Machine Learning for Audio Signals

ECE 272/472 Audio Signal Processing

Bochen Li

University of Rochester

Outline

- **Introduction**
- Audio Feature Extraction
- Audio Alignment and Matching
- Classifiers
- Evaluation Measures
- Application 1: Sound Classification
- Application 2: Keyword Spotting

Introduction

Audio Signal Processing ⟺ Machine Learning

- **Speech**
  - Speech Recognition
  - Talker Recognition
  - Emotion Detection
  - Speech Enhancement

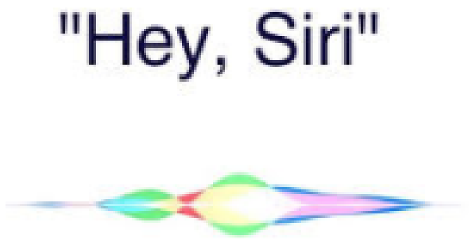
- **Music**
  - Pitch/Chord Estimation
  - Genre Classification
  - Source Separation

- **Other**
  - Sound Event Detection
  - Auditory Scene Classification

Introduction

Applications

Voice Assistant



Introduction

Applications

Algorithmic Music Recommendation

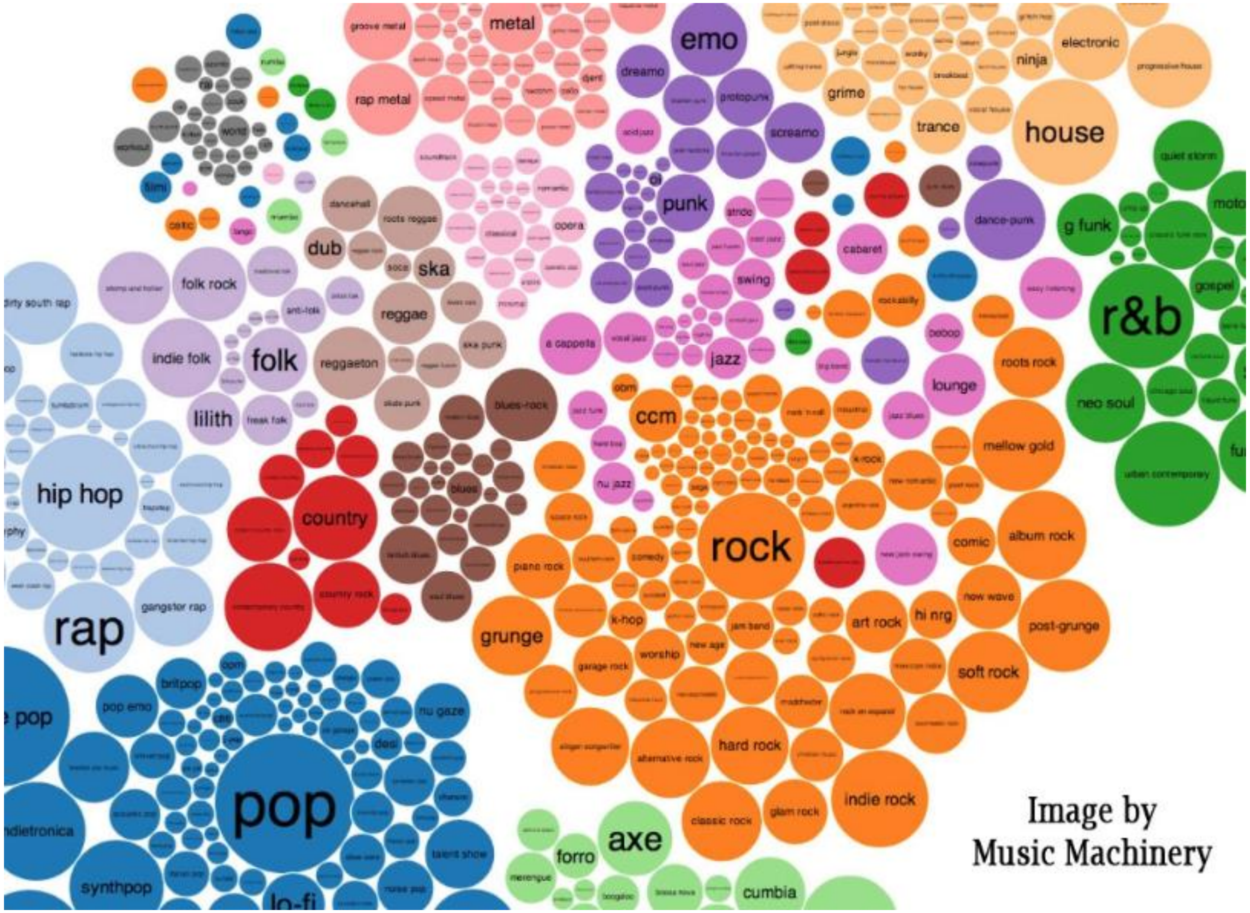
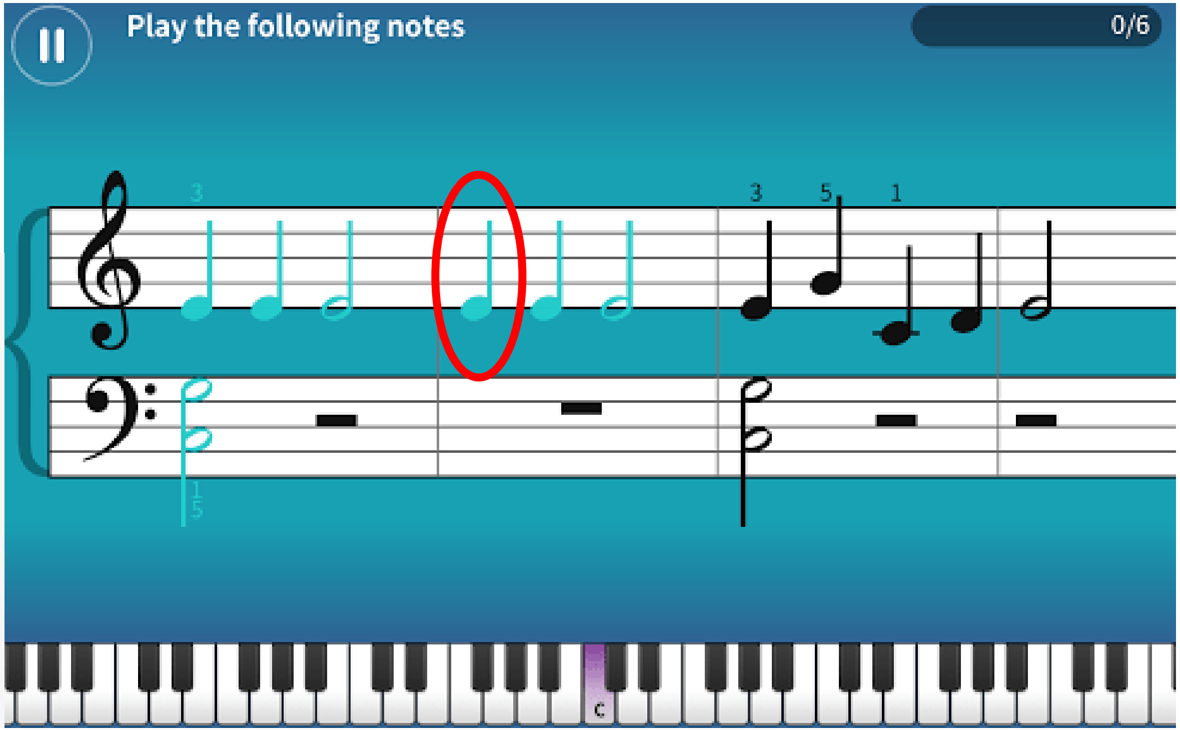


Image by
Music Machinery

Introduction

Applications

Music Tutor

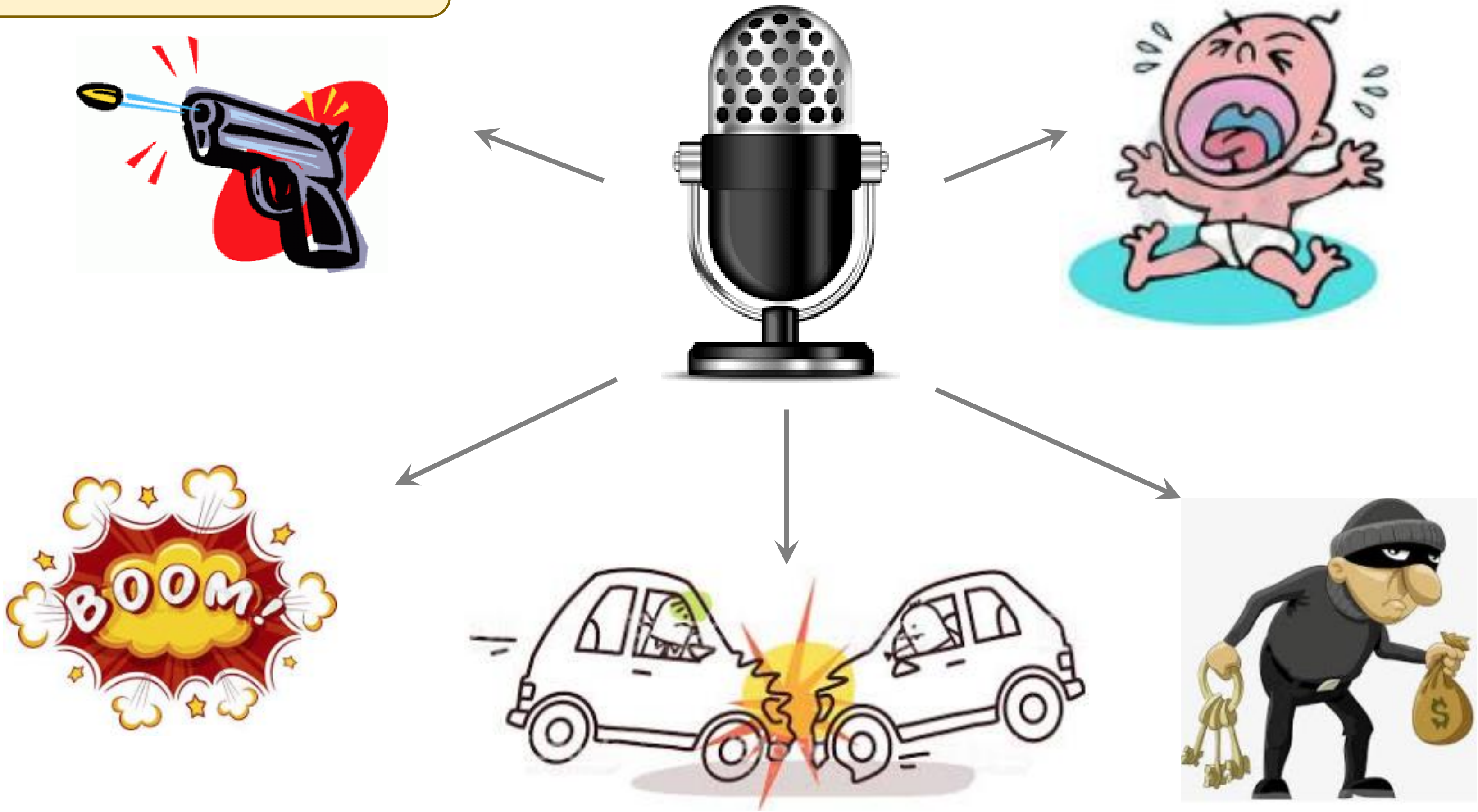


Play the following notes

Introduction

Applications

Security surveillance

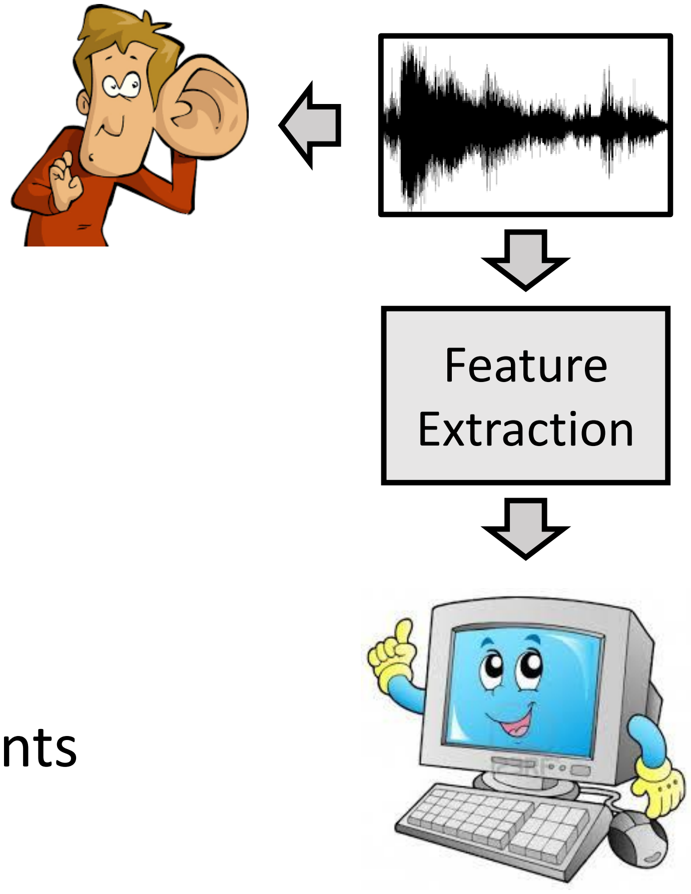


Outline

- Introduction
- **Audio Feature Extraction**
- Audio Alignment and Matching
- Classifiers
- Evaluation Measures
- Application 1: Sound Classification
- Application 2: Keyword Spotting

Audio Feature Extraction

- Energy
- Zero-Crossing Rate
- Pitch
- Chromagram
- Spectrogram
- Log-Mel Spectrogram
- Mel-Frequency Cepstral Coefficients



