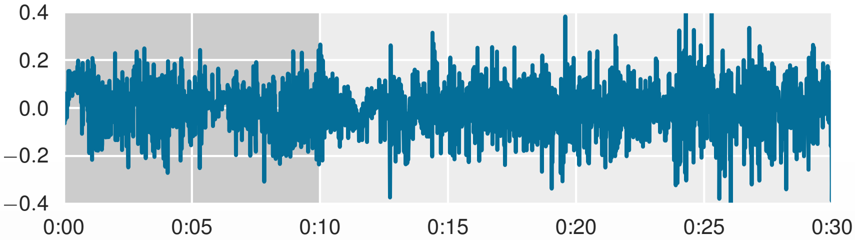
Design vs. Learning



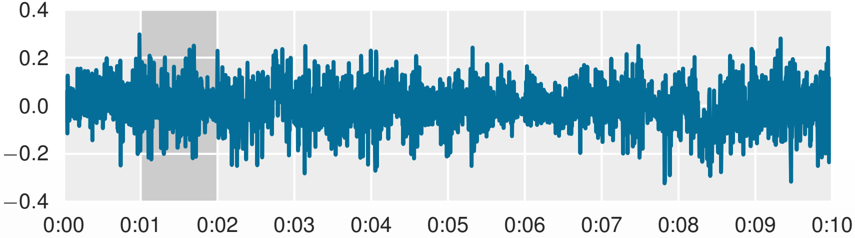
Music Signals



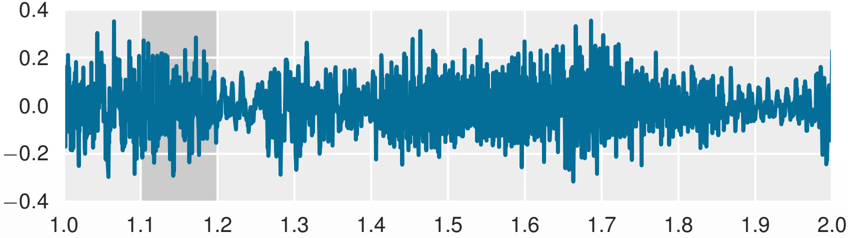
Music Signals



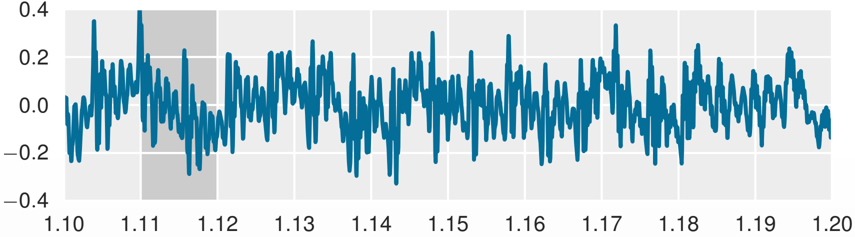
Music Signals



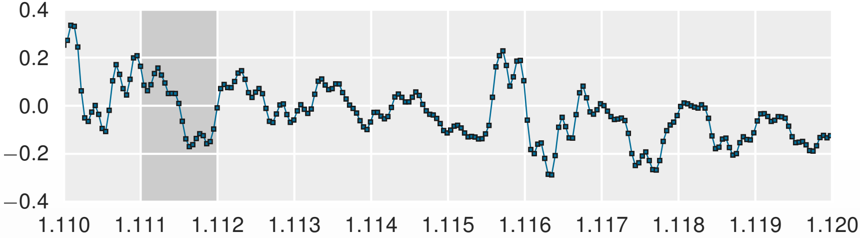
Music Signals



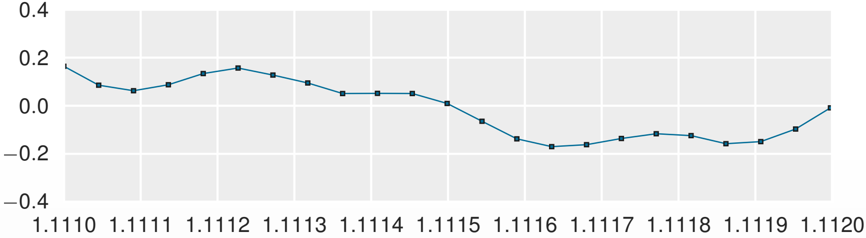
Music Signals



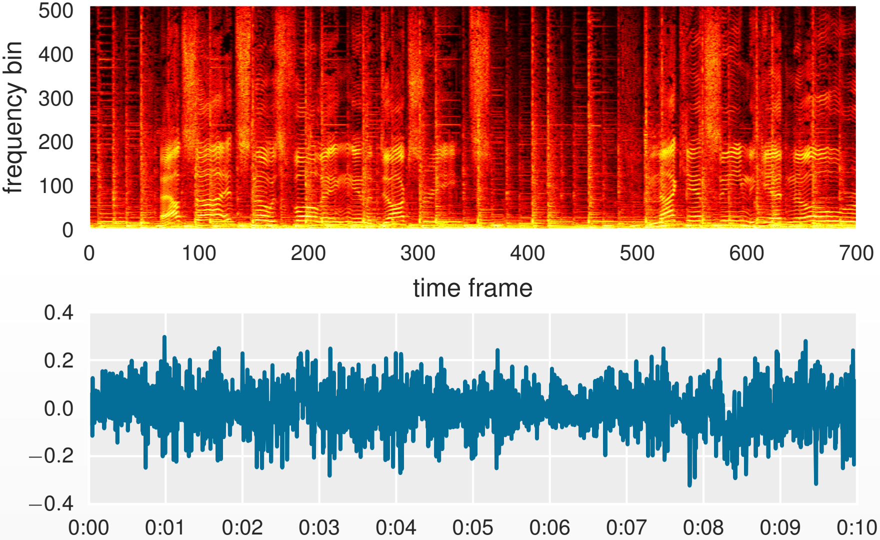
Music Signals



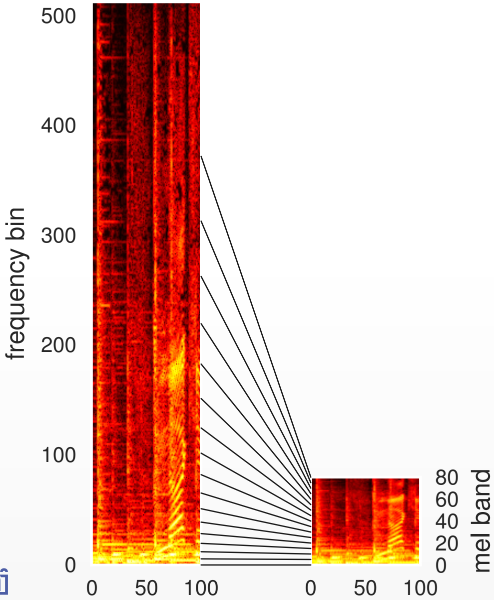
Music Signals



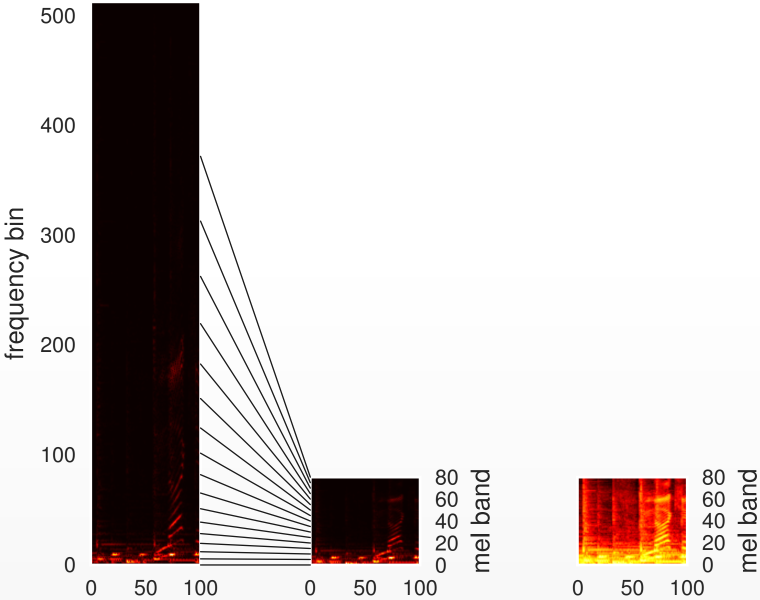
Spectrogram



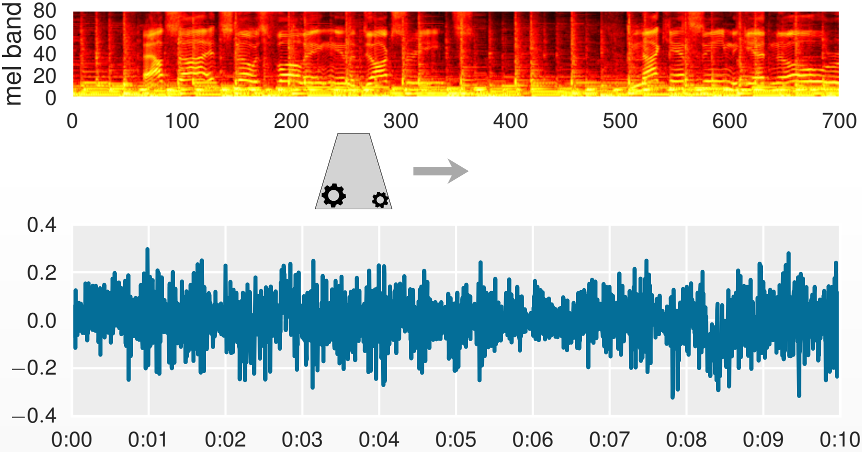
Frequency scale



Magnitude scale

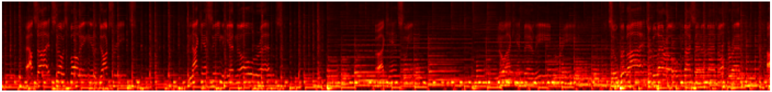


Designed steps

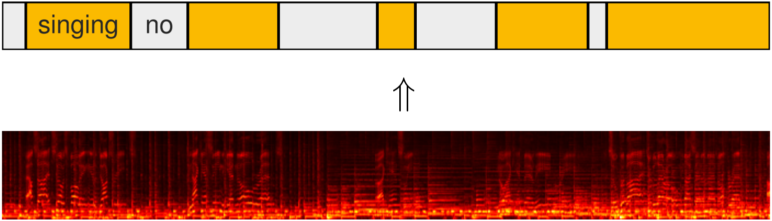


TRAINING A NETWORK FOR SINGING VOICE DETECTION

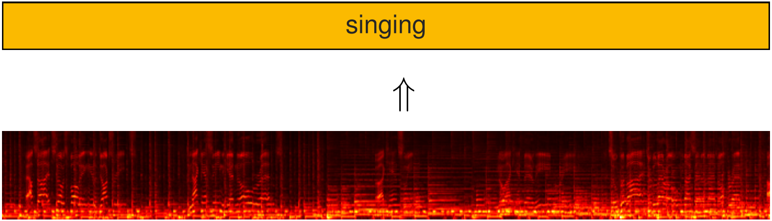
Task



Task



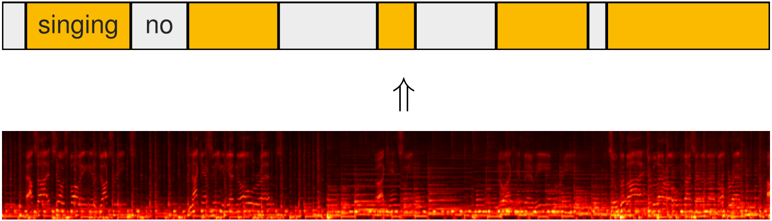
Task



Challenge: Weak training examples

10,000 30-second song clips with single label each:
“contains voice” or “does not contain voice”

Task



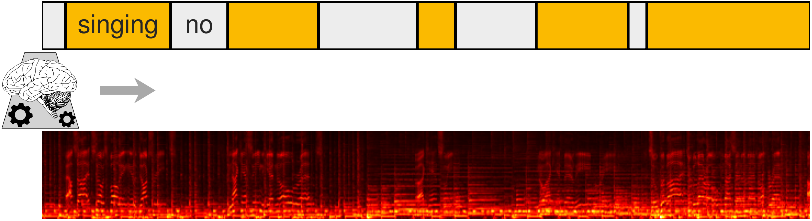
Challenge: Weak training examples

10,000 30-second song clips with single label each:
