Instrument Recognition

Maddy Cappelloni, Autumn Coe

Overview

- Applications and Context
- Methods
  - HMM
  - Source Filter Model
  - Cepstral Coefficients
  - Neural Networks
- Future Research Directions

Applications and human ability

- Label music for search by instrument
- Aid in genre recognition
- Inform music transcription

TABLE I. Summary of percent correct for previous human perception experiments on wind instruments. Results for the oboe, sax, clarinet, and flute are given when possible. The final column is the total number of instruments included in the experiment.

	Date	Oboe	Sax	Clar	Flute	Overall	Number of instruments
Eagleson/Eagleson	1947		59	45	20	56	9
Saldanha/Corso	1964	75		84	61	41	10
Berger	1964					59	10
Clark/Milner	1964					90	3 (flute, clar, oboe)
Strong/Clark	1967a					85	8
Campbell/Heller	1978					72	6 (2-note legato)
Kendall	1986					84	3 (trumpet, clar, violin)
Brown	1999	85	92			89	2 (oboe, sax)
Martin	1999					46	27 (isolated tone)
						67	27 (10-s excerpt)
Houix/McAdams/Brown		87	87	71	93	85	4 (oboe, sax, clar, flute)

Brown et al, 2001

Hidden Markov Models

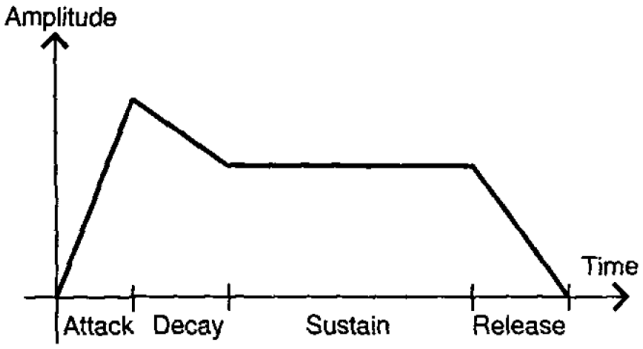
- Test which instrument-specific HMM best explains observations

$$\hat{i} = \underset{i}{\operatorname{argmax}}\, P(Y|\lambda_i)$$

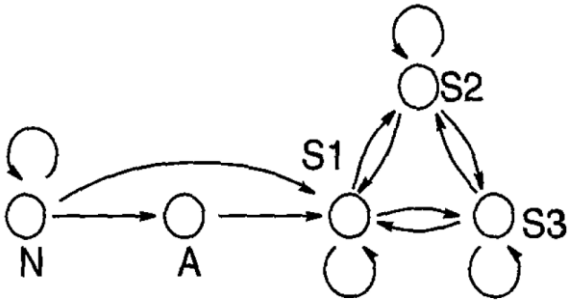
where i is the instrument, λ_i are the parameters for each instrument, and Y are the observations

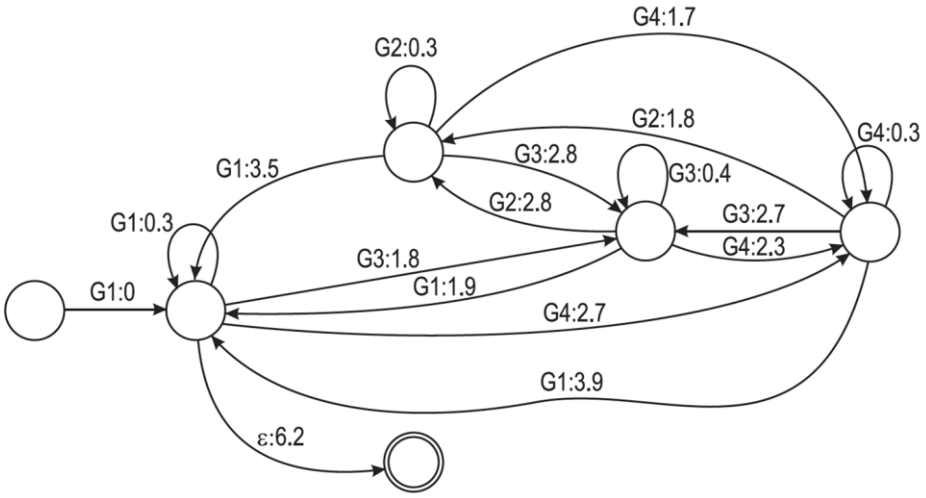
- Choose model states
- Choose audio features