Feature Extraction

Audio Feature Extraction

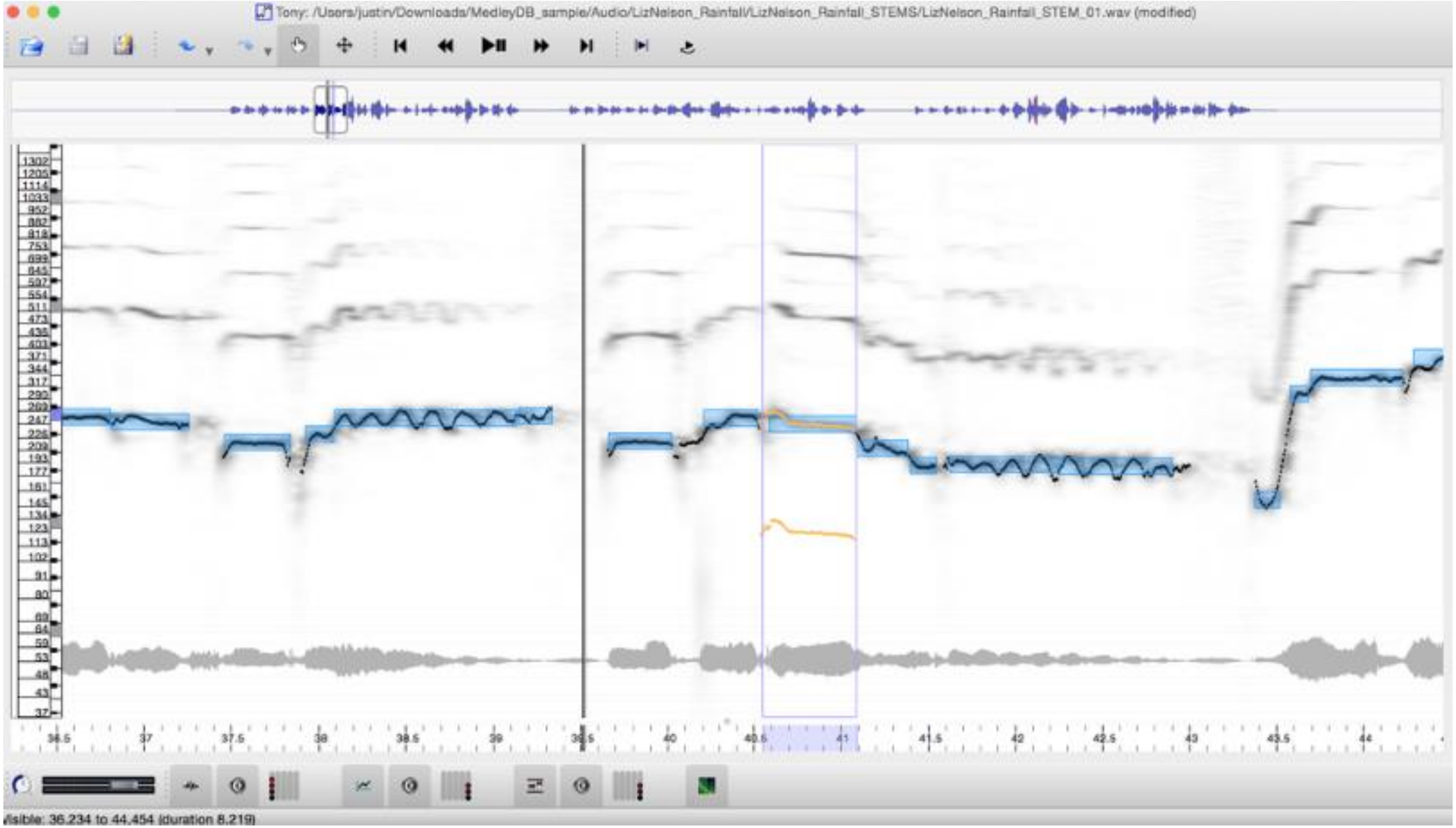
Pitch

Single Pitch Detection Methods

• Time domain: F0 = 1/periods	• Signal is periodic
• Frequency domain: F0 = greatest common divisor	• Spectral peaks have harmonic relations
• Cepstral domain: F0 = frequency gap	• Spectral peaks are equally spaced

Audio Feature Extraction

Pitch

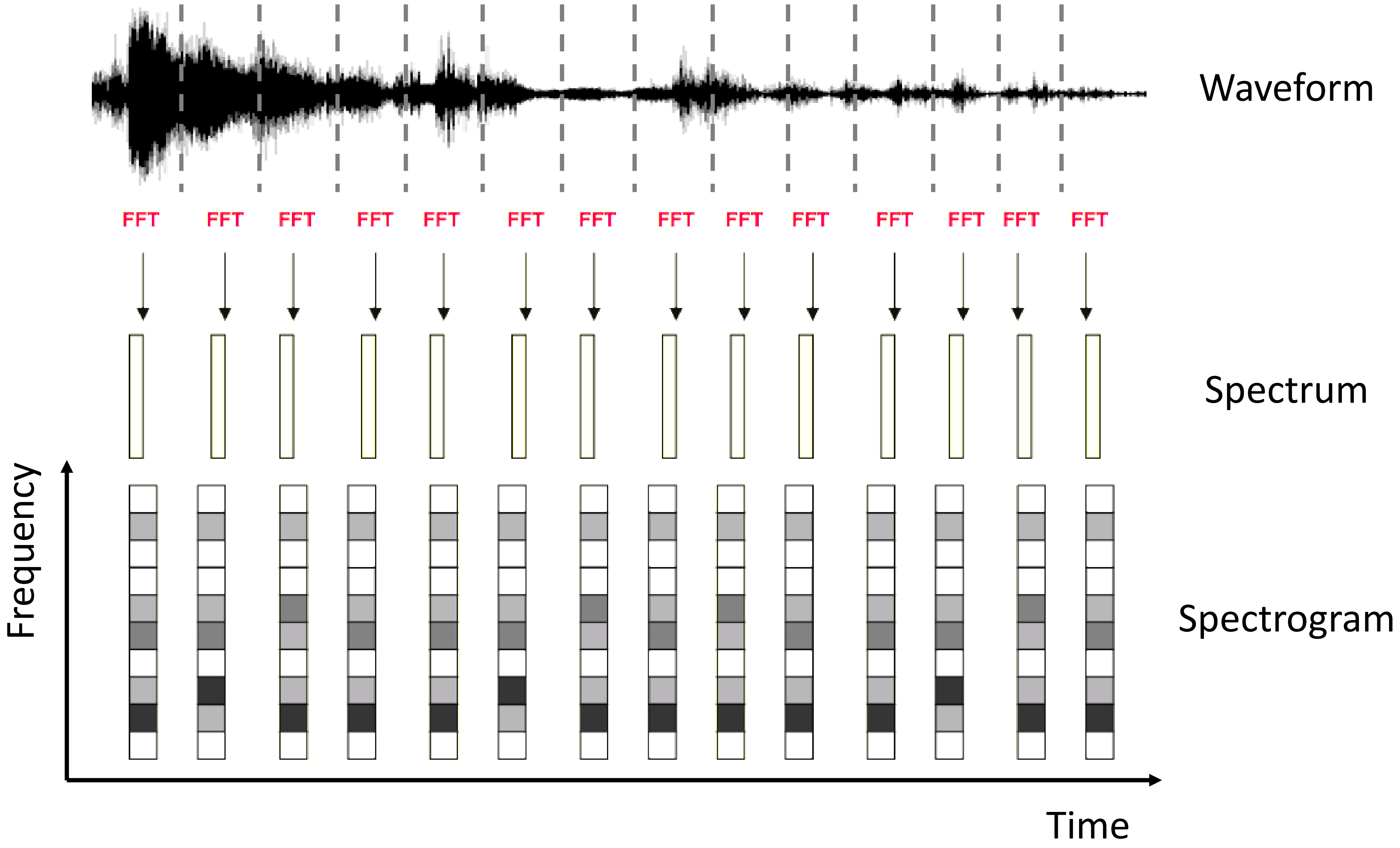


Tony: /Users/justin/Downloads/MedleyDB_sample/Audio/LizNelson_Rainfall/LizNelson_Rainfall_STEMS/LizNelson_Rainfall_STEM_01.wav (modified)

Visible: 36.234 to 44.454 (duration 8.219)

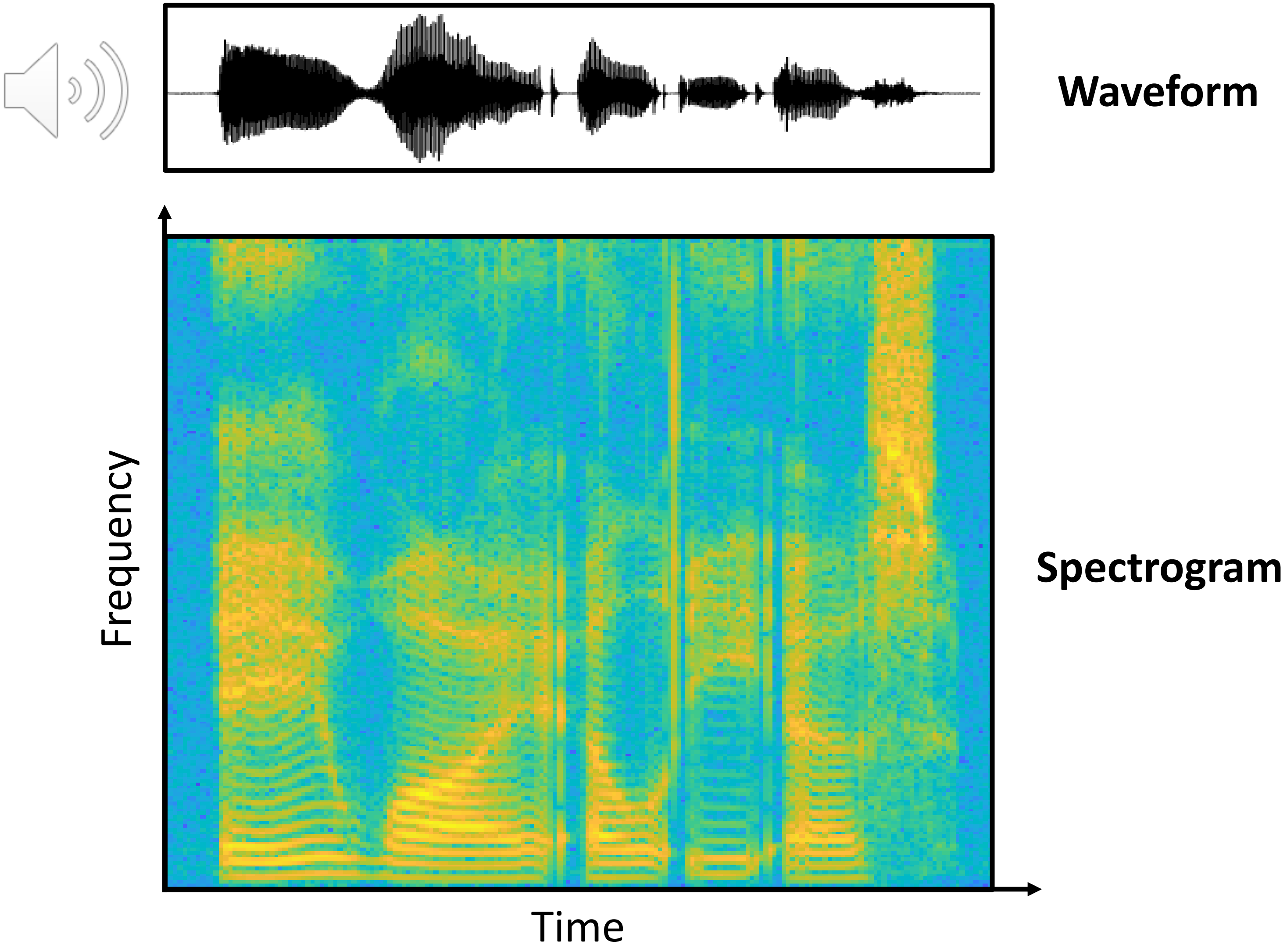
Audio Feature Extraction

Spectrogram



Audio Feature Extraction

Spectrogram



Audio Feature Extraction

Log-Mel Spectrogram

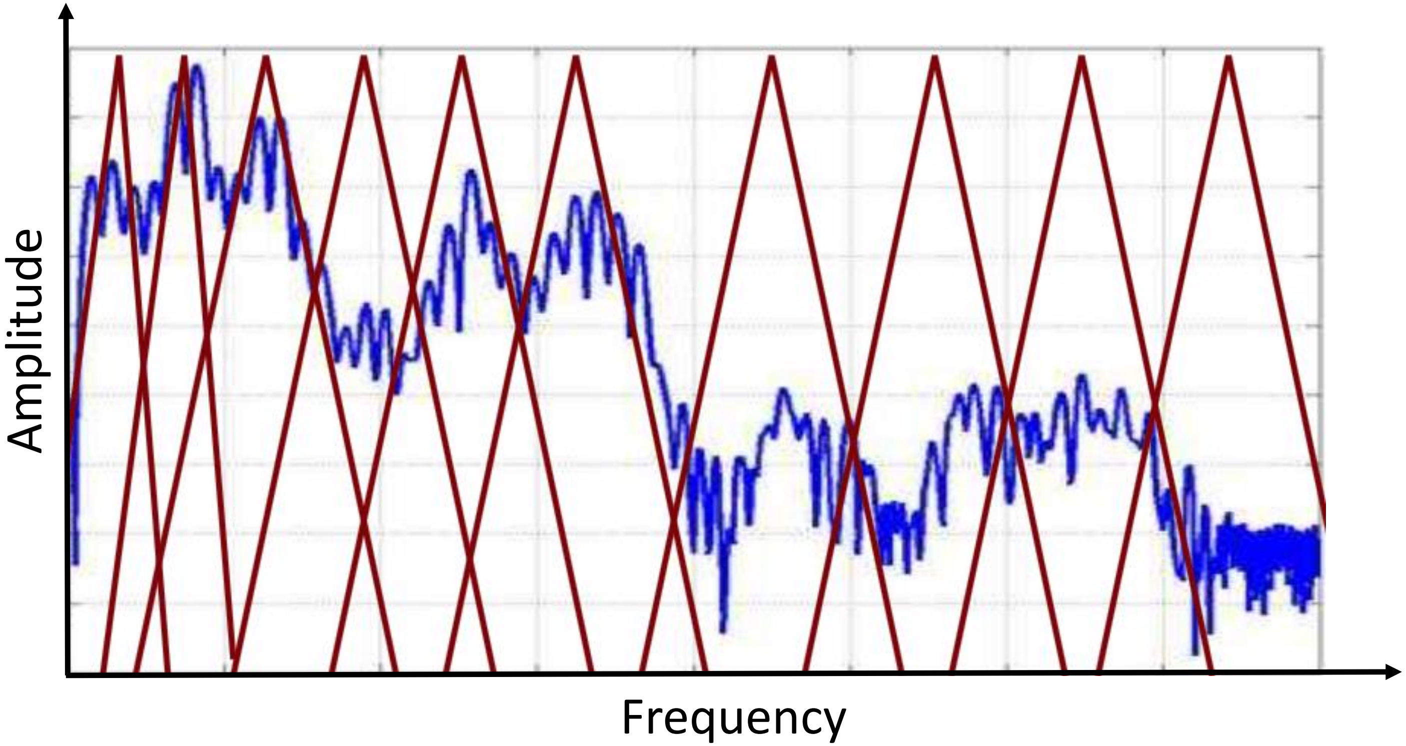
Mel-Frequency Analysis

- Human auditory systems respond to frequencies in log-scale
  - Finer frequency resolution for low frequencies
  - Coarser frequency resolution for high frequencies
- Mel-frequency (mel-scale) analysis is inspired by human auditory systems
  - More filters in low frequencies
  - Less filters in high frequencies
- Human auditory systems respond to amplitudes in log-scale → Log-mel spectrogram