“contains voice” or “does not contain voice”

Baseline

100 30-second song clips with subsecond-wise annotations

Task



Challenge: Weak training examples

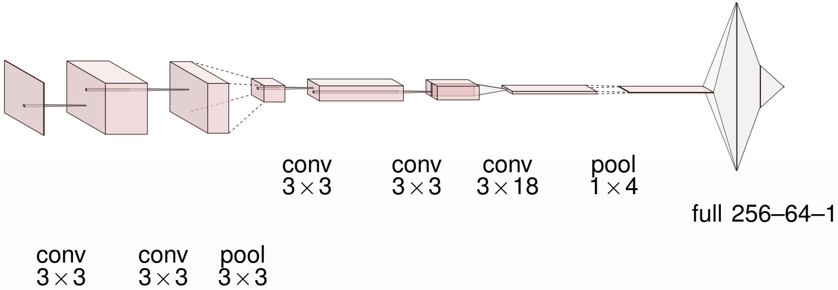
10,000 30-second song clips with single label each:
“contains voice” or “does not contain voice”

Baseline

100 30-second song clips with subsecond-wise annotations

Network Architecture

Input: a 115 × 80 spectrogram excerpt (1.6 s)

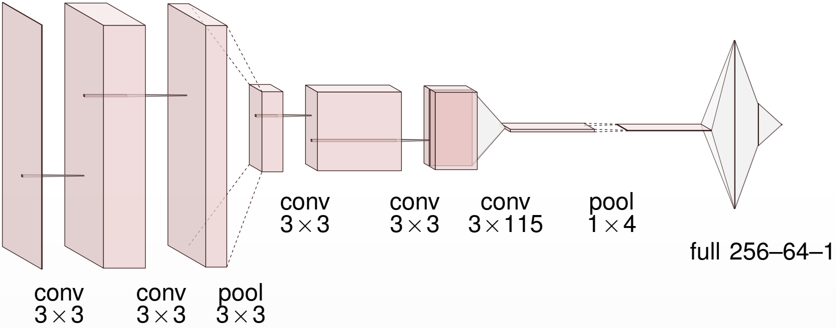


Output: Probability of singing voice at center of input

Postprocessing: Sliding median filter, thresholding

Network Architecture

Input: a 115×372 spectrogram excerpt (1.6 s), no mel scaling

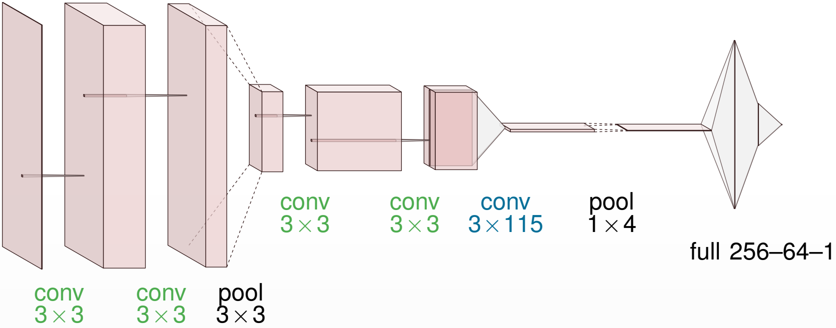


Output: Probability of singing voice at center of input

Postprocessing: Sliding median filter, thresholding

Network Architecture

Input: a 115 × 372 spectrogram excerpt (1.6 s), no mel scaling

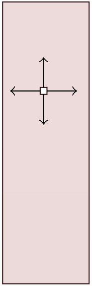


Output: Probability of singing voice at center of input

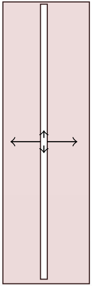
Postprocessing: Sliding median filter, thresholding