Model states



- **Attack, Decay, Sustain, Release**
  - Relative timing depends on the instrument
  - Lee et al, 2002
- Ergodic model
  - N: silence
  - A: attack
  - S1, S2, S3: vibrato
  - Lee et al, 2002
- Arbitrary Topology
  - Eichner et al, 2006





Audio features

Generally related to timbre of the instrument

- Amplitude of harmonics
  - Lee et al, 2002
- Interpolated spectral envelope
  - Lee et al, 2002
- Cepstral coefficients and their derivatives
  - Lee et al, 2002, Eronen, 2003, Eichner et al, 2006

Pros/Cons of HMM

Pros

- Performance approaches human capabilities

Cons

- Performance drops quickly with increasing number of instruments
- Observations may be very similar across instruments
  - Use ICA to maximize statistical independence
  - Discriminative training (maximum mutual information)
  - Eronen et al, 2006

Source-Filter Model

- **Each instrument has its own spectral envelope shape and can be modeled as a note source (excitation spectrum) with a filter specific to the instrument.**

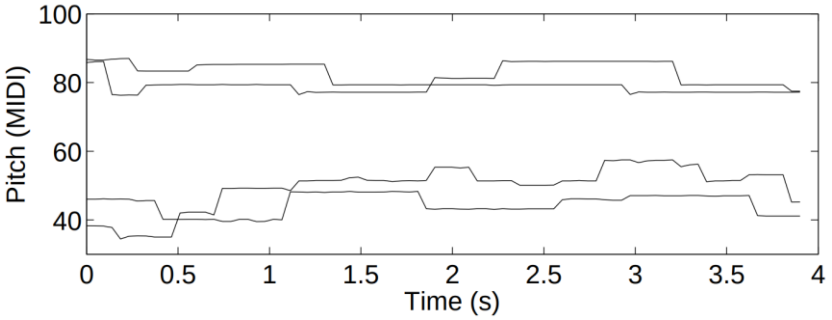
$$\hat{x}_t(k) = \sum_{n=1}^{N} \sum_{i=1}^{I} g_{n,i,t} e_{n,t}(k) h_i(k)$$

- $\hat{x}_t(k)$: Magnitude of STFT for frame t
- n: Notes
- i: Instruments
- $g_{n,i,t}$: Gain of each instrument playing each note in the frame
- $e_{n,t}(k)$: Excitation spectra for notes
- $h_i(k)$: Filters for instruments (linear combination of elementary responses)