Audio Feature Extraction

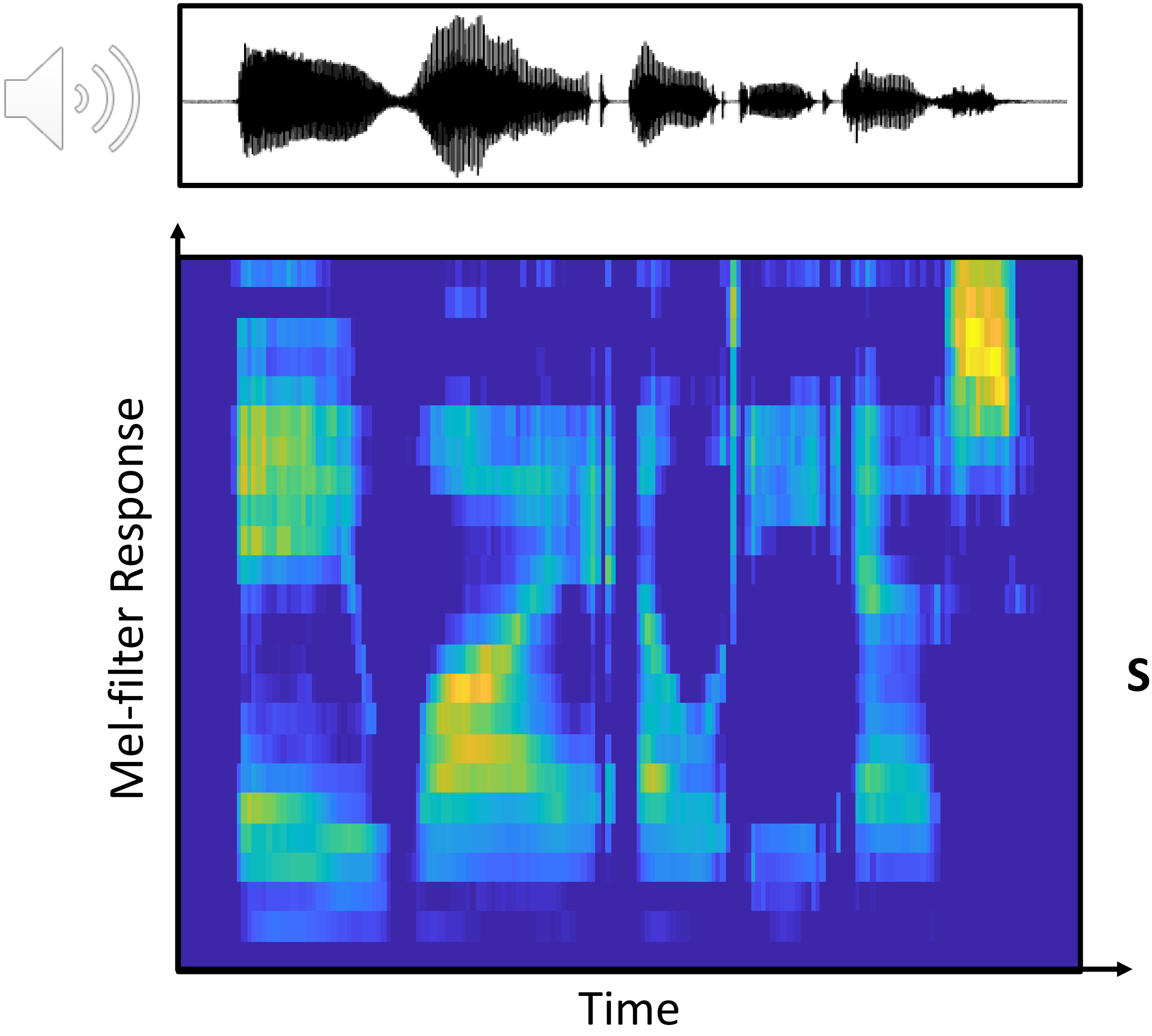
Log-Mel Spectrogram

Mel-Frequency Analysis



Audio Feature Extraction

Log-Mel Spectrogram



Waveform

Log-Mel Spectrogram

Audio Feature Extraction

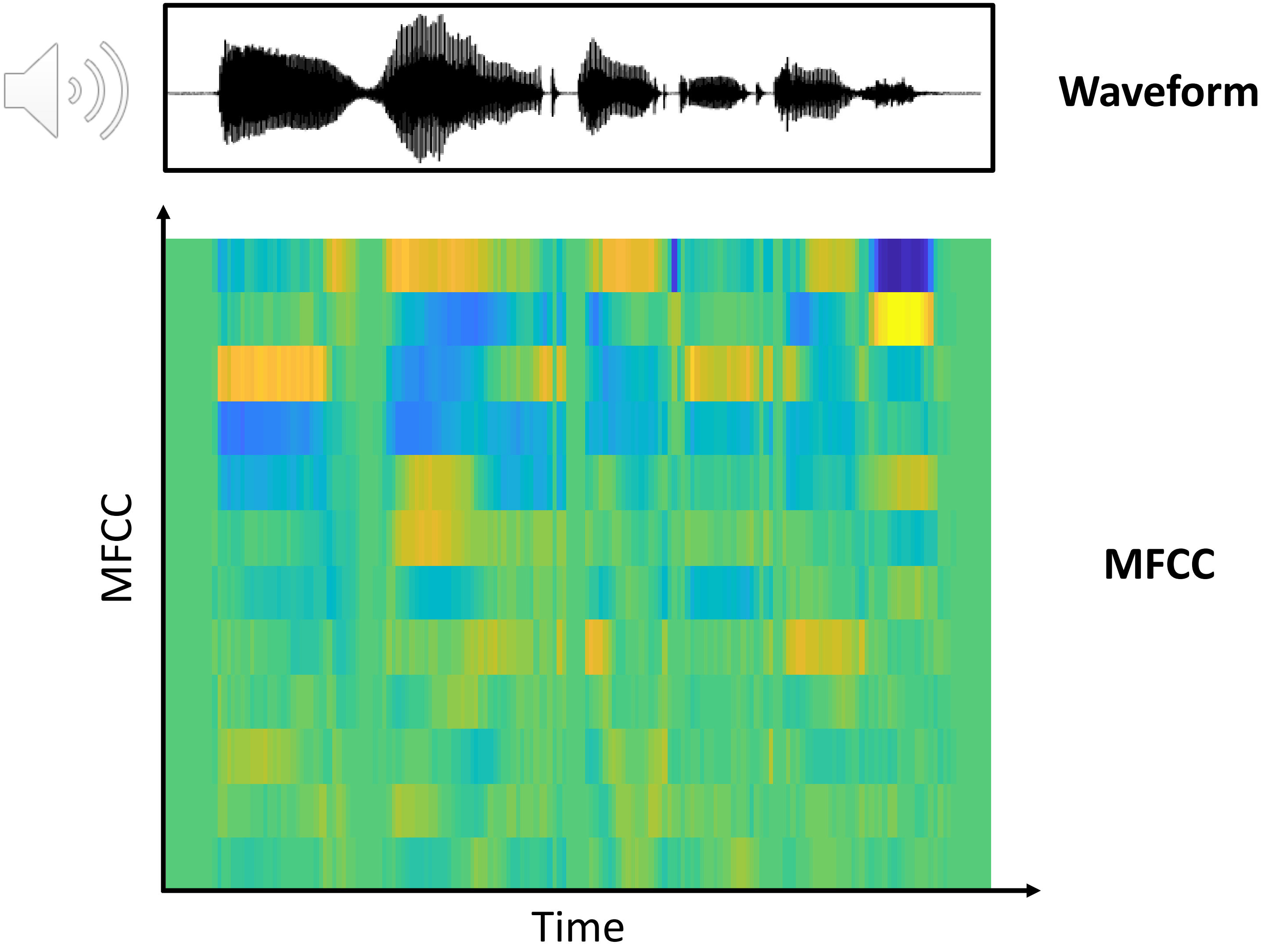
Mel-Frequency Cepstral Coefficients (MFCC)

Steps

1. Audio frame → FFT → Spectrum
2. Spectrum → Mel-Filters → Log-Mel Spectrum
3. Perform cepstral analysis
4. Take the first multiple cepstral coefficients as MFCCs

Audio Feature Extraction

Mel-Frequency Cepstral Coefficients (MFCC)



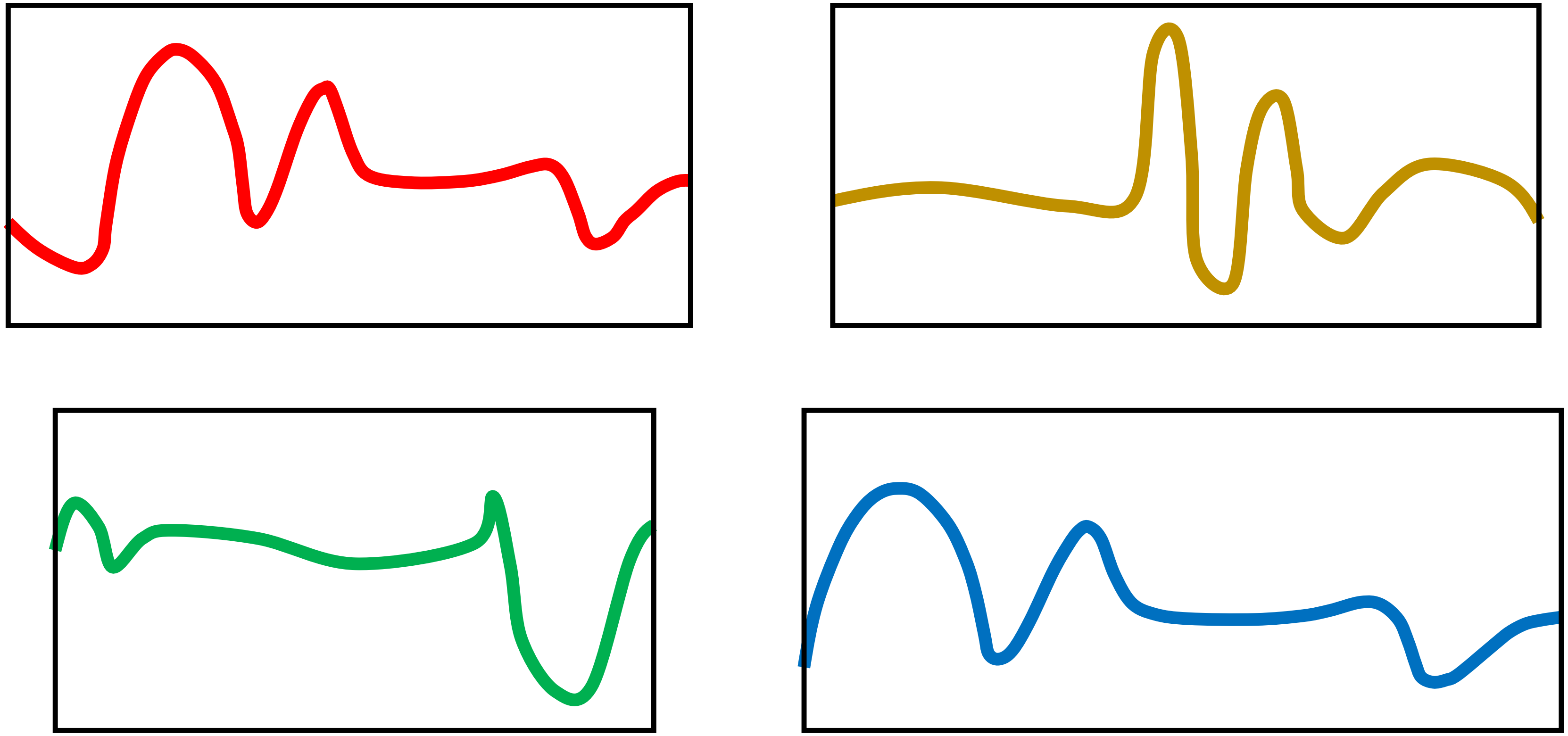
Outline

- Introduction
- Audio Feature Extraction
- **Audio Alignment and Matching**
- Classifiers
- Evaluation Measures
- Application 1: Sound Classification
- Application 2: Keyword Spotting

Audio Alignment and Matching

Motivation

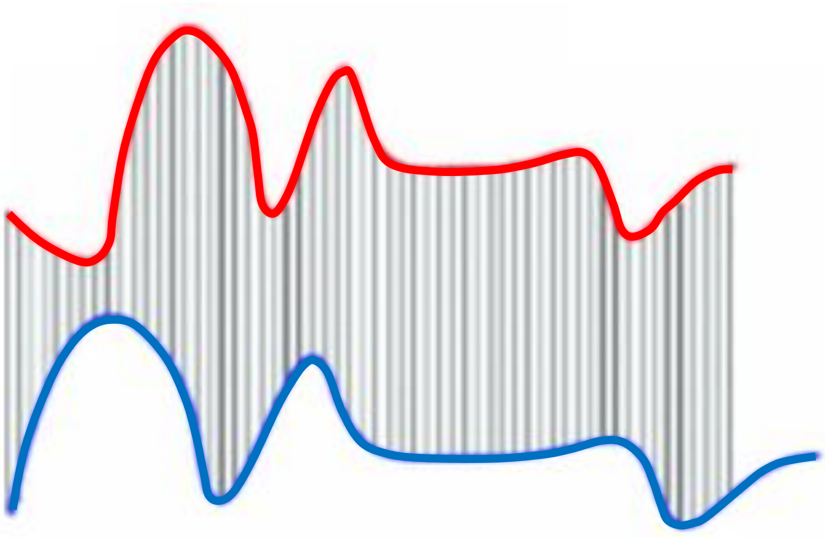
- Audio signals are time sequences
- How to measure the similarity?