Network Architecture

small convolutions at first



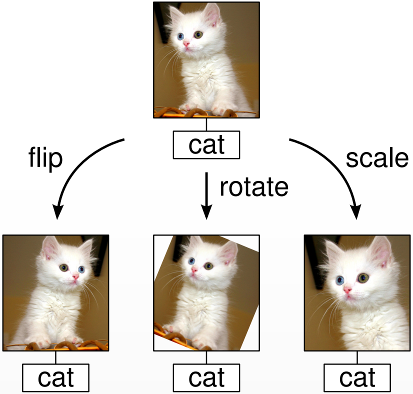
large convolution near end



Data Augmentation

Goal: Enhance variability of small training set, for fair comparison

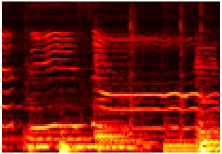
Idea: Augment training data, as common in computer vision



Catch: not invented for music yet

Data Augmentation

original

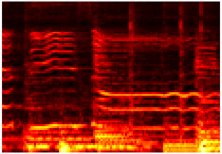


augmented

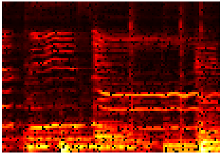
?

Data Augmentation

original



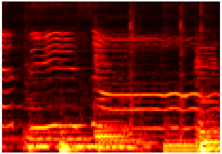
augmented



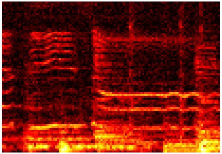
noise

Data Augmentation

original



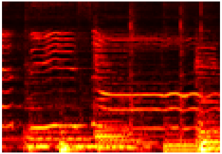
augmented



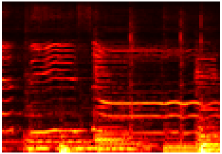
noise

Data Augmentation

original



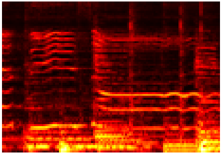
augmented



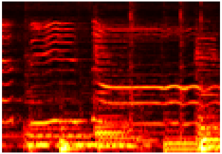
pitch shift

Data Augmentation

original



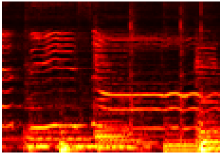
augmented



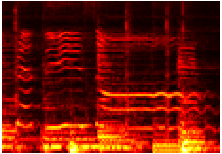
pitch shift

Data Augmentation

original



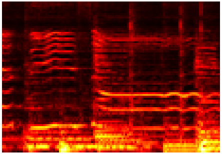
augmented



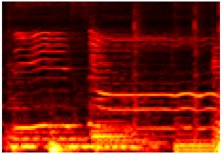
time stretch

Data Augmentation

original



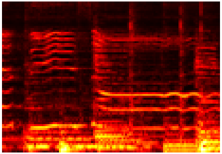
augmented



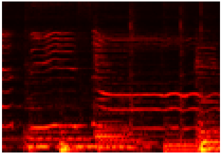
time stretch

Data Augmentation

original



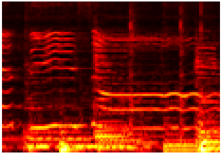
augmented



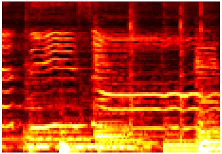
loudness

Data Augmentation

original



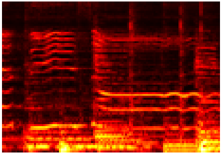
augmented



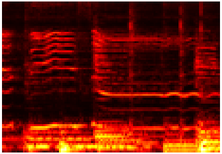
loudness

Data Augmentation

original



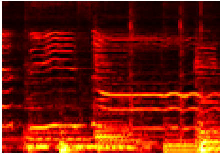
augmented



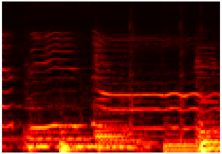
filtering

Data Augmentation

original



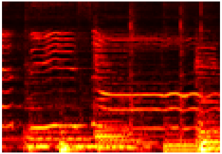
augmented



filtering

Data Augmentation

original



augmented

!

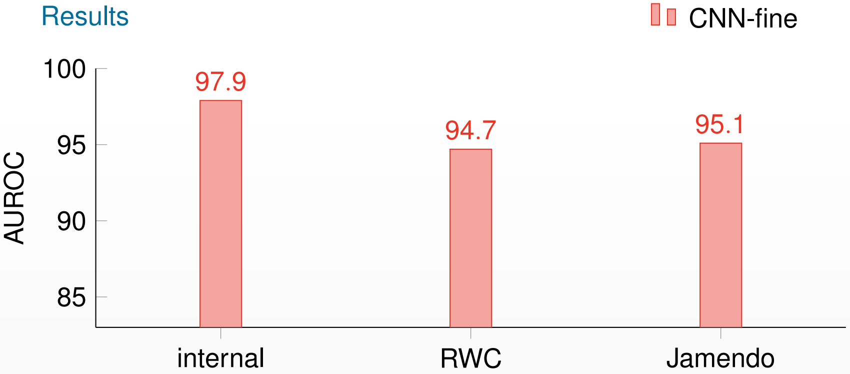
Best: pitch shifting ±30%,
time stretching ±30%,
filtering ±10 dB

Starting Point

Training data

100 30-second song clips with accurate annotations:
“when is voice?”

Results



Weak Labels: Naive Approach

Training data

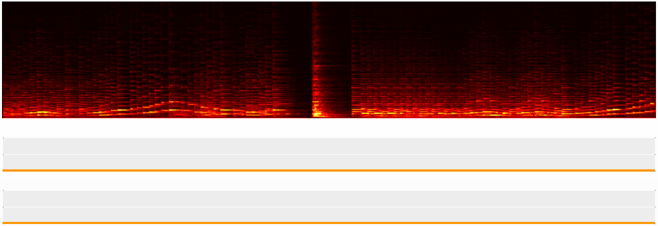
10,000 30-second song clips with single label each:
“contains voice” or “does not contain voice”

Weak Labels: Naive Approach

Training data

10,000 30-second song clips with single label each:
“contains voice” or “does not contain voice”

“does not contain voice”: propagate to all instances



train labels

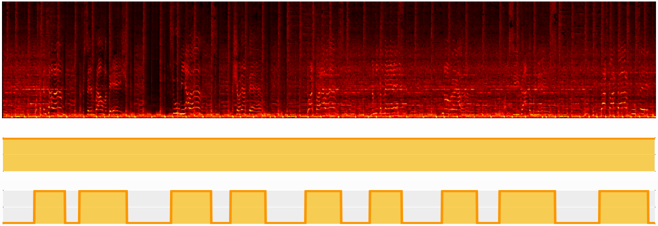
actual truth

Weak Labels: Naive Approach

Training data

10,000 30-second song clips with single label each:
“contains voice” or “does not contain voice”