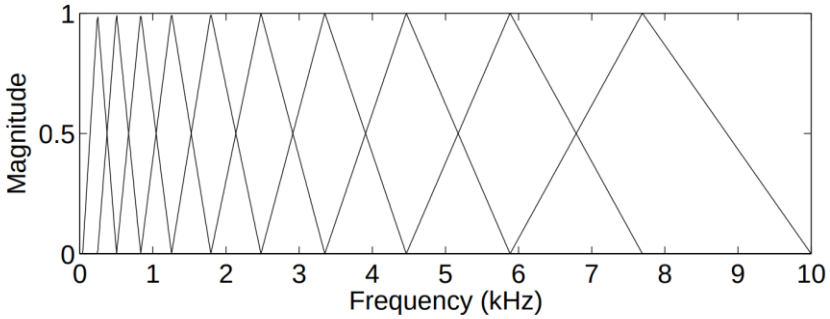
Heittola et al, 2009

Fitting the model

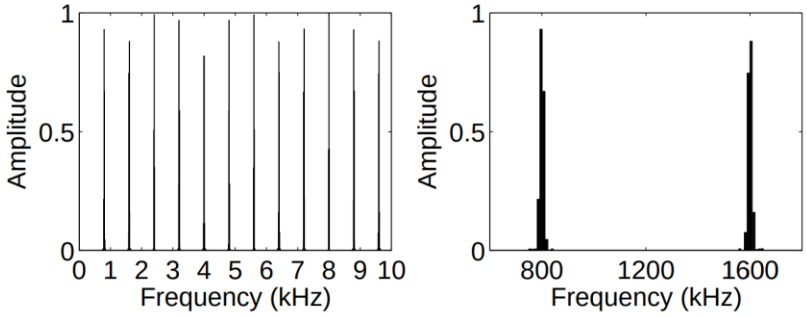
- Do multipitch estimation



- Construct filter responses as a *sum* of triangular bandpass filters with instrument specific weights



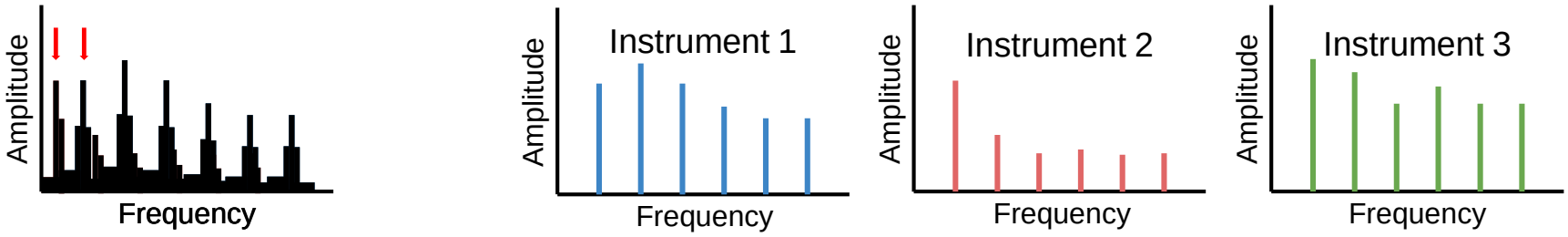
- Excitation spectra are harmonic combs with amplitude 1



Heittola et al, 2009

Fitting the model

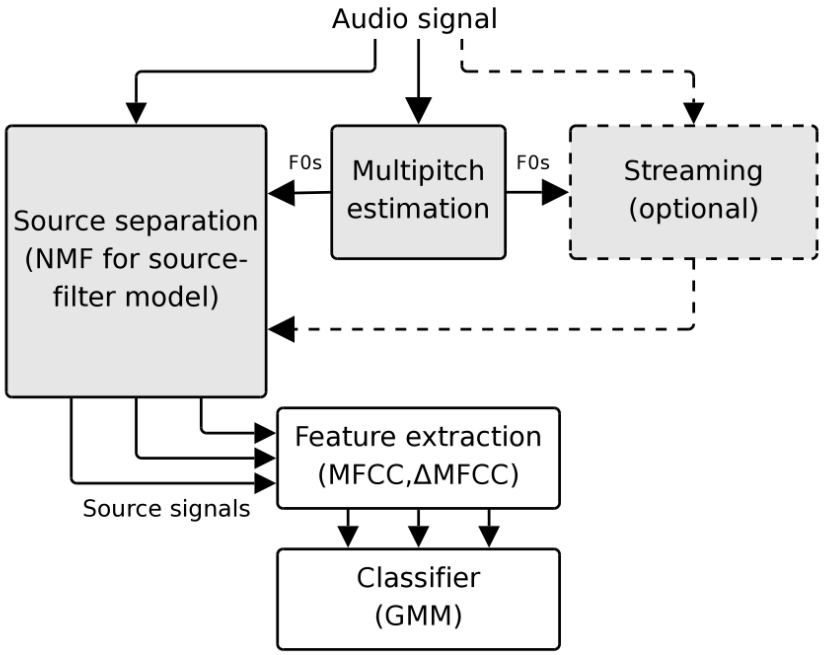
- Fit filter weights and gains for each instrument/note combination by doing NMF with multiplicative update
- For each frame
  - Choose an excitation spectrum based on the multipitch estimation
  - Update filter weights
  - Update gains
  - Iterate until reconstruction error sufficiently small