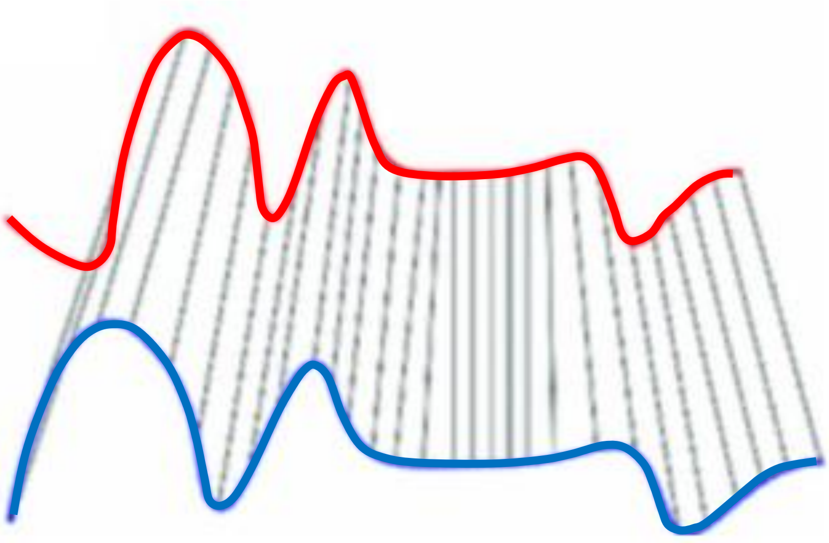
Audio Alignment and Matching

Motivation

- Audio signals are time sequences
- How to measure the similarity?



Pair-wise matching

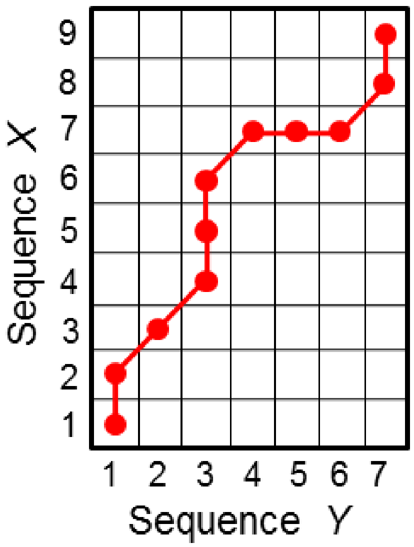


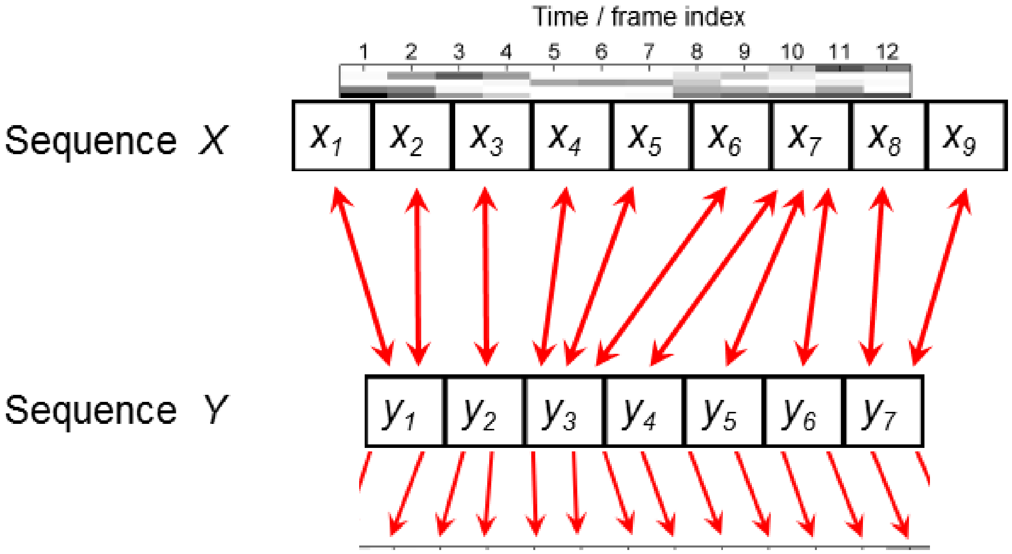
Warped matching

Audio Alignment and Matching

Dynamic Time Warping

Find the warping path

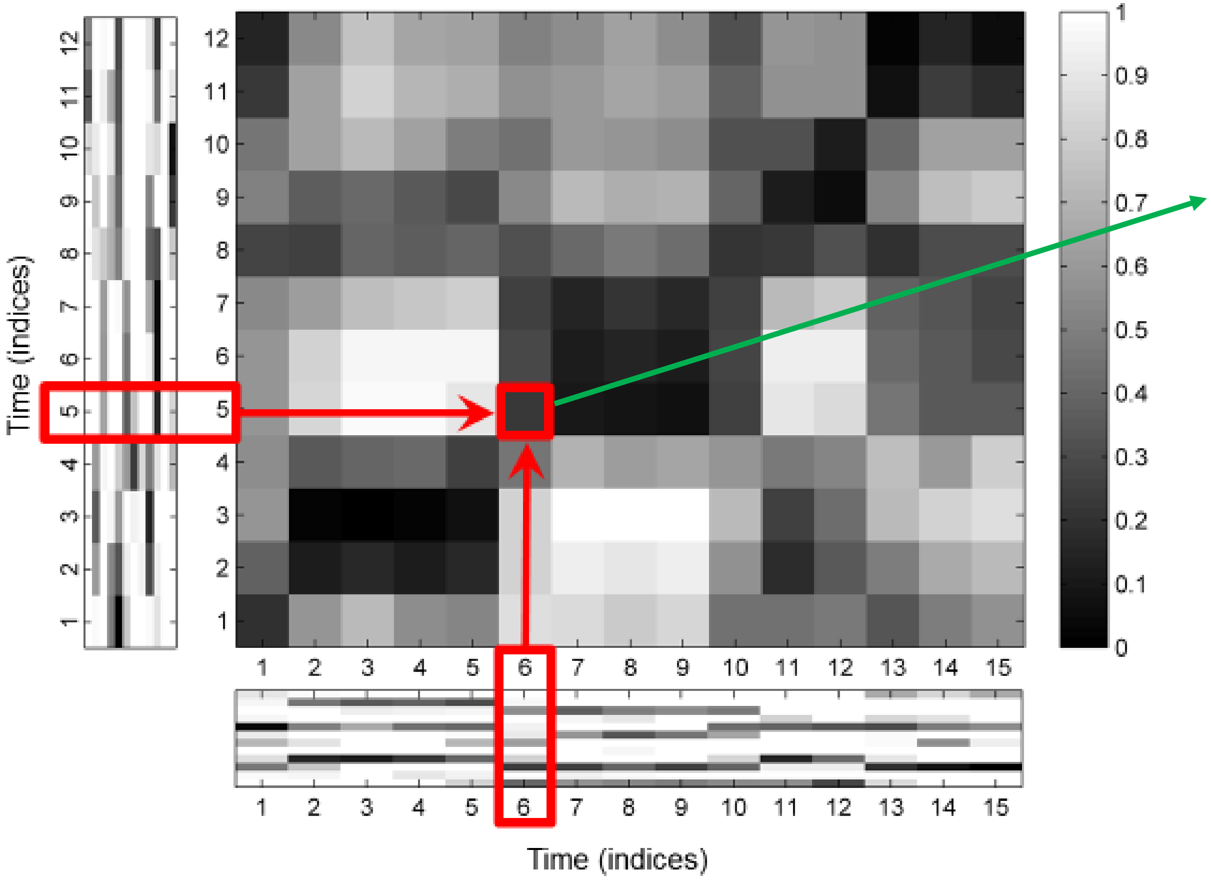



Time / frame index
1 2 3 4 5 6 7 8 9 10 11 12
Sequence X
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9
Sequence Y
y_1 y_2 y_3 y_4 y_5 y_6 y_7

Audio Alignment and Matching

Dynamic Time Warping

Step1: Calculate the local distance matrix $\mathbf{C} \in R^{M \times N}$



Vector distance:

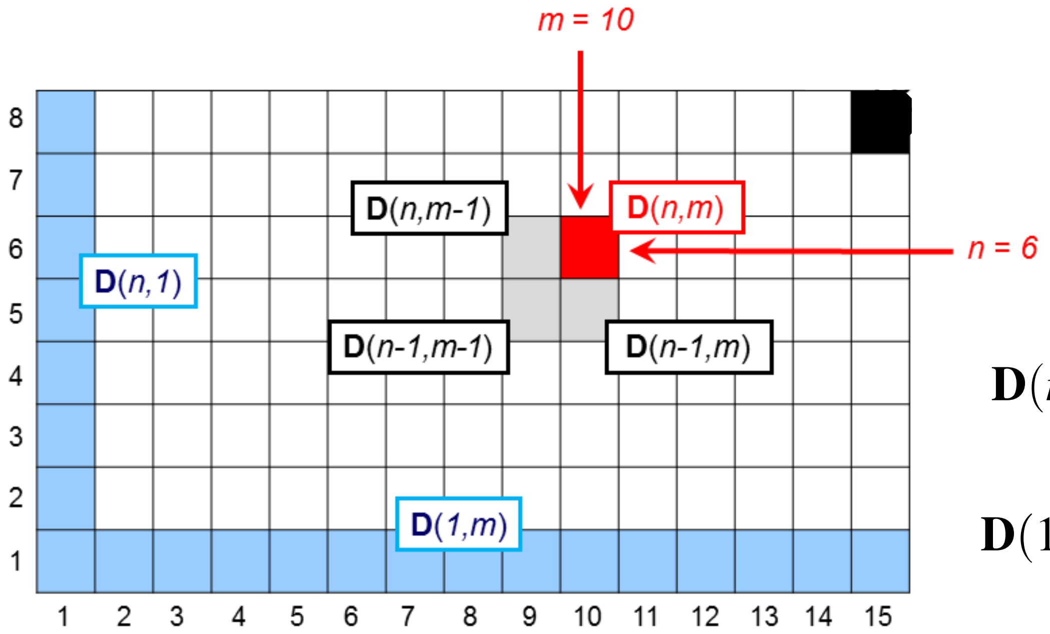
Euclidean distance

Cosine “distance”

Audio Alignment and Matching

Dynamic Time Warping

Step2: Calculate the accumulated distance matrix $\mathbf{D} \in R^{M\times N}$



$$\mathbf{D}(n,1) = \sum_{k=1}^{n} \mathbf{C}(k,1) \quad \text{for} \quad n \in [1:N],$$

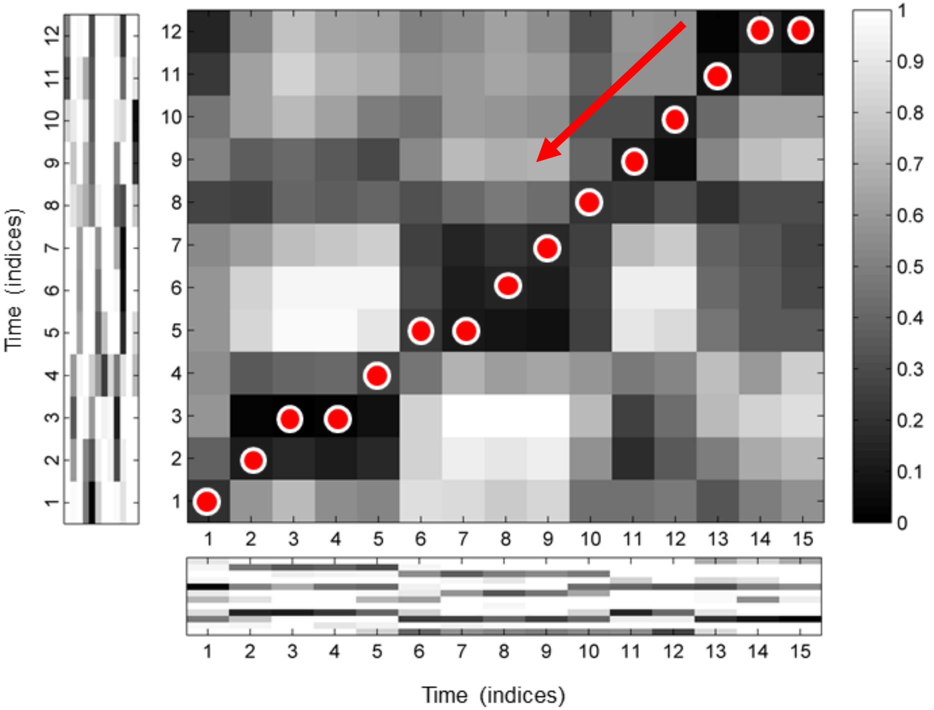
$$\mathbf{D}(1,m) = \sum_{k=1}^{m} \mathbf{C}(1,k) \quad \text{for} \quad m \in [1:M],$$

$$\mathbf{D}(n,m) = \mathbf{C}(n,m) + \min \begin{cases} \mathbf{D}(n-1,m-1) \\ \mathbf{D}(n-1,m) \\ \mathbf{D}(n,m-1) \end{cases}$$

