

# Deep neural networks for music and audio tagging

Jordi Pons

jordipons.me – @jordiponsdotme

supervisor: Xavier Serra

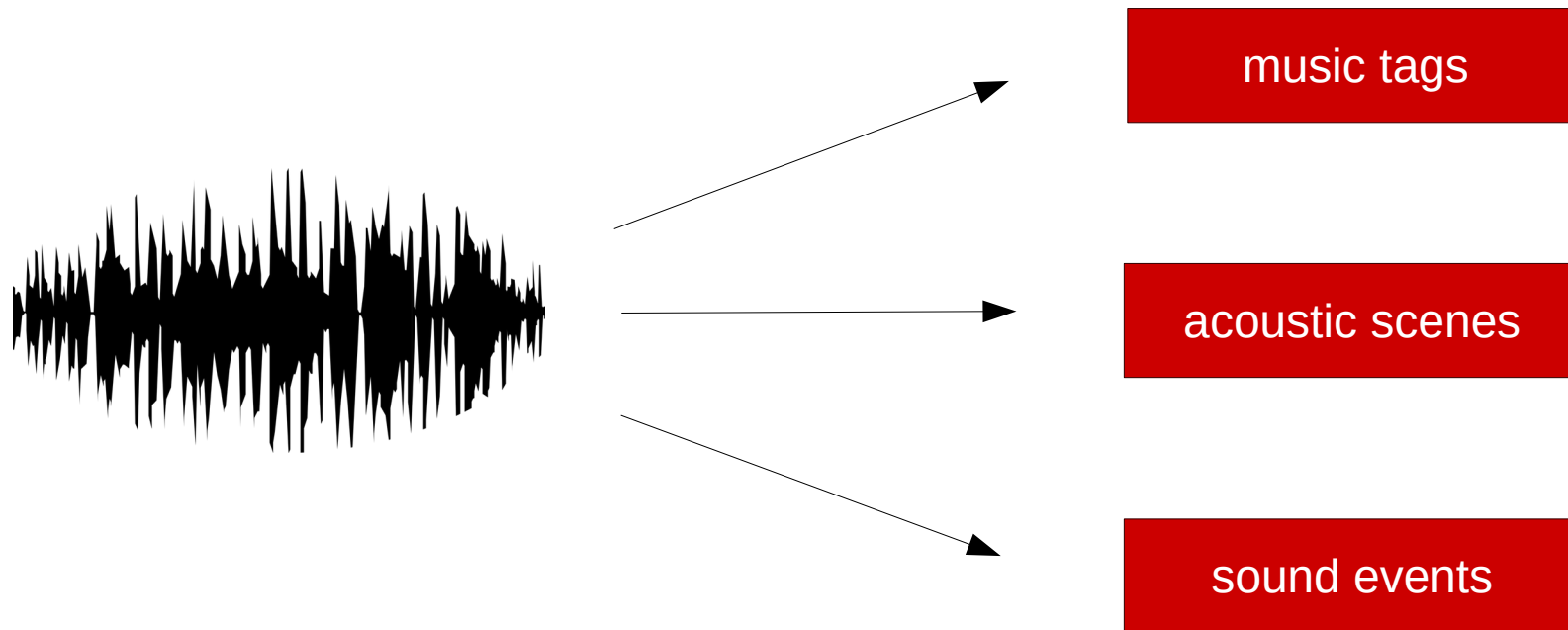


**M**usic  
**T**echnology  
**G**roup