Heittola et al, 2009

Classification

- Test likelihood of signal for each instrument class
  - Likelihoods accumulate over the frames
  - Weight by energy (RMS of signal)
  - Classification by maximum likelihood
- Calculate F-score to evaluate
  - Weighted harmonic mean of precision and recall (more detailed than reporting just accuracy)



	Polyphony					
	1	2	3	4	5	6
no separation	62.0	18.7	12.1	13.7	24.4	29.5
no streaming	62.0	49.5	42.1	42.6	39.3	42.7
streaming (given F0s)	62.0	59.0	58.0	57.9	57.8	56.0
streaming (est. F0s)	61.0	60.2	53.5	56.7	55.2	53.8
streaming (timbre)	62.0	57.6	51.9	57.0	55.9	59.1

Heittola et al, 2009

Pros/Cons of Source-Filter Model

Pros

- Performance is not significantly reduced for polyphonic music

Cons

- Will not work well if multiple instruments are playing the same note
- Assumes all instruments are playing in every frame
- Assumes harmonicity

Cepstral Coefficients

- Cepstrum: Inverse Discrete Fourier Transform (IDFT) of the log magnitude of the Fourier transform of the signal

$$\mathcal{C}_{\log}(\mathbf{x}) = \mathcal{F}^{-1}(\log|\mathcal{F}(\mathbf{x})|)$$

- Can use a power transformation instead of a log. (Yu et al, 2014)

$$\mathcal{C}(\mathbf{x}) = \mathcal{F}^{-1}(g(\mathcal{F}(\mathbf{x})))$$

$$g(\mathbf{x}) = |\mathbf{x}|^{1/2}, |\mathbf{x}|^{1/3}, |\mathbf{x}|^{1/4}, |\mathbf{x}|^{1/5}$$

- Can use a Discrete Cosine Transform (DCT) instead of inverse Fourier transform

Sparse Cepstral Codes

- Identifies predominant instrument
- Input Cepstrum is calculated using a log function, power functions, or MFCC.
- Dictionary is trained using a joint optimization problem

$$\hat{\mathbf{D}} = \arg\min_{\mathbf{D}} \frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{2} \|\mathbf{y}_i - \mathbf{D}\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \right)$$

$\hat{\mathbf{D}}$: Dictionary

$\mathbf{y}_i$: Input (Cepstrum training corpus)

$\lambda \|\alpha_i\|_1$: Sparse representation

$\mathbf{d}_j \in \mathbb{R}^m, j \in \{1, k\}$

$\mathbf{d}_j^T \mathbf{d}_j \leq 1.$

$\mathbf{d}_j$: Dictionary atom

$\alpha \in \mathbb{R}^k$

$\mathbf{y} \in \mathbb{R}^m$

$\mathbf{D} \in \mathbb{R}^{m \times k}$

- Sparse representation is calculated using trained dictionary and Least Angle Regression LASSO algorithm

$$\hat{\alpha} = \arg\min_{\alpha} \|\mathbf{y} - \mathbf{D}\alpha\|_2^2 + \lambda \|\alpha\|_1$$

Yu et al, 2014

Sparse Cepstral Codes Continued

- The more granular the dictionary, the better the result
- Power transforms perform better for polyphony than the traditional log transform

k	$f(\mathcal{F})$	$f(\mathcal{C}_{\log})$	$f(\mathcal{C}_2)$	$f(\mathcal{C}_3)$	$f(\mathcal{C}_4)$	$f(\mathcal{C}_5)$
64	0.818	0.833	0.910	0.892	0.919	0.916
128	0.873	0.875	0.926	0.924	0.918	0.921
256	0.902	0.920	0.936	0.935	0.936	0.938
512	0.907	0.926	0.947	0.942	0.944	0.947
1024	0.928	0.944	0.939	0.939	0.946	0.955
$\triangle$	0.110	0.111	0.029	0.047	0.027	0.039