Audio Alignment and Matching

Dynamic Time Warping

Step3: Backward trace the path $\boldsymbol{P}^* = (q_L, q_{L-1}, \dots, q_1)$

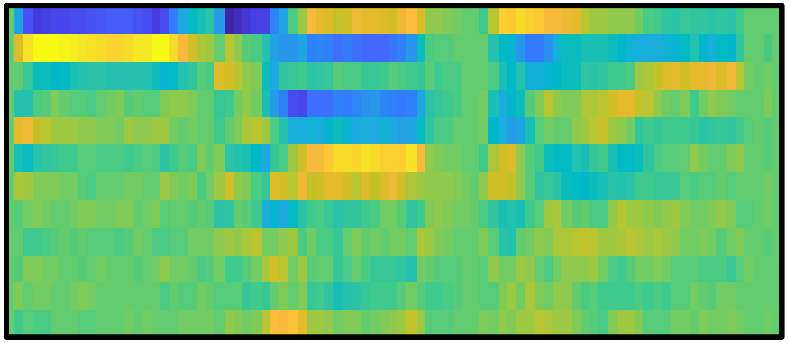


$$
\begin{aligned}
q_1 &= (N, M) \\
q_{\ell+1} &= (1, m-1) \quad \text{if } n = 1, \\
q_{\ell+1} &= (n-1, m) \quad \text{if } m = 1, \\
q_{\ell+1} &= \operatorname{argmin} \begin{cases} \mathbf{D}(n-1, m-1), \\ \mathbf{D}(n-1, m), \\ \mathbf{D}(n, m-1) \end{cases}
\end{aligned}
$$

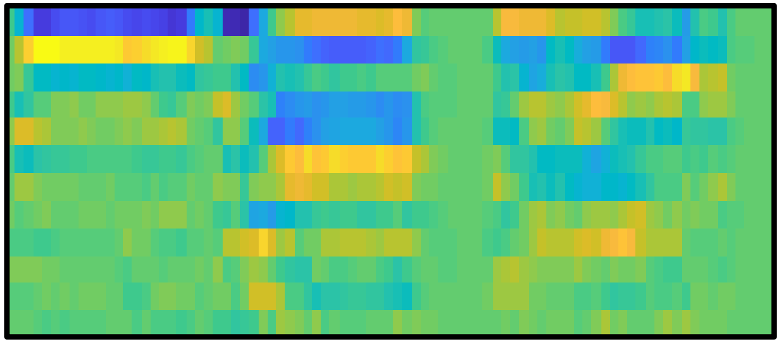
Audio Alignment and Matching

Application: Keyword Matching

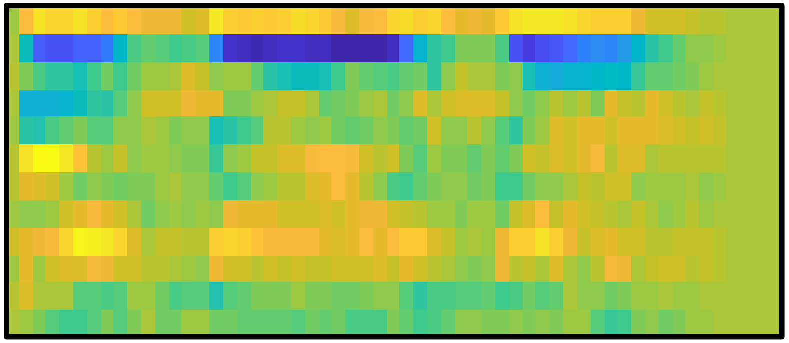
“Strawberry”



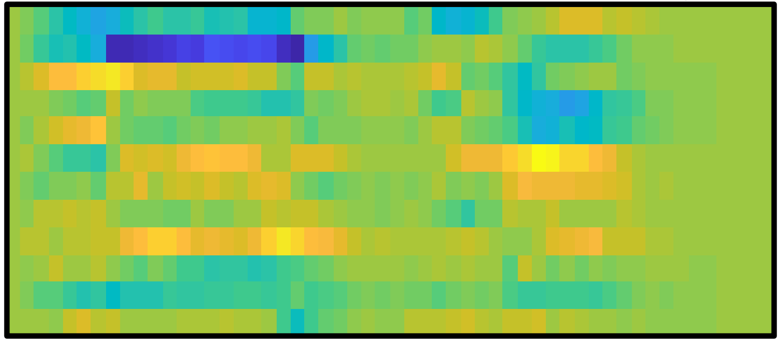
“Strawberry”



“banana”



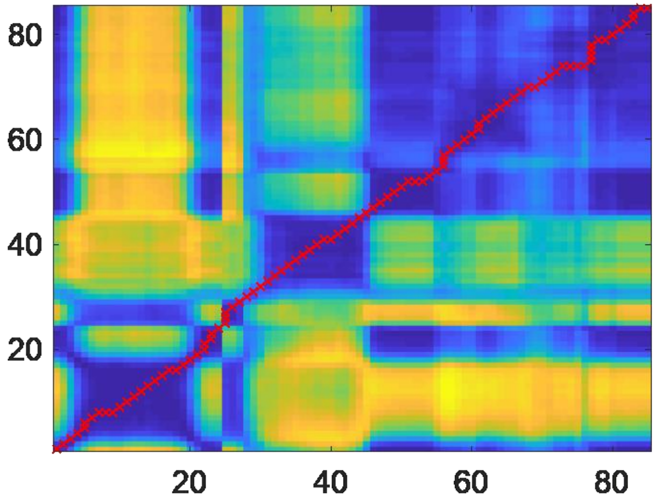
“apple”



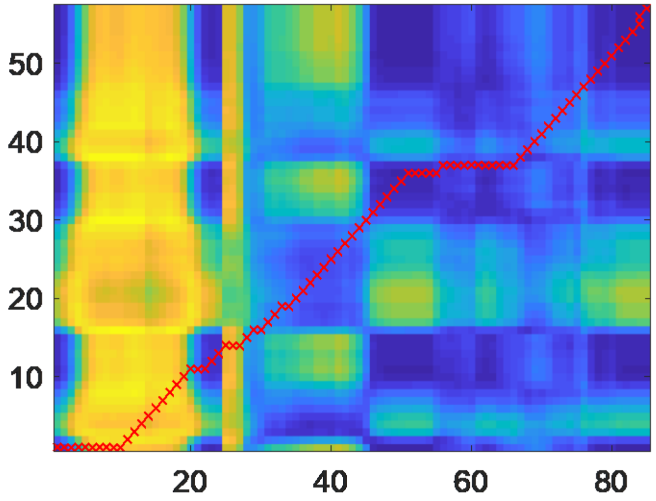
Audio Alignment and Matching

Application: Keyword Matching

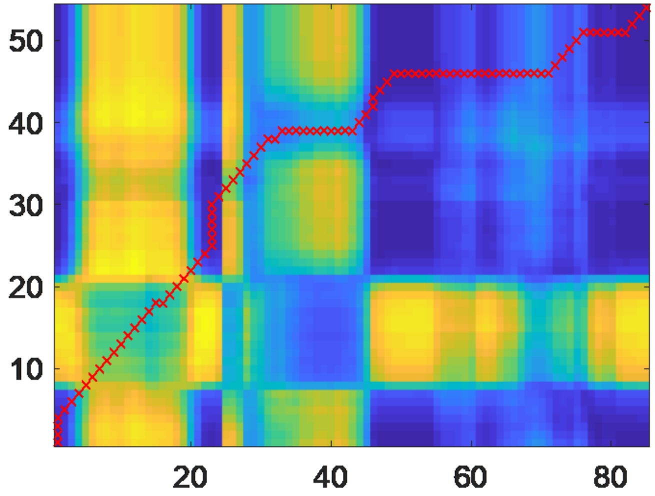
strawberry – strawberry, cost is 1.25



banana – strawberry, cost is 12.72



apple – strawberry, cost is 10.30



Accumulated cost:

The sum of local distance matrix values through the warping path

Outline

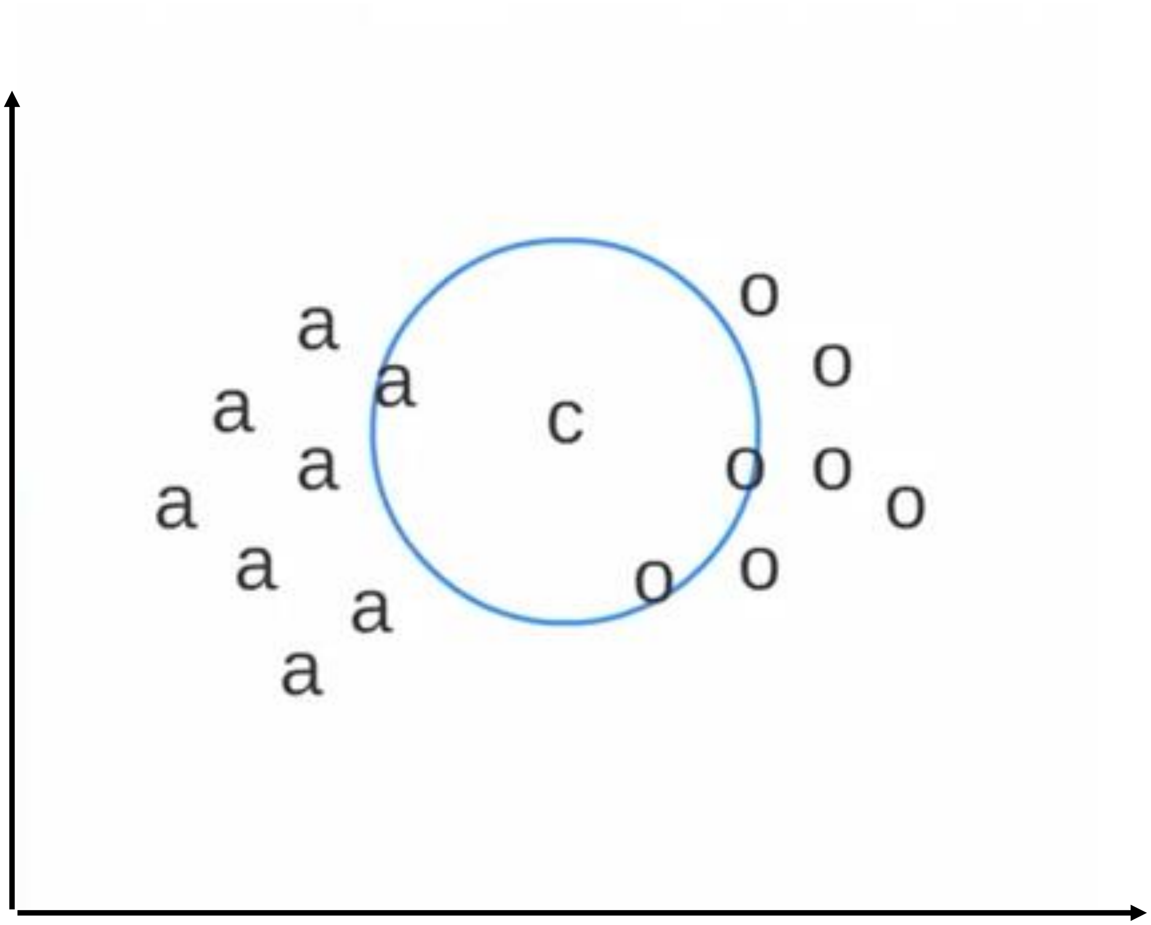
- Introduction
- Audio Feature Extraction
- Audio Alignment and Matching
- **Classifiers**
- Evaluation Measures
- Application 1: Sound Classification
- Application 2: Keyword Spotting

Classifiers

- **K-Nearest Neighbor Classification**
- **Support Vector Machine**
- **Gaussian Mixture Models**
- **Deep Neural Networks**
- **…**