“contains voice”: also propagate to all instances



train labels

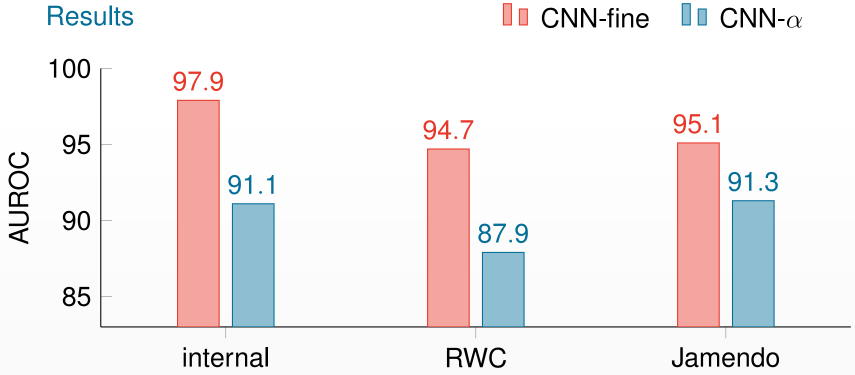
actual truth

Weak Labels: Naive Approach

Training data

10,000 30-second song clips with single label each:
“contains voice” or “does not contain voice”

Results

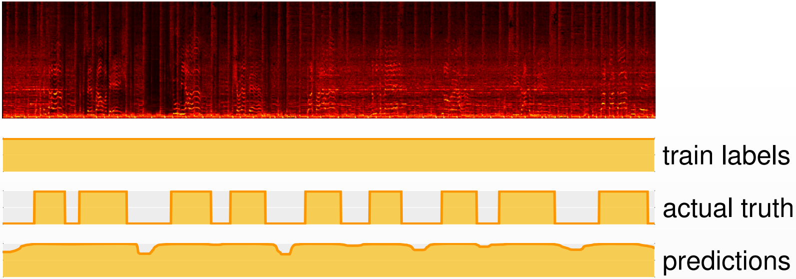


Weak Labels: Naive Approach

Training data

10,000 30-second song clips with single label each:
“contains voice” or “does not contain voice”

Prediction on training example

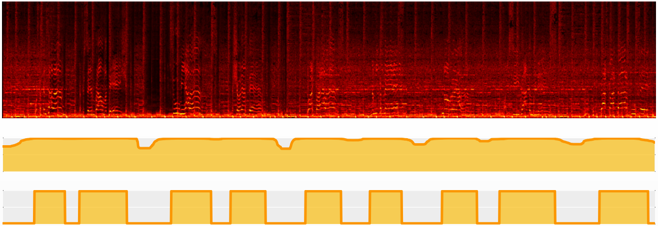


Self-Improvement 1: Pseudo-Labeling

Training data

10,000 30-second song clips with single label each:
“contains voice” or “does not contain voice”

“contains voice”: use predictions of CNN-α



train labels

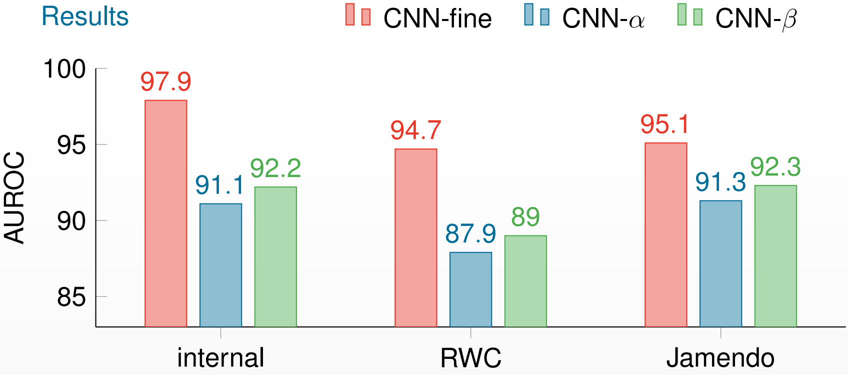
actual truth

Self-Improvement 1: Pseudo-Labeling

Training data

10,000 30-second song clips with single label each:
“contains voice” or “does not contain voice”

Results



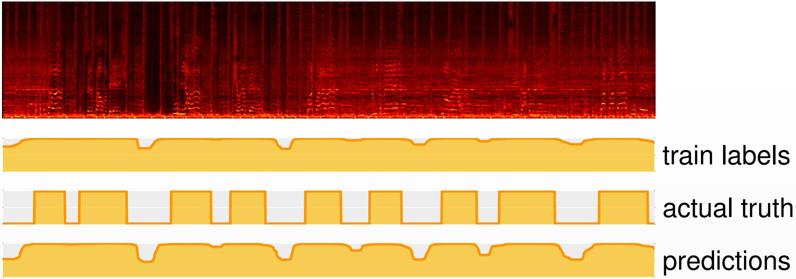
Self-Improvement 1: Pseudo-Labeling

Self-Improvement 1: Pseudo-Labeling

Training data

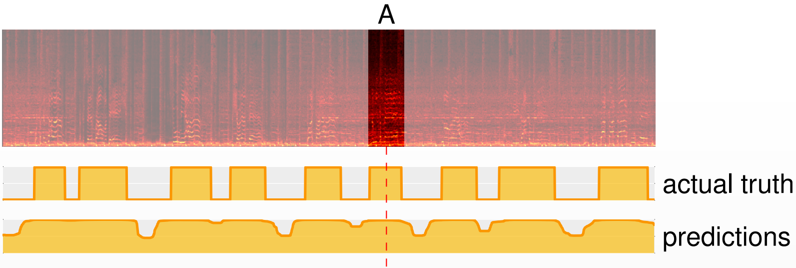
10,000 30-second song clips with single label each:
“contains voice” or “does not contain voice”

Prediction on training example



Overshoot

A excerpt contains voice throughout ✔
B excerpt does not contain voice
C excerpt contains voice at edge

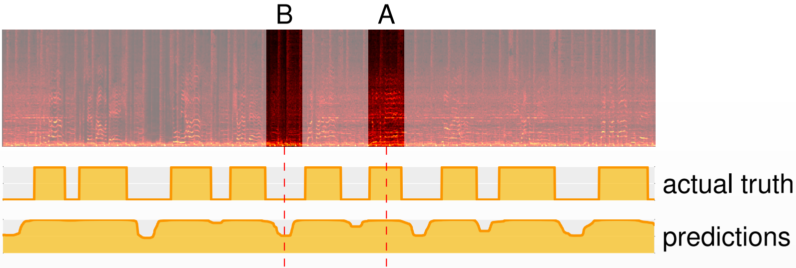


Overshoot

A excerpt contains voice throughout ✔

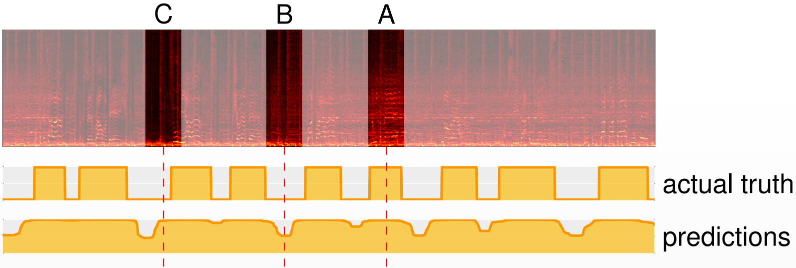
B excerpt does not contain voice ✔

C excerpt contains voice at edge



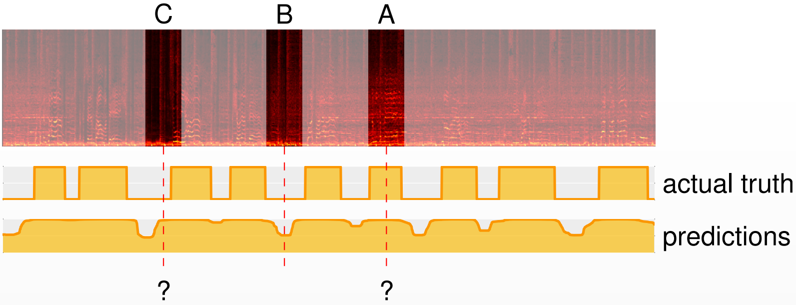
Overshoot

A excerpt contains voice throughout ✔
B excerpt does not contain voice ✔
C excerpt contains voice at edge ✘