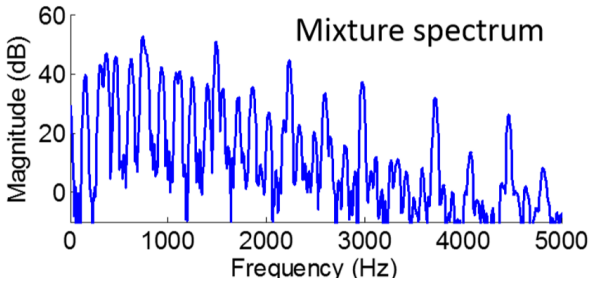
Task	existing work		low-level feature			SC feature					
	[17]	[18]	MFCC	$\mathcal{F}$	$\mathcal{C}_{\log}$	$f(\mathcal{F})$	$f(\mathcal{C}_{\log})$	$f(\mathcal{C}_2)$	$f(\mathcal{C}_3)$	$f(\mathcal{C}_4)$	$f(\mathcal{C}_5)$
Uni-source (ParisTech)	0.820	–	0.541	0.684	0.705	0.934	0.945	0.941	0.939	0.948	0.957
Multi-source (MTG)	–	0.630	0.229	0.256	0.276	0.611	0.610	0.603	0.637	0.649	0.661

Yu et al, 2014

Uniform Discrete Cepstrum

$$\mathbf{a} = \mathbf{M}\mathbf{c}$$

$$\mathbf{c}_{\text{oc}} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{a} = \frac{1}{N}\mathbf{M}^T\mathbf{a}$$

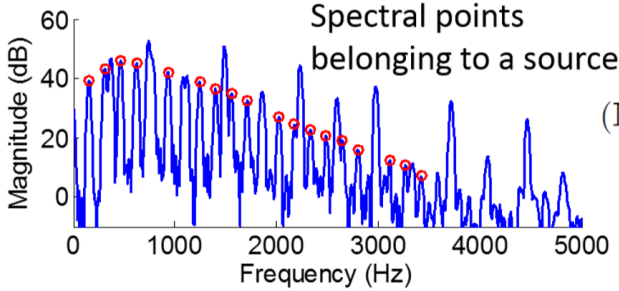


$\frac{1}{N}\mathbf{M}^T$ (Discrete Cosine Transform)

Ordinary Cepstrum (OC) of mixture signal

$$\hat{\mathbf{a}} = \hat{\mathbf{M}}\mathbf{c}$$

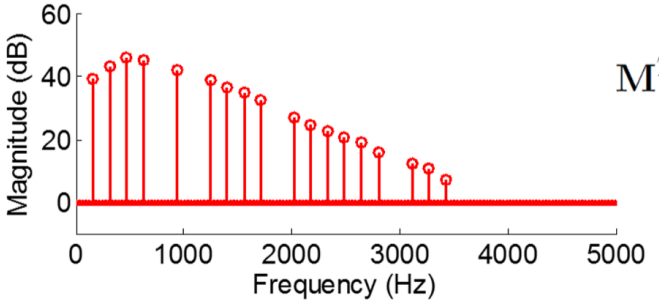
$$\mathbf{c}_{\text{dc}} = (\hat{\mathbf{M}}^T\hat{\mathbf{M}})^{-1}\hat{\mathbf{M}}^T\hat{\mathbf{a}}$$



$(\hat{\mathbf{M}}^T\hat{\mathbf{M}})^{-1}\hat{\mathbf{M}}^T$

Discrete Cepstrum (DC) of the source

$$\mathbf{c}_{\text{udc}} = \hat{\mathbf{M}}^T\hat{\mathbf{a}}$$
$$= \mathbf{M}^T\tilde{\mathbf{a}} = N(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\tilde{\mathbf{a}}$$