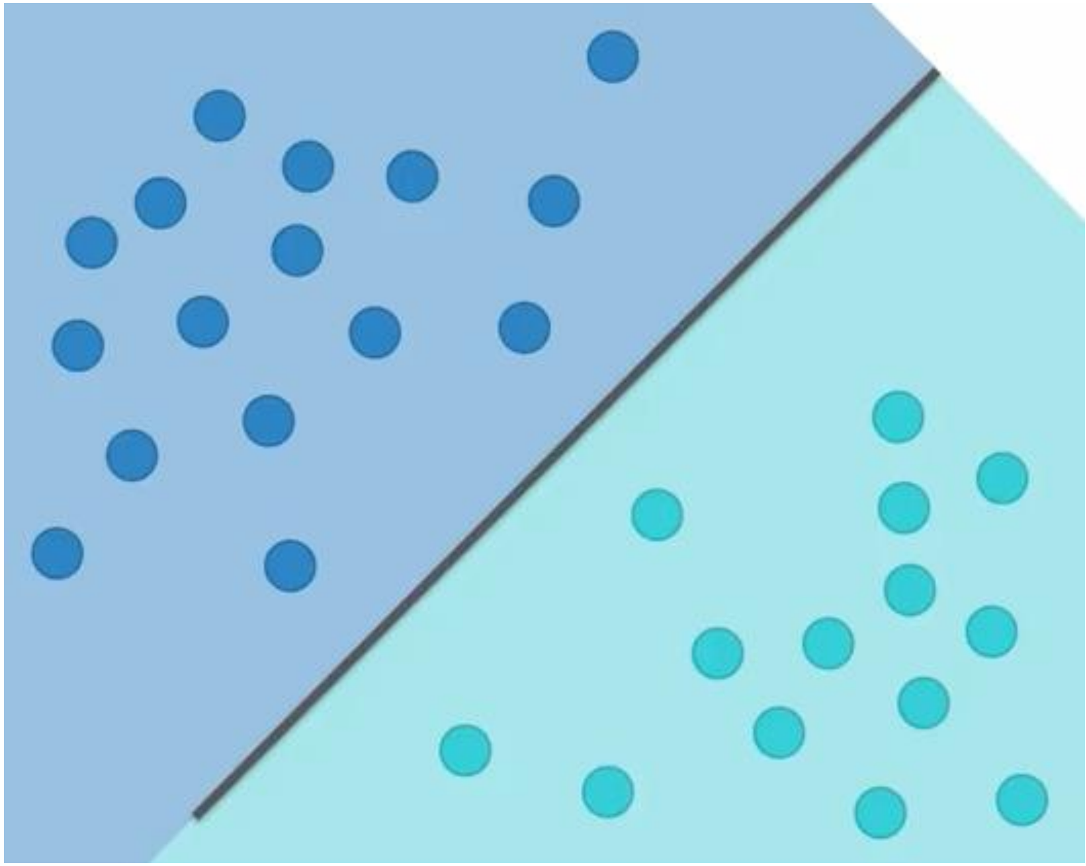
Classifiers

K-Nearest Neighbor Classification



Classifiers

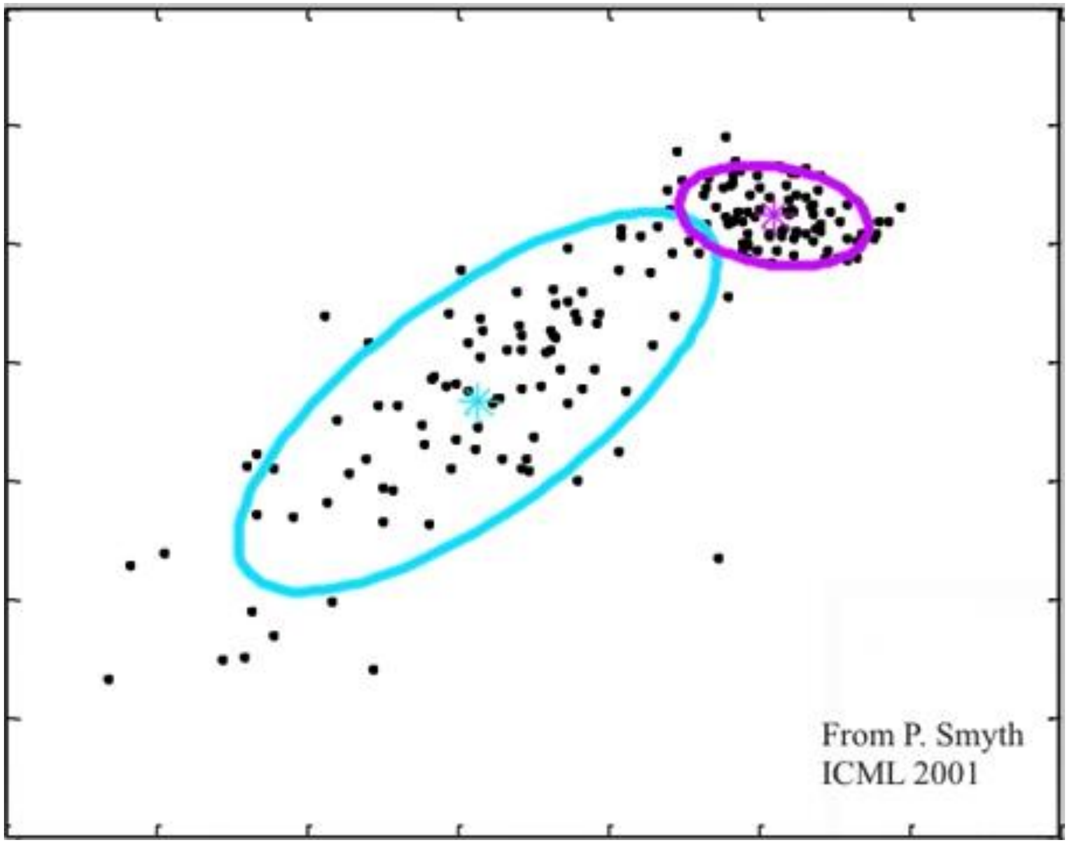
Support Vector Machine



Classifiers

Gaussian Mixture Model

Step 1. Model each class using a mixture of Gaussians with different means, covariance and weights.



From P. Smyth ICML 2001

Step 2. Explain the test data using the GMM model from each class, then choose the class that explains the test data the best.

Outline

- Introduction
- Audio Feature Extraction
- Audio Alignment and Matching
- Classifiers
- **Evaluation Measures**
- Application 1: Sound Classification
- Application 2: Keyword Spotting

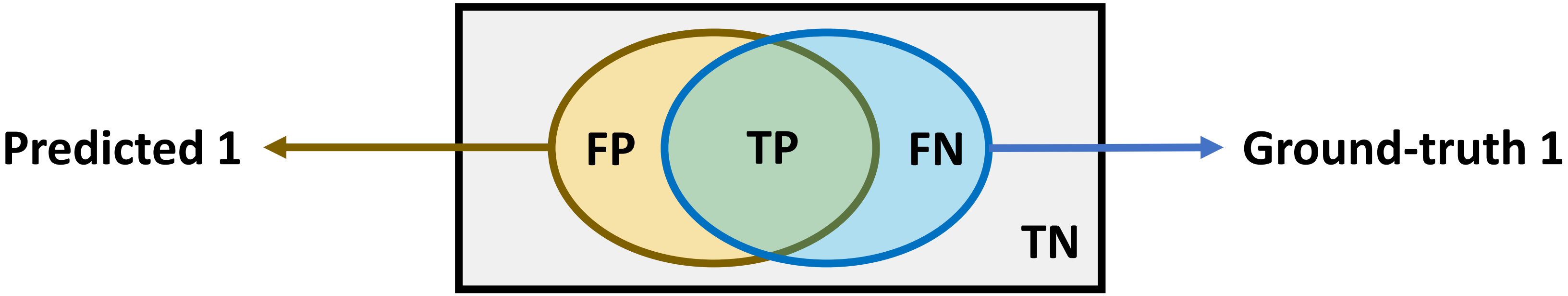
Evaluation Measure

Binary Classification

Model Output $\mathbf{y}$ V.S. **Ground-truth Label $\mathbf{t}$**

Two class labels: 1 and -1

- True Positive (TP): Model predicts 1, ground-truth is 1
- False Positive (FP): Model predicts 1, ground-truth is -1
- True Negative (TN): Model predicts -1, ground-truth is -1
- False Negative (FN): Model predicts -1, ground-truth is 1



Evaluation Measure

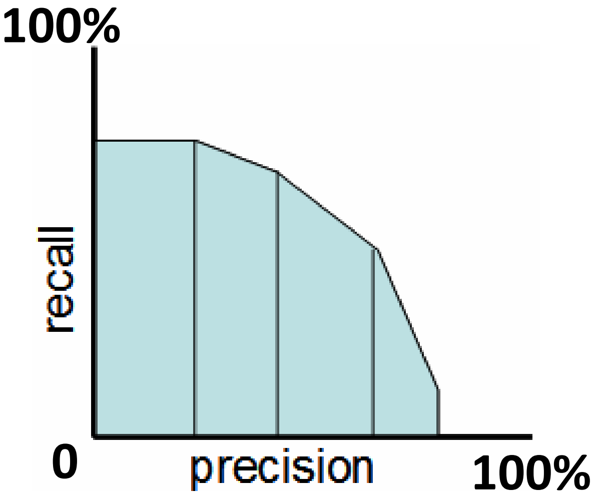
Binary Classification

Model Output $\mathbf{y}$ V.S. **Ground-truth Label** $\mathbf{t}$

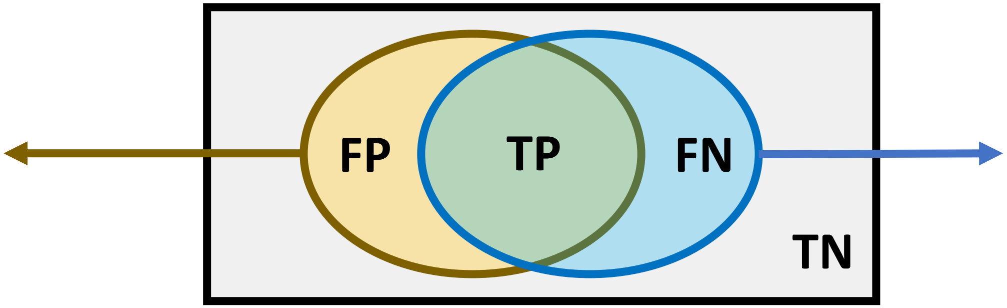
- **Accuracy:** (TP + TN) / (TP+FP + TN+FN)

  = (TP + TN) / (P + N)

- **Precision:** TP / (TP + FP)
- **Recall:** TP / (TP + FN)



Predicted 1



Ground-truth 1

Evaluation Measure

Multi-Class Classification

Model Output $\mathbf{y}$ V.S. **Ground-truth Label $\mathbf{t}$**

Multiple class labels: A, B, C, D, …

- Confusion Matrix

Predicted Label

Ground-truth Label

	A	**B**	**C**	**D**
A	%	%	%	%
B	%	%	%	%
C	%	%	%	%
D	%	%	%	%

Outline

- Introduction
- Audio Feature Extraction
- Audio Alignment and Matching
- Classifiers
- Evaluation Measures
- **Application 1: Sound Classification**
- Application 2: Keyword Spotting

Sound Classification

General process to train and test a classifier

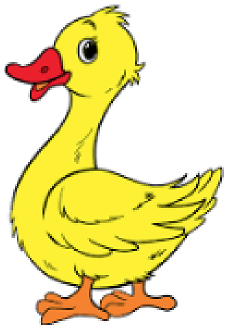
1. Data preparation
   - Divide into training set and test set
   - Feature extraction
   - Annotate the labels
2. Train a classifier on the training set
3. Evaluate the classifier on the test set

Sound Classification

Data Preparation

Dataset:

- Animal sound
- 4 animal categories: cat, dog, sheep, duck
- Each has 15 1-sec recording samples
- 16K sample rate, mono channel
- First 12 samples for training, the other 3 for test