Overshoot

- A excerpt contains voice throughout ✔
- B excerpt does not contain voice ✔
- C excerpt contains voice at edge ✘

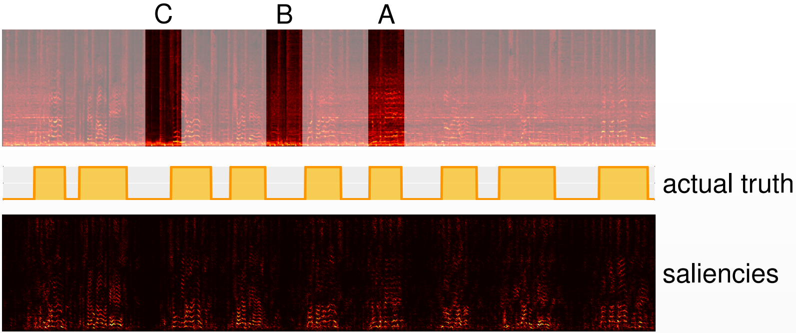


Overshoot correction:

Check if central input frame was relevant for positive prediction

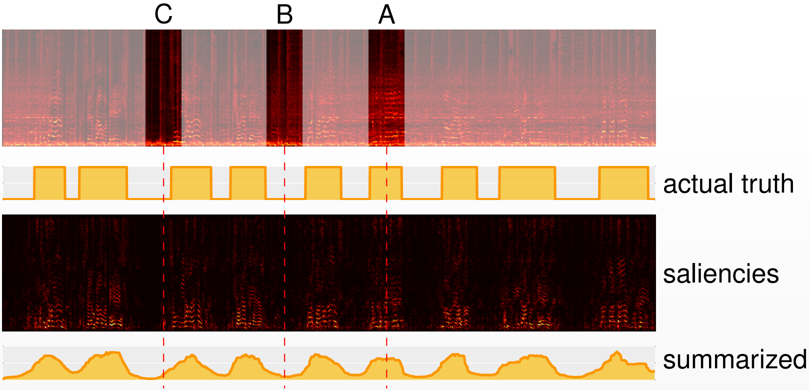
Overshoot Correction

- A excerpt contains voice throughout ✔
- B excerpt does not contain voice ✔
- C excerpt contains voice at edge ✘



Overshoot Correction

- A excerpt contains voice throughout ✔
- B excerpt does not contain voice ✔
- C excerpt contains voice at edge ✔

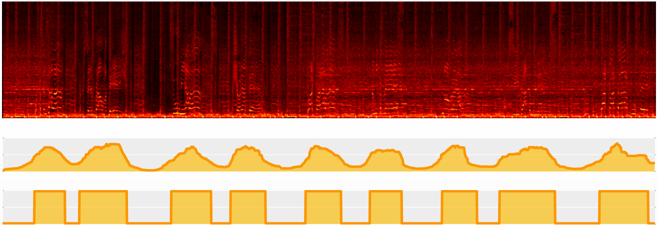


Self-Improvement 2: Overshoot Correction

Training data

10,000 30-second song clips with single label each:
“contains voice” or “does not contain voice”

“contains voice”: use saliencies of CNN-β



train labels

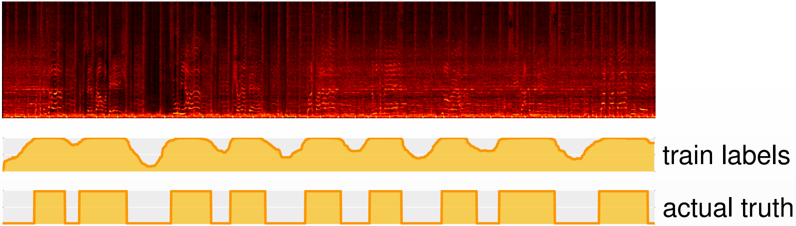
actual truth

Self-Improvement 2: Overshoot Correction

Training data

10,000 30-second song clips with single label each:
“contains voice” or “does not contain voice”

“contains voice”: use squashed saliencies of CNN-β

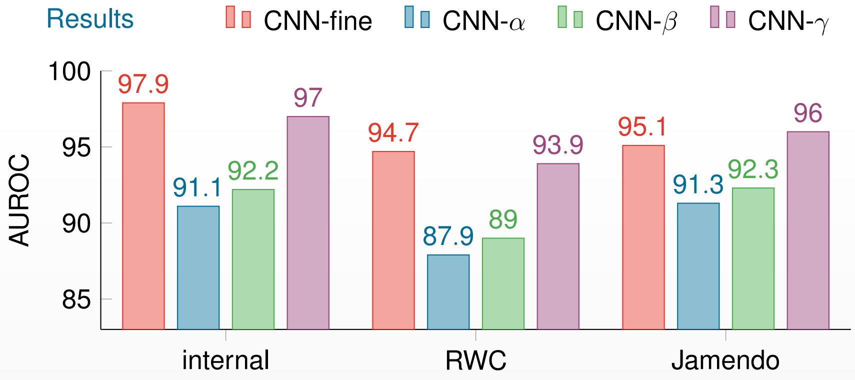


Self-Improvement 2: Overshoot Correction

Training data

10,000 30-second song clips with single label each:
"contains voice" or "does not contain voice"

Results

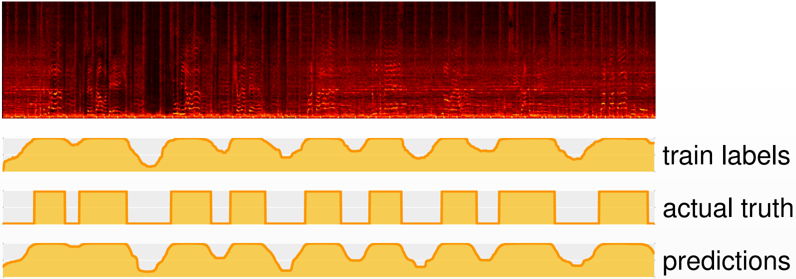


Self-Improvement 2: Overshoot Correction

Training data

10,000 30-second song clips with single label each:
“contains voice” or “does not contain voice”

Prediction on training example

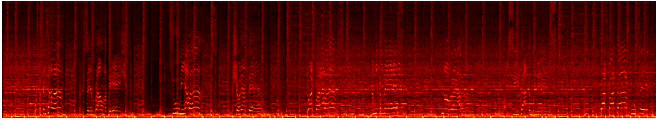


Bonus: Spectral Localization

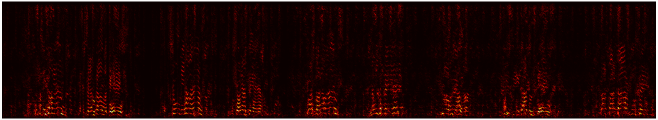
Goal

Take a song and predict ~~when~~ it contains voice
where

Implementation



Compute saliency map, scale to match value range of input.