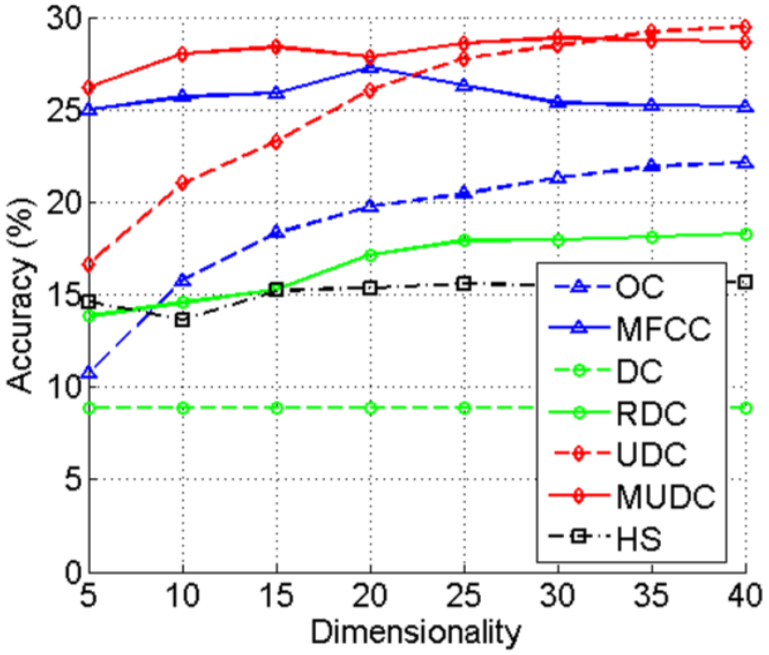
$\mathbf{M}^T$ (Discrete Cosine Transform)

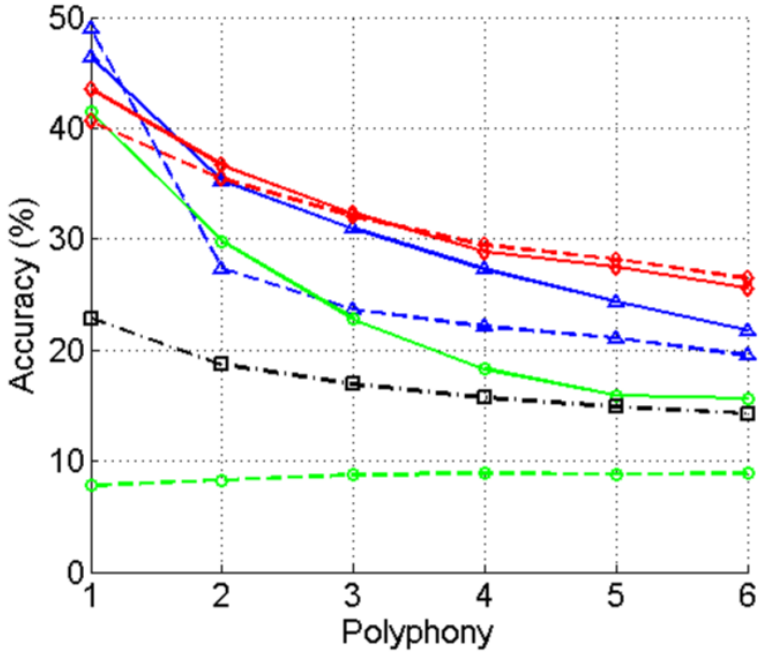
UDC of the source

Duan et al, 2014

Unified Discrete Cepstrum Continued

- 13-Class SVM trained on 5 single frames of single notes from 13 instruments
- Tested on randomly generated chords from notes in a different data set