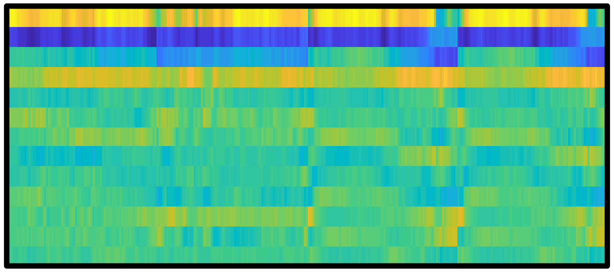
Sound Classification

Data Preparation

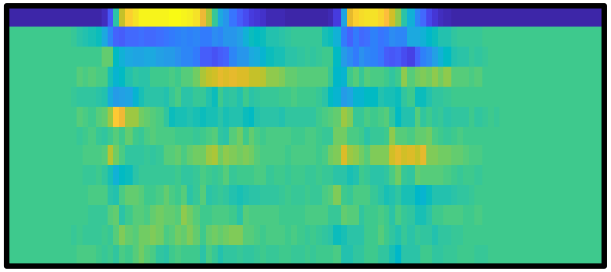
Feature Extraction

- MFCC Feature

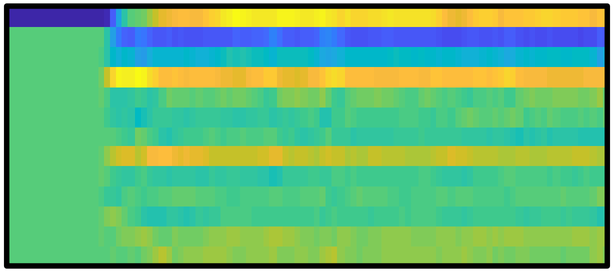
Cat



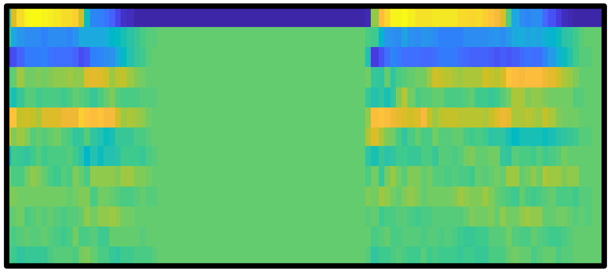
Dog



Sheep



Duck

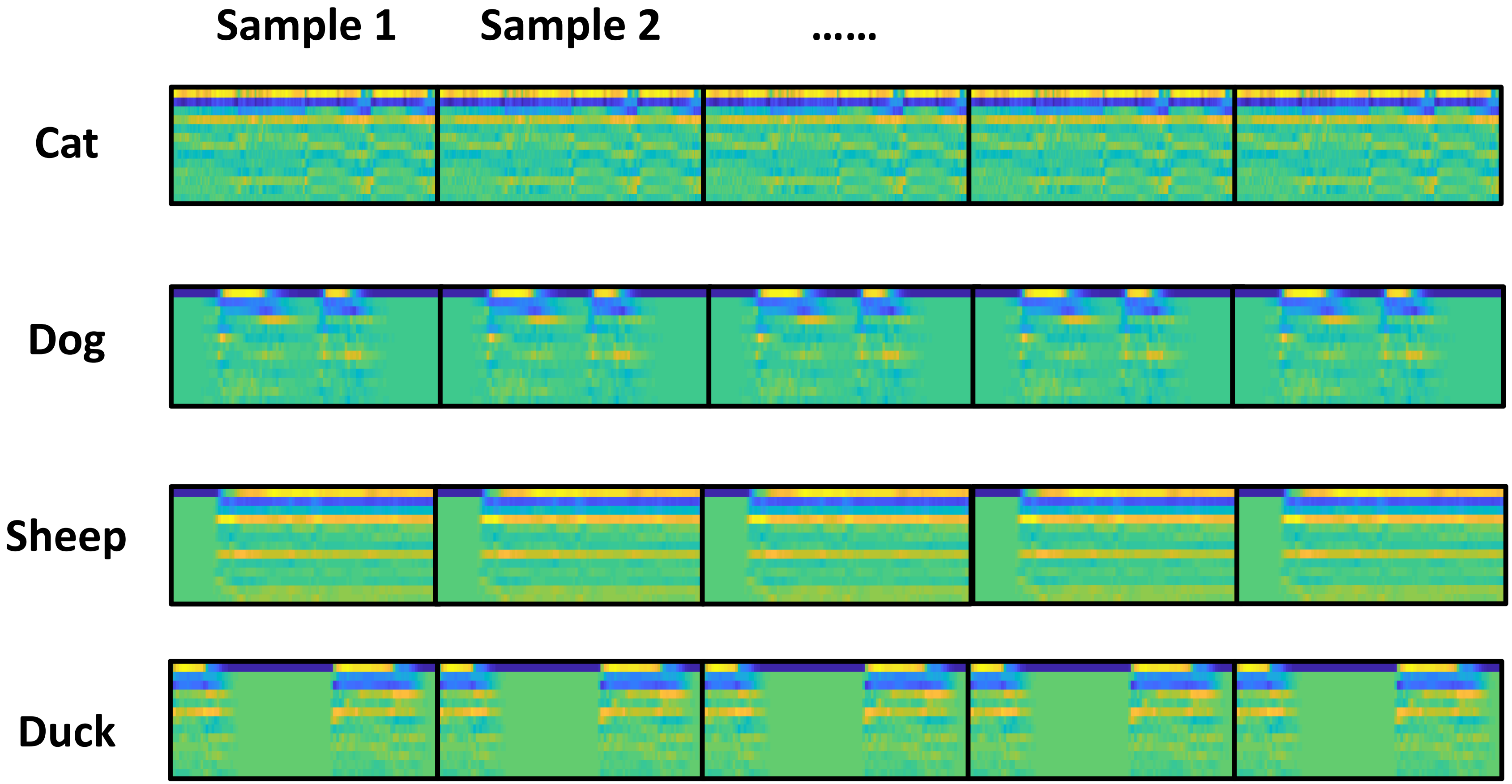


Sound Classification

Data Preparation

Concatenate all of the samples

Tip: Remove low-volume frames

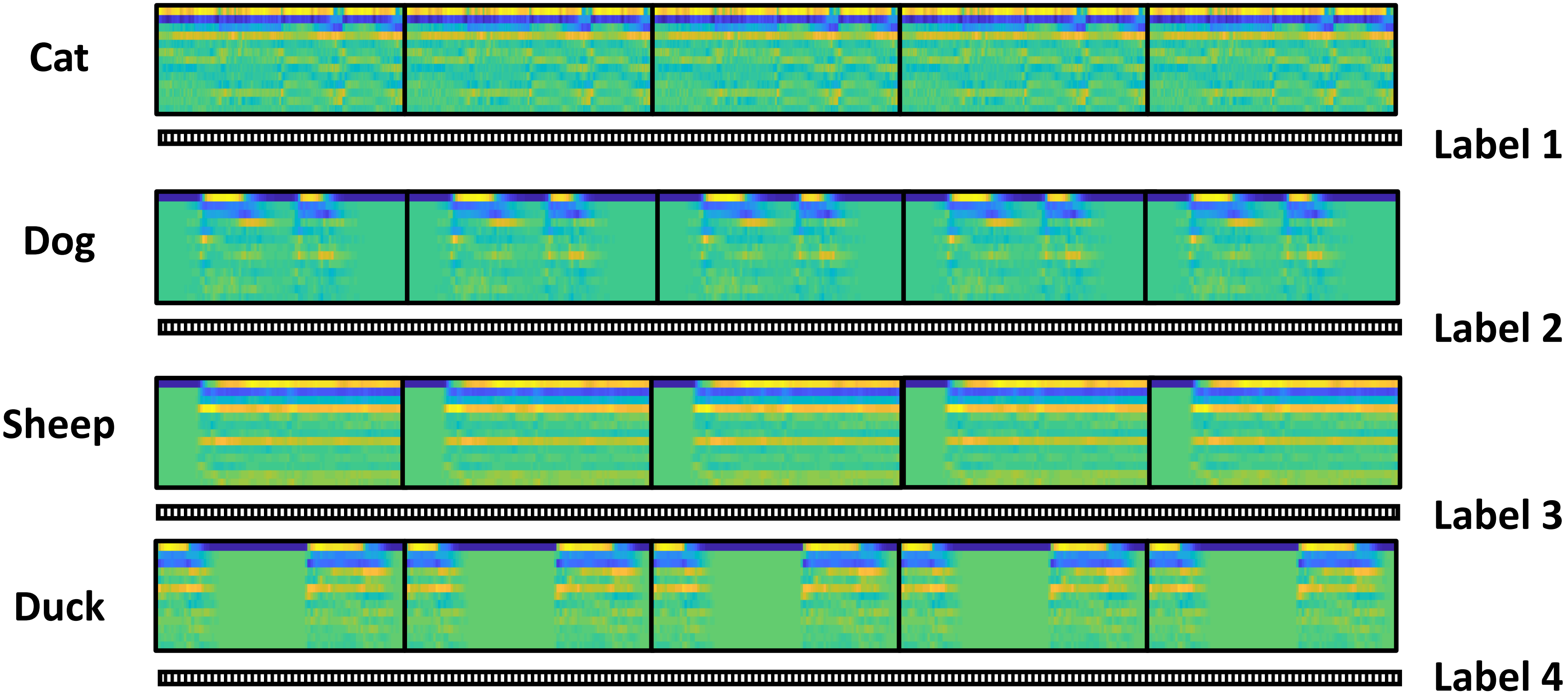


Sound Classification

Data Preparation

Add labels

Tip: Label each frame



Cat

Label 1

Dog

Label 2

Sheep

Label 3

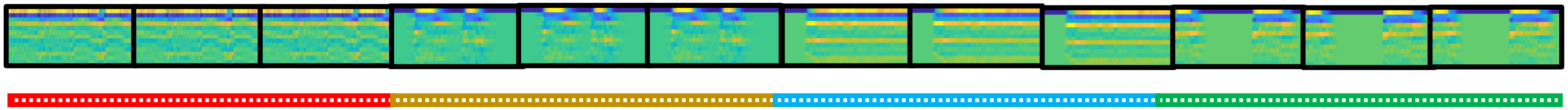
Duck

Label 4

Sound Classification

Train the Classifier

- Feed the concatenated features and labels to the classifier
- Multi-class Support Vector Machine (SVM)
- MATLAB built-in function
- Save the model (classifier parameters)



Sound Classification

Evaluate the Classifier

- Repeat the same data preparation process on the test set
- Load the model
- Feed the concatenated features to the model
- Get the model output and compare with labels
- Evaluate the model using the **confusion matrix**

Sound Classification

Evaluate the Classifier

- Confusion Matrix

Predicted

Ground-truth

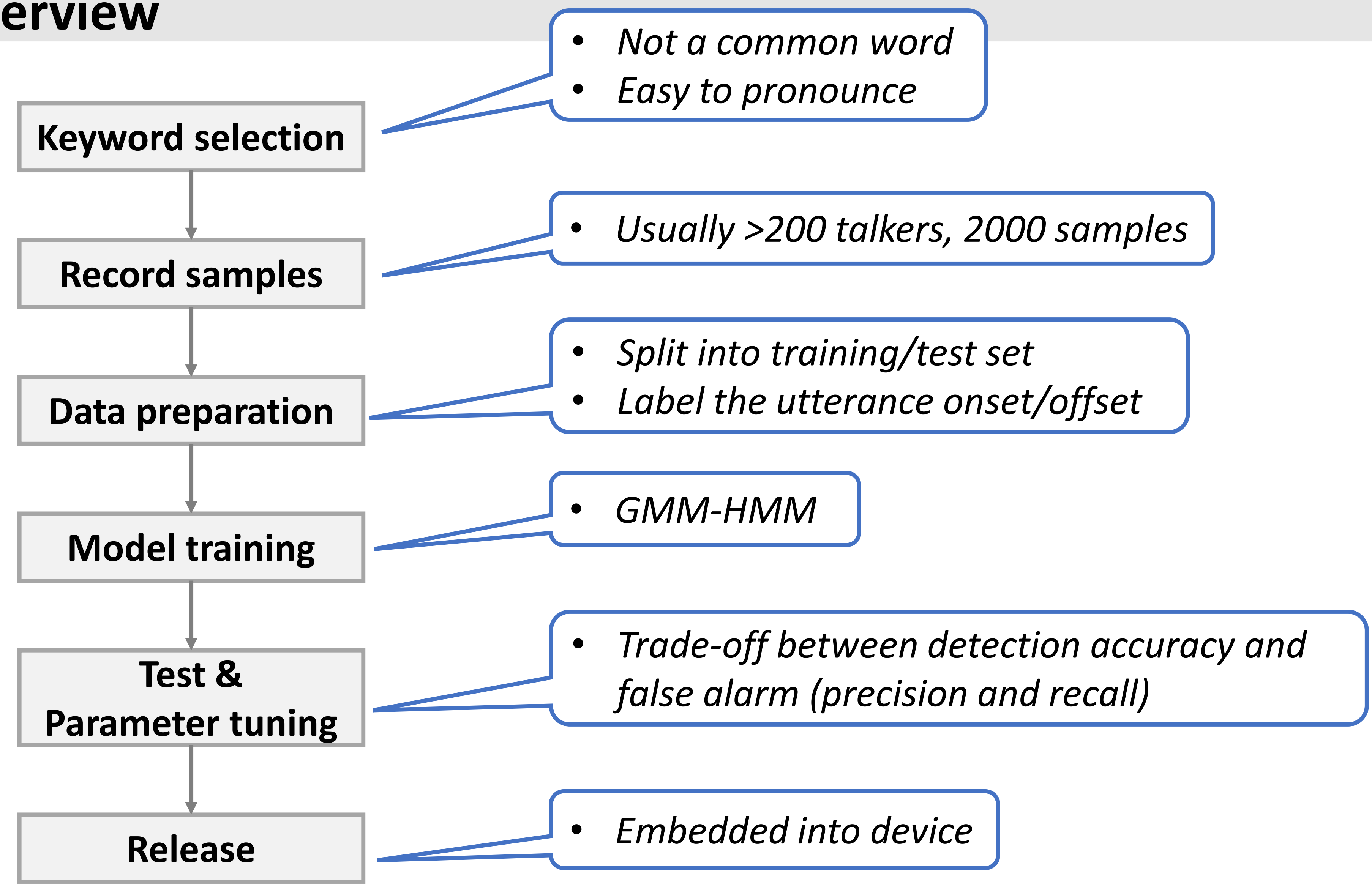
	Cat	Dog	Sheep	Duck
Cat	95.71%	0.00%	4.29%	0.00%
Dog	0.00%	94.20%	0.00%	5.80%
Sheep	7.17%	0.00%	92.83%	0.00%
Duck	4.92%	5.74%	7.38%	81.97%

Outline

- Introduction
- Audio Feature Extraction
- Audio Alignment and Matching
- Classifiers
- Evaluation Measures
- Application 1: Sound Classification
- **Application 2: Keyword Spotting**

Keyword Spotting

Overview



Keyword selection

- *Not a common word*
- *Easy to pronounce*

Record samples

- *Usually >200 talkers, 2000 samples*

Data preparation

- *Split into training/test set*
- *Label the utterance onset/offset*

Model training

- *GMM-HMM*

Test &
Parameter tuning

- *Trade-off between detection accuracy and false alarm (precision and recall)*

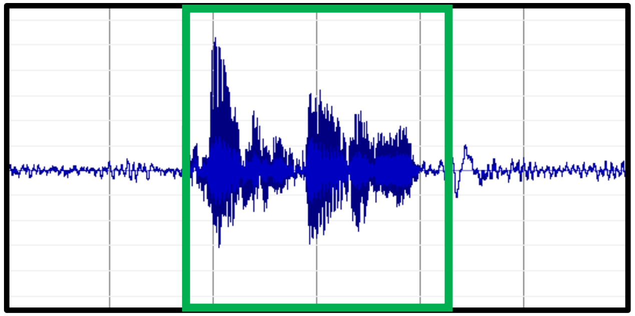
Release

- *Embedded into device*

Keyword Spotting

Data Preparation

1. Collect recording, 16K Hz, mono-channel
2. Label the utterance onset/offset

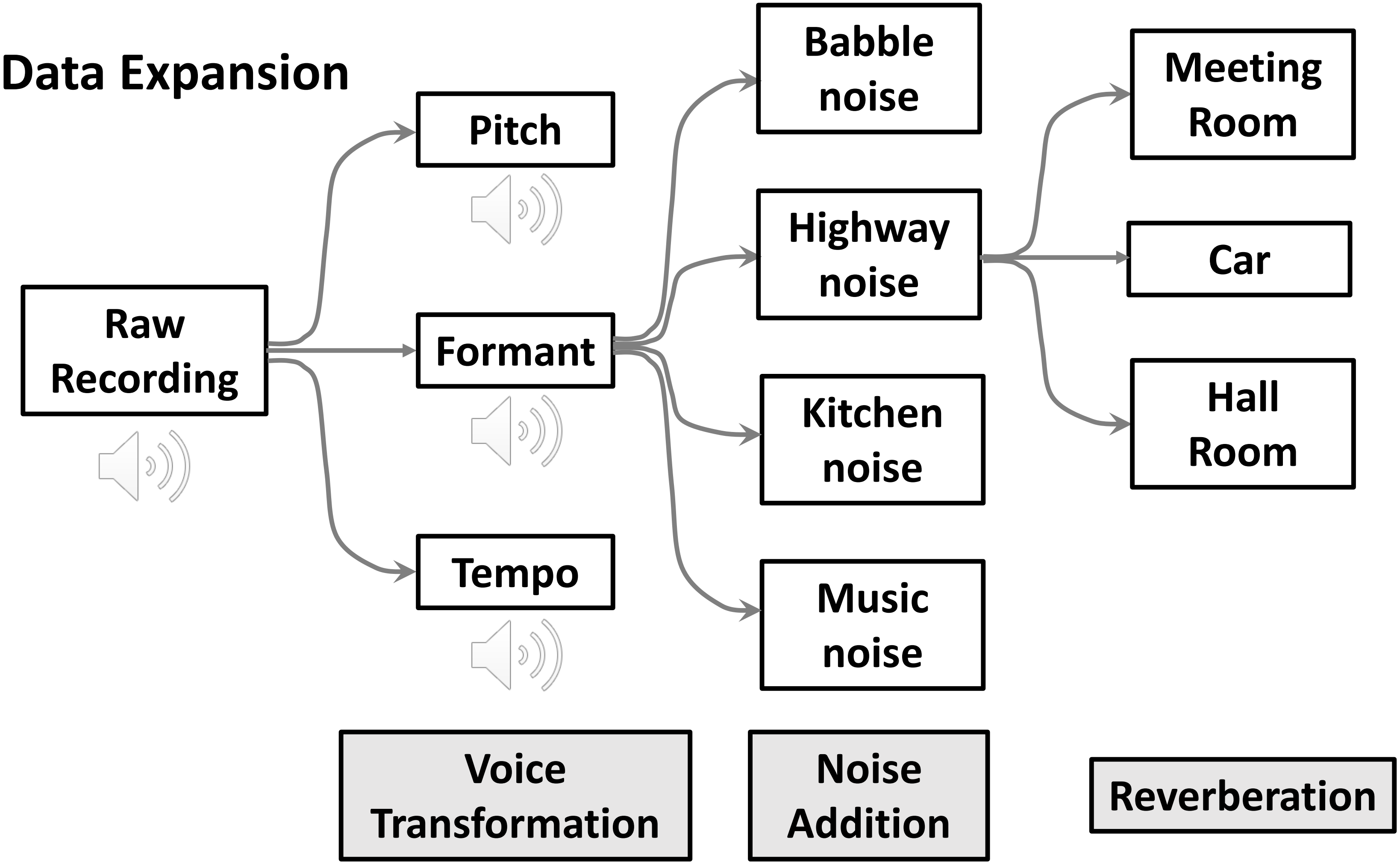


3. Split training/test set
    - Training 85%, test 15%
    - Appropriate ratio for male/female, native/non-native talker
    - No talker overlap in two sets
4. Prepare background data (continuous non-keyword speech)