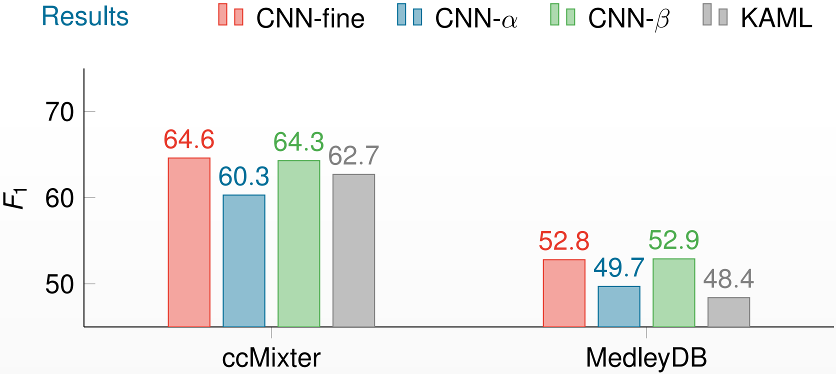


Bonus: Spectral Localization

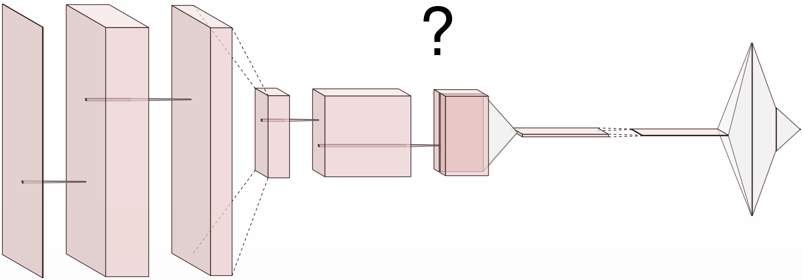
Goal

Take a song and predict ~~when~~ it contains voice
where

Results



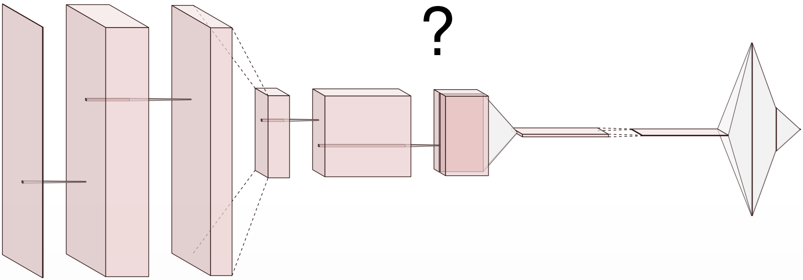
How does the net work?



Demo

FAILURE MODE 1

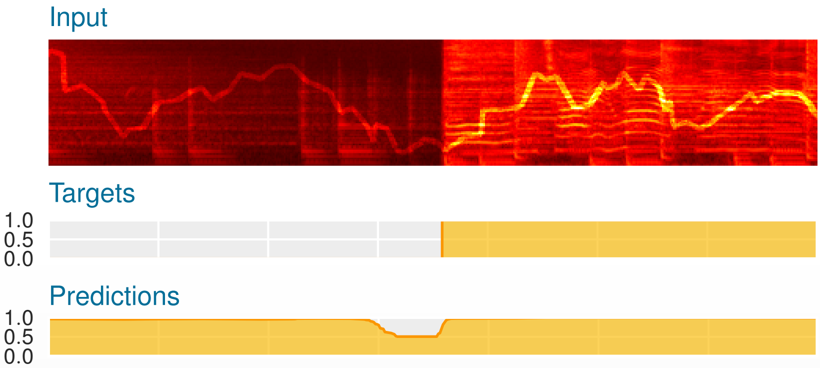
How does the net work?



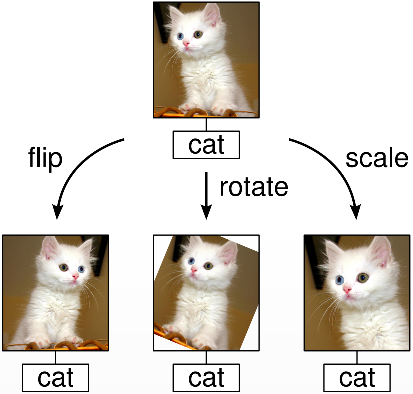
Demo

Demo 2

How does the net fail?



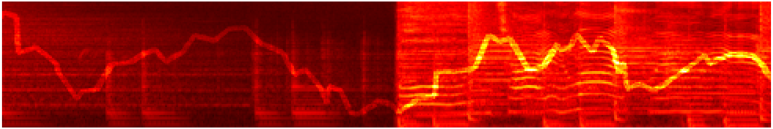
Taming the horse



Taming the horse

Taming the horse

Input



Targets

Predictions

Predictions when trained with sloped lines augmentation

However, accuracy on unmodified data does not improve.