Duan et al, 2014

Pros/Cons of Cepstral Coefficients

Pros

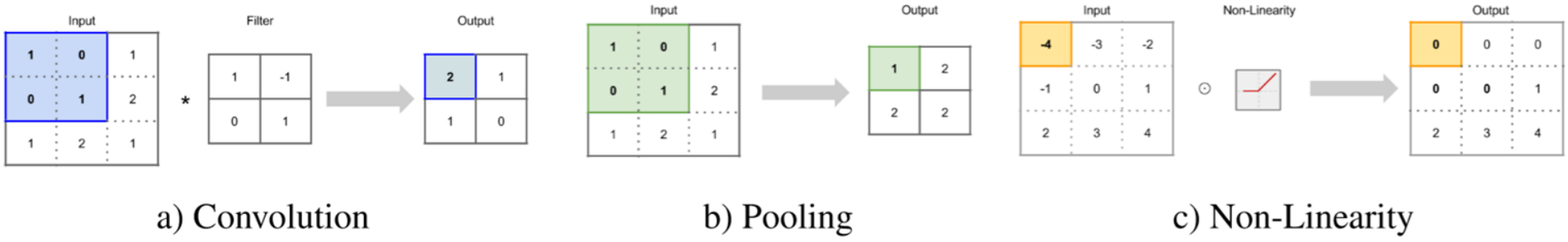
- Can be used on polyphony without source separation

Cons

- Does not work well for unison or octave notes
- Requires isolated instrument note training data

Neural Networks

- Convolutional Neural Networks (CNNs) can be seen as a trainable feature extractor coupled with a learning model. (Li et al, 2015)
- Feature identification is done by the neural network instead of the designing engineer.
- No longer requires musical domain knowledge



a) Convolution b) Pooling c) Non-Linearity

Comparing Two CNNs

Li et al, 2014

- Input is raw audio
- 11 instrument classes
- Trained on audio clips with multiple instruments

Parameter	
Convolution Feature Maps 1	256
Convolution Filter Size 1	3101
Maxpooling Stride Size 1	20
Maxpooling Size 1	40
Convolution Feature Maps 2	384
Convolution Filter Size 2	300
Maxpooling Size 2	30
Maxpooling Stride Size 2	20
Convolution Feature Maps 3	384
Convolution Filter Size 3	20
Maxpooling Size 3	8
Maxpooling Stride Size 3	4
Layer 4 Output Size	400
Final Output Size	11

Han et al, 2014

- Input is mel spectrogram
- 11 instruments
- Trained on audio clips chosen to contain a continuous presence of a single predominant instrument

Input size	Description
$1 \times 43 \times 128$	mel-spectrogram
$32 \times 45 \times 130$	3×3 convolution, 32 filters
$32 \times 47 \times 132$	3×3 convolution, 32 filters
$32 \times 15 \times 44$	3×3 max-pooling
$32 \times 15 \times 44$	dropout (0.25)
$64 \times 17 \times 46$	3×3 convolution, 64 filters
$64 \times 19 \times 48$	3×3 convolution, 64 filters
$64 \times 6 \times 16$	3×3 max-pooling
$64 \times 6 \times 16$	dropout (0.25)
$128 \times 8 \times 18$	3×3 convolution, 128 filters
$128 \times 10 \times 20$	3×3 convolution, 128 filters
$128 \times 3 \times 6$	3×3 max-pooling
$128 \times 3 \times 6$	dropout (0.25)
$256 \times 5 \times 8$	3×3 convolution, 256 filters
$256 \times 7 \times 10$	3×3 convolution, 256 filters
$256 \times 1 \times 1$	global max-pooling
1024	flattened and fully connected
1024	dropout (0.50)
11	sigmoid