Keyword Spotting

Data Preparation

Data Expansion



Thousands of samples → Millions of samples

Keyword Spotting

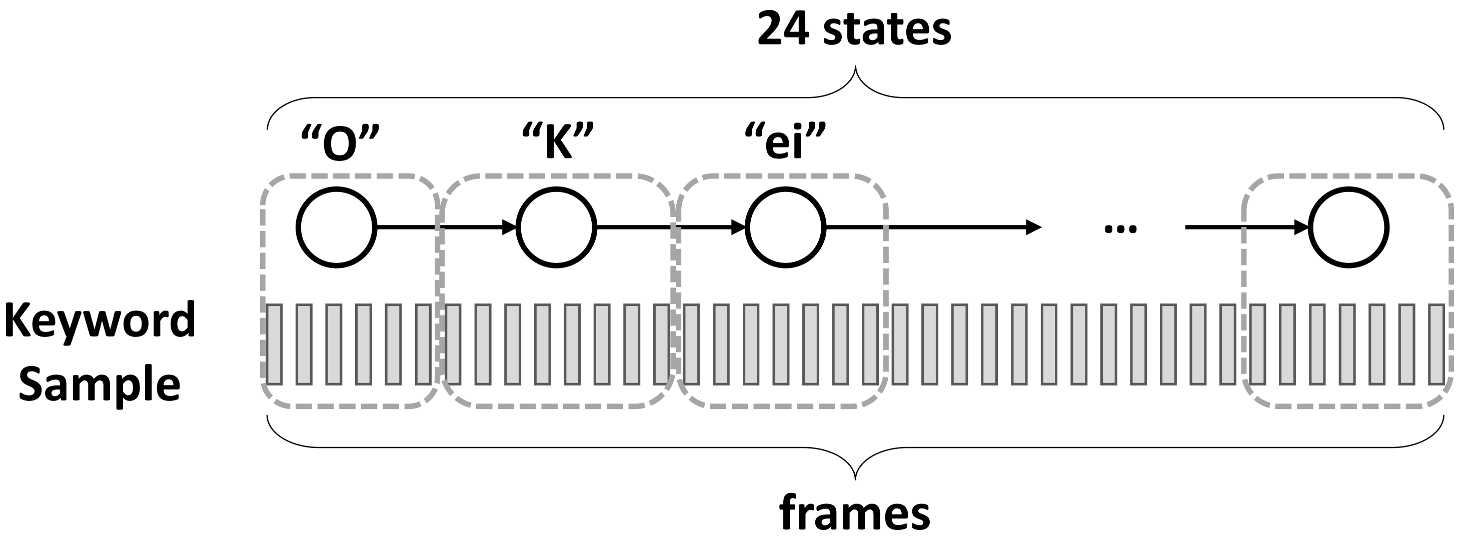
Model Training

Model

- Hidden Markov Model (HMM)
- Gaussian Mixture Model (GMM)

Feature

- MFCC Feature



Keyword Spotting

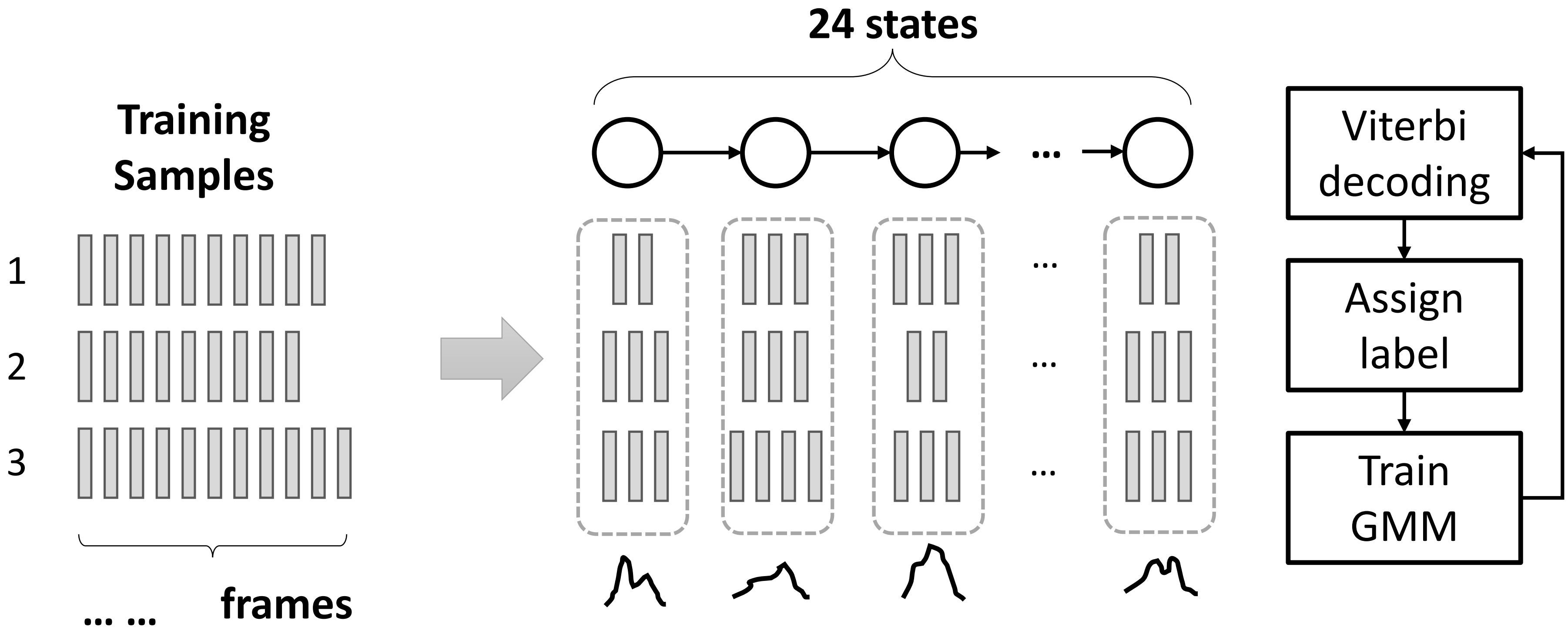
Model Training

Model

- Hidden Markov Model (HMM)
- Gaussian Mixture Model (GMM)

Feature

- MFCC Feature

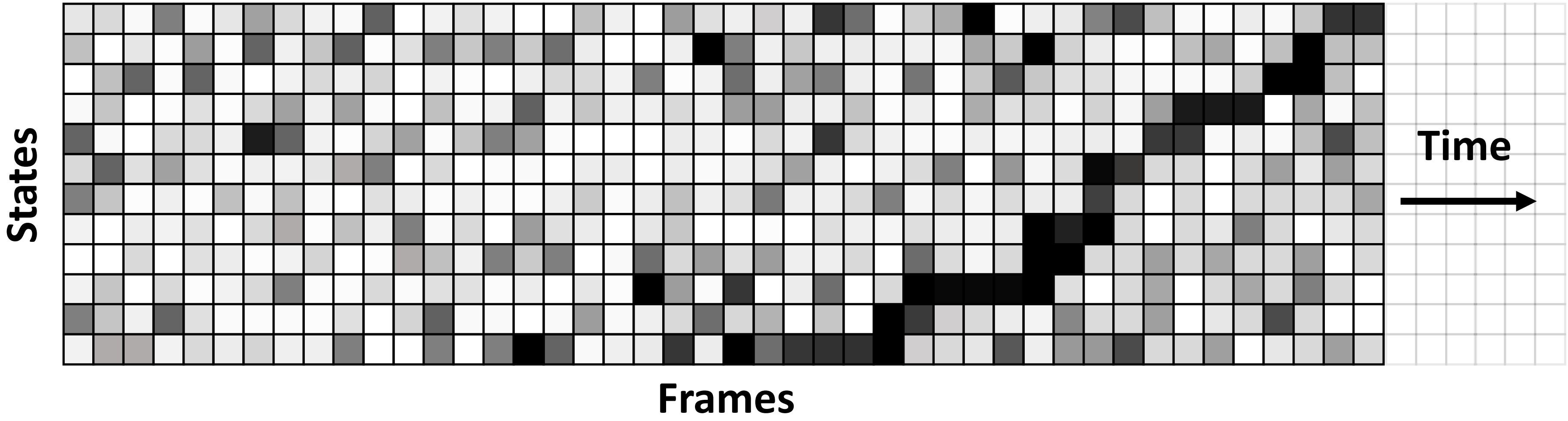


Keyword Spotting

Live Model Test

- Each coming audio stream →MFCC → GMM → State probability
- State probability → Local distance matrix
- Calculate global distance matrix in real-time
- Run **backwarding tracing** in real-time
- Thresholding the **accumulated cost**

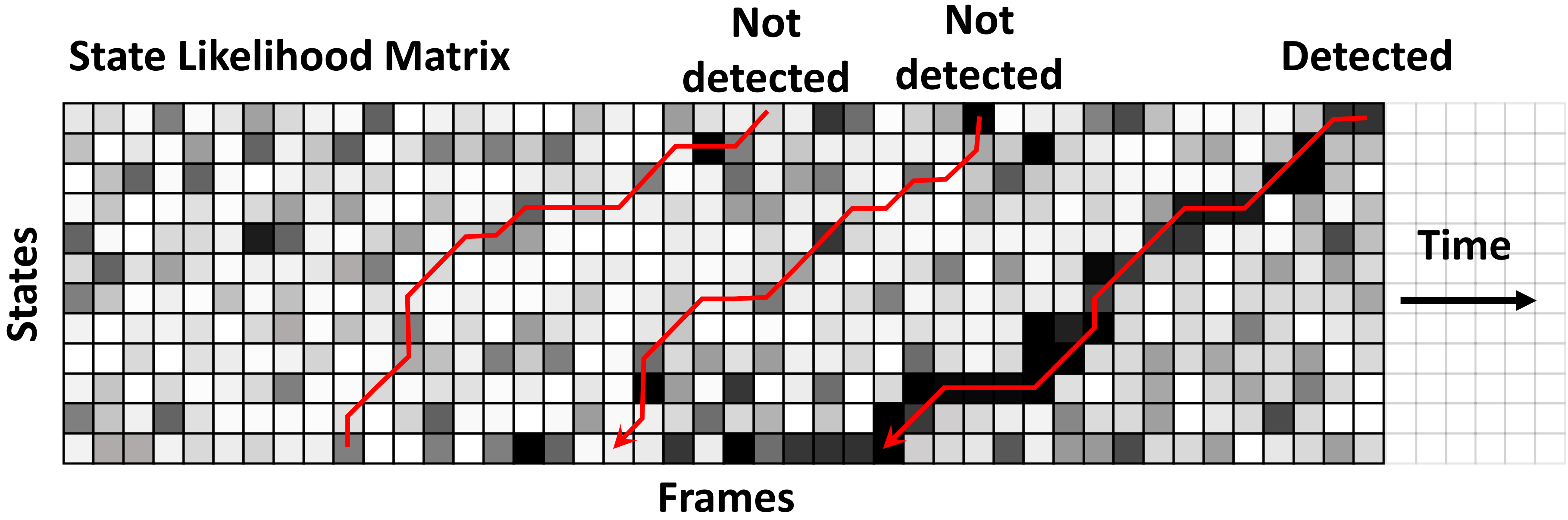
State Likelihood Matrix



Keyword Spotting

Live Model Test

- Each coming audio stream →MFCC → GMM → State probability
- State probability → Local distance matrix
- Calculate global distance matrix in real-time
- Run **backwarding tracing** in real-time
- **Thresholding** the **accumulated cost**



Keyword Spotting

Live Model Test

- Tune the parameter: **Threshold**
- Precision & Recall trade-off
- Threshold ↘, Recall ↗, False Alarm ↗, Precision ↘
- Threshold ↗, Recall ↘, False Alarm ↘, Precision ↗
- Big Regression Test:
  - Test on 72-hour background speech, keep false alarm within 10
  - Fix the threshold, and observe the detection recall on keyword samples

Keyword Spotting

Release the Product

“Alexa”

“OK Google”

“Hi Siri”

“Bixby”

“天猫精灵”

“小艾同学”

“小渡小渡”