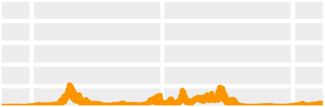
FAILURE MODE 2

Listening test

Listening test

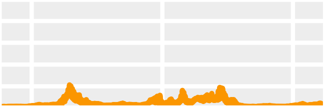
Listening test

0 dB



Listening test

0 dB

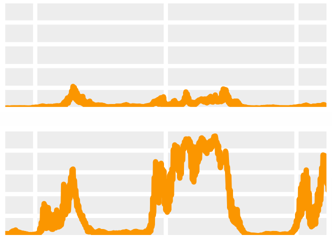


+6 dB

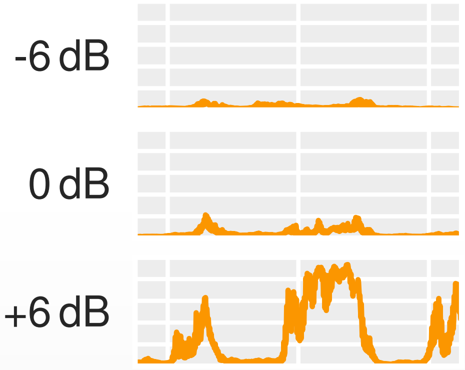
Listening test

0 dB

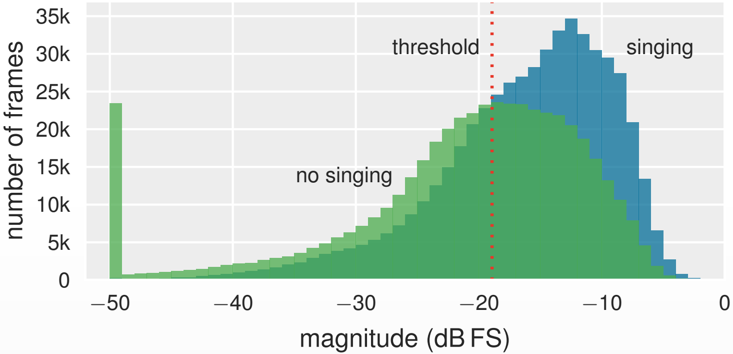
+6 dB



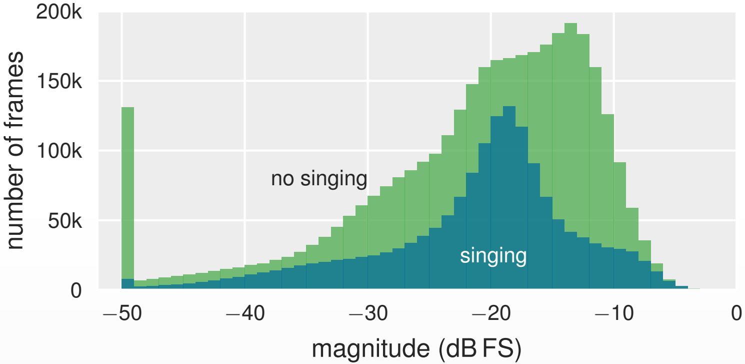
Listening test



Understanding the horse



Taming the horse

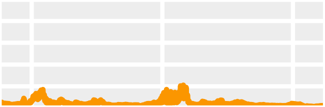


Listening test

Listening test

Listening test

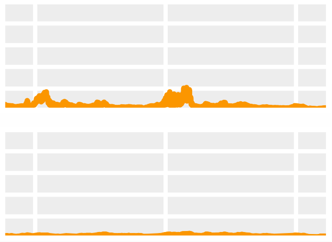
0 dB



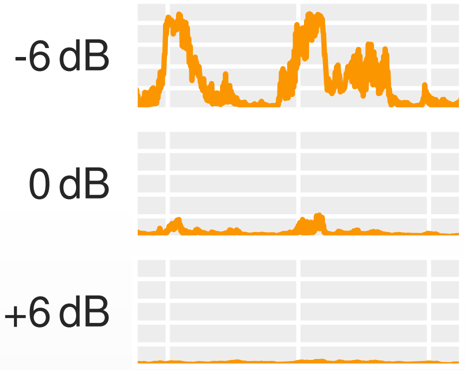
Listening test

0 dB

+6 dB



Listening test

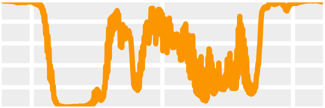


Listening test

Listening test

Listening test

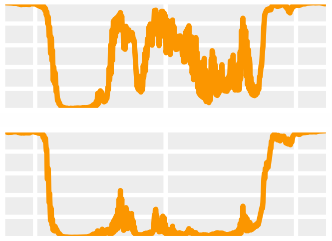
0 dB



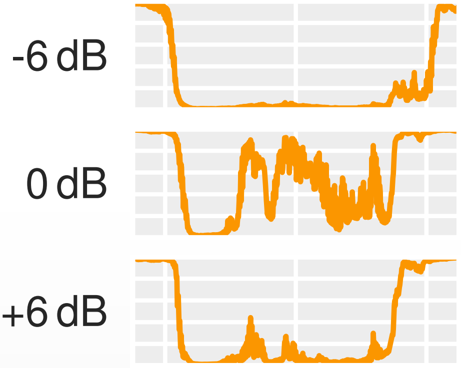
Listening test

0 dB

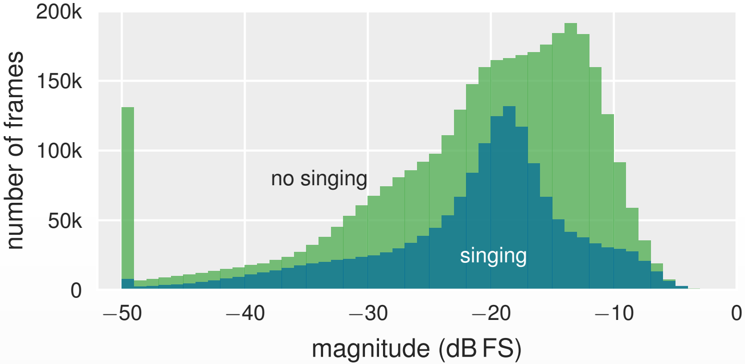
+6 dB



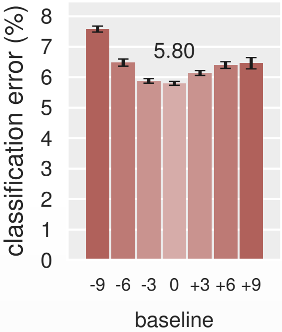
Listening test



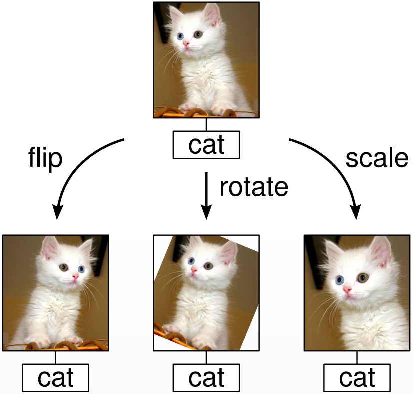
Understanding the horse



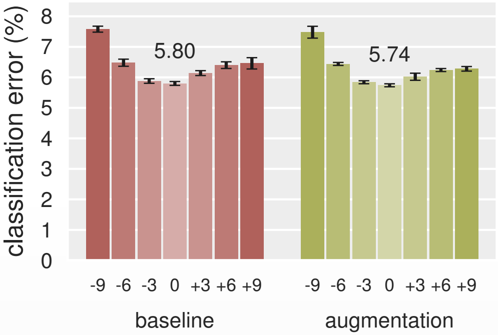
Understanding the horse



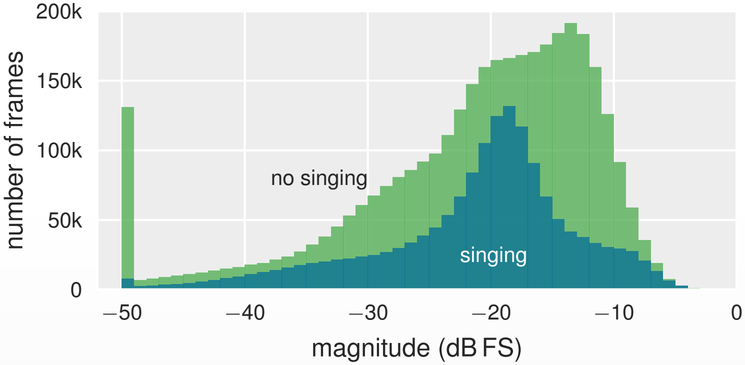
Taming the horse



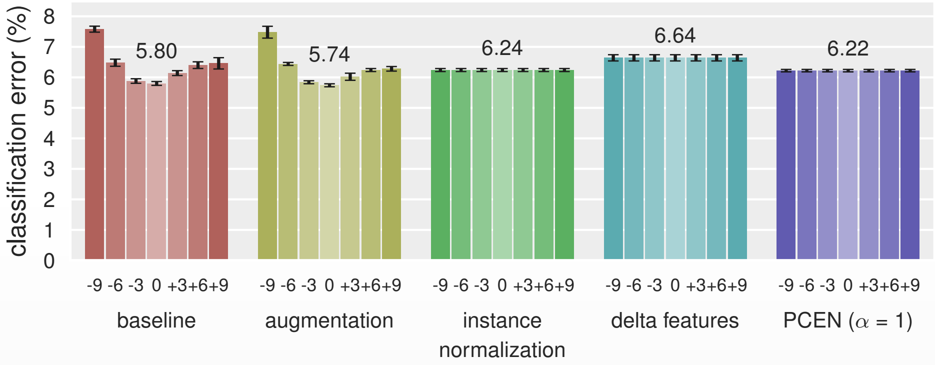
Taming the horse



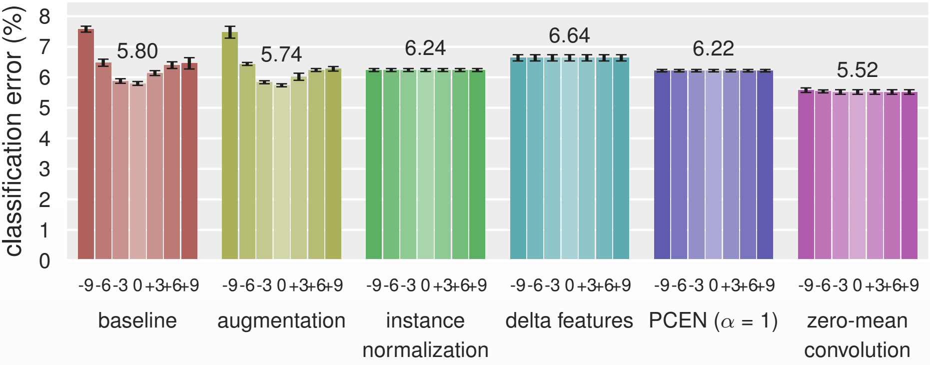
Understanding the horse



Taming the horse

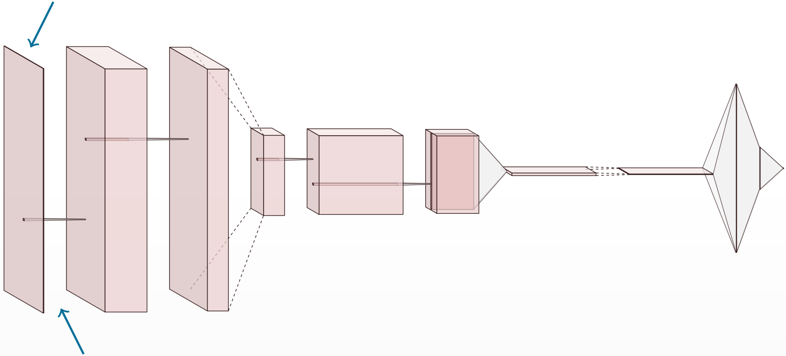


Taming the horse



Zero-mean convolution

Log-magnitude input, turns scale into shift: $\log(\beta x) = \log(\beta) + \log(x)$



First convolution coefficients constrained to sum to zero, removes shift

CONCLUSION

Recap and Takeaways

1. Trained a neural network on weakly-labeled audio recordings
   - found recipe to reach same accuracy as using strong labels
   - 10,000 weak examples $\approx$ 100 strong examples
2. Network found overly sensitive to wiggly lines
   - misses long drawn notes, mistakes e-guitars/sax for vocals
   - data augmentation only helps against hand-drawn fakes
3. Network found sensitive to sound level
   - went unnoticed in standard train/test setting
   - customized model avoids this

- If there is any bias in the data, it will be exploited
- Finding such exploits requires custom interventions
- Leave the comfort zone of your test set!