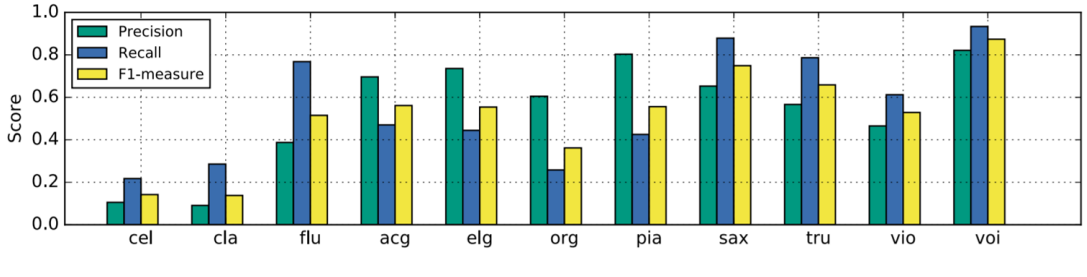
Convolutional Neural Network Results

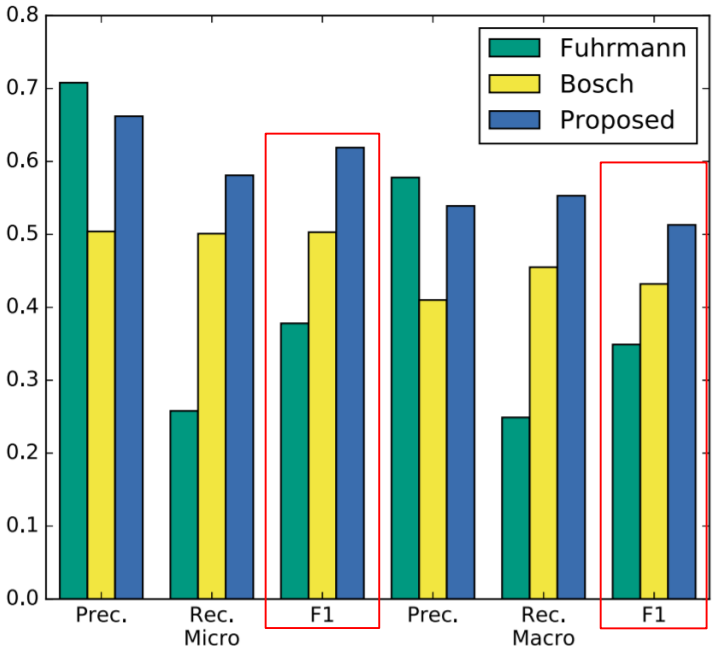
Models	Accuracy	Exact Match	Precision	Recall	F-micro	F -macro
Audio + CNN	**82.74%**	**25.78%**	0.7560	**0.6888**	**0.7208**	**0.6433**

Li et al (above)

Han et al (right)

Han et al single instrument breakdown (below)





Pros/Cons of Neural Networks

Pros

- Training data can be less structured
- Does not require extensive domain knowledge

Cons

- Time consuming to train
- Requires extensive computing power
- Requires large amounts of data

Future Directions

Continue expanding research on neural network applications

Crowd source data labeling to increase and improve training/testing data sets

Begin applying musical domain knowledge to neural networks

Questions?

References

J. Lee and J. Chun, "Musical instruments recognition using hidden markov model," in 2002, . DOI: 10.1109/ACSSC.2002.1197175.

M. Eichner, M. Wolff, R. Hoffmann, "Instrument classification using hidden Markov models", Proc. ISMIR, pp. 349-350, 2006.

A. Eronen, "Musical instrument recognition using ICA-based transform of features and discriminatively trained HMMs," in 2003, . DOI: 10.1109/ISSPA.2003.1224833.

T. Heittola, A. Klapuri, and T. Virtanen, “Musical instrument recognition in polyphonic audio using source-filter model for sound separation,” in Proc. Int. Soc. Music Inf. Retrieval Conf., 2009, pp. 327–332.

P. Li, J. Qian, and T. Wang, “Automatic instrument recognition in polyphonic music using convolutional neural networks,” arXiv:1511.05520, 2015.

L.-F. Yu, L. Su, and Y.-H. Yang, “Sparse cepstral codes and power scale for instrument identification,” in Proc. 2014 IEEE Int. Conf. Acoust., Speech Signal Process., 2014, pp. 7460–7464.

Z. Duan, B. Pardo, and L. Daudet, “A novel cepstral representation for timbre modeling of sound sources in polyphonic mixtures,” in Proc. 2014 IEEE Int. Conf. Acoust., Speech Signal Process., 2014, pp. 7495–7499.

Y. Han, J. Kim, K. Lee, “Deep Convolutional Neural Networks for Predominant Instrument Recognition in Polyphonic Music” in IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 25, No. 1, 2016, pp 208-221