Code: `github.com/f0k/ismir2015`, `singing_horse`, `ismir2018`

Recap and Takeaways

1. Trained a neural network on weakly-labeled audio recordings
   - found recipe to reach same accuracy as using strong labels
   - 10,000 weak examples $\approx$ 100 strong examples
2. Network found overly sensitive to wiggly lines
   - misses long drawn notes, mistakes e-guitars/sax for vocals
   - data augmentation only helps against hand-drawn fakes
3. Network found sensitive to sound level
   - went unnoticed in standard train/test setting
   - customized model avoids this

- If there is any bias in the data, it will be exploited
- Finding such exploits requires custom interventions
- Leave the comfort zone of your test set!

Code: `github.com/f0k/ismir2015`, `singing_horse`, `ismir2018`

Recap and Takeaways

1. Trained a neural network on weakly-labeled audio recordings
   - found recipe to reach same accuracy as using strong labels
   - 10,000 weak examples $\approx$ 100 strong examples
2. Network found overly sensitive to wiggly lines
   - misses long drawn notes, mistakes e-guitars/sax for vocals
   - data augmentation only helps against hand-drawn fakes
3. Network found sensitive to sound level
   - went unnoticed in standard train/test setting
   - customized model avoids this

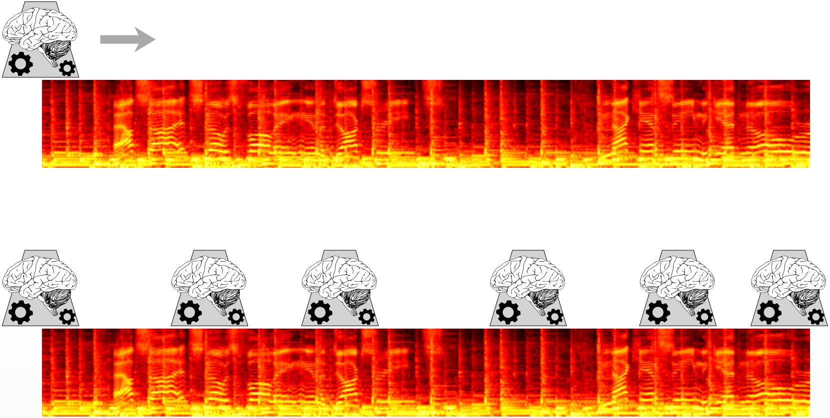
- If there is any bias in the data, it will be exploited
- Finding such exploits requires custom interventions
- Leave the comfort zone of your test set!

  Code: `github.com/f0k/ismir2015`, `singing_horse`, `ismir2018`

APPENDIX

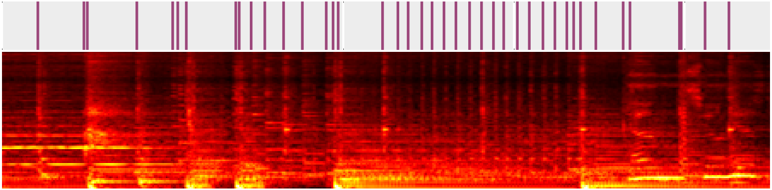
K

Disadvantages of RNNs

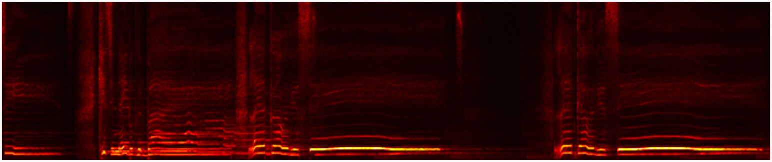


Advantages of RNNs?

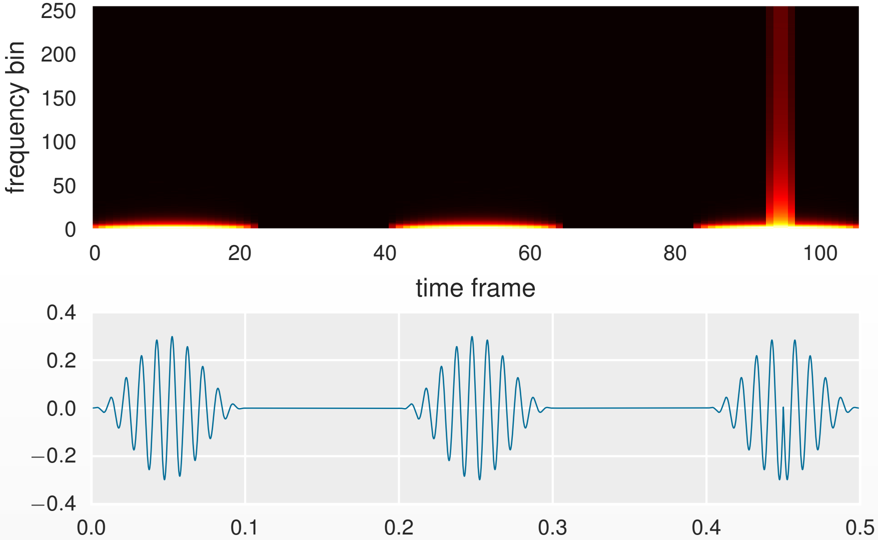
- Larger context? Infinite context?
- Make use of regularities in music?



- Make use of temporal continuity?



Phase invariance



Spectral Localization Evaluation

Goal

Take a song and predict ~~when~~ it contains voice
where

Evaluation

Compare saliency map P_{ij} to spectrogram of pure-vocal track T_{ij}.

True positives:

$$t = \sum_{i,j} \min(P_{ij}, T_{ij})$$

Precision:

$$p = \frac{t}{\sum_{i,j} P_{ij}}$$

Recall:

$$r = \frac{t}{\sum_{i,j} T_{ij}}$$

F_1-measure:

$$f = \frac{2pr}{p + r}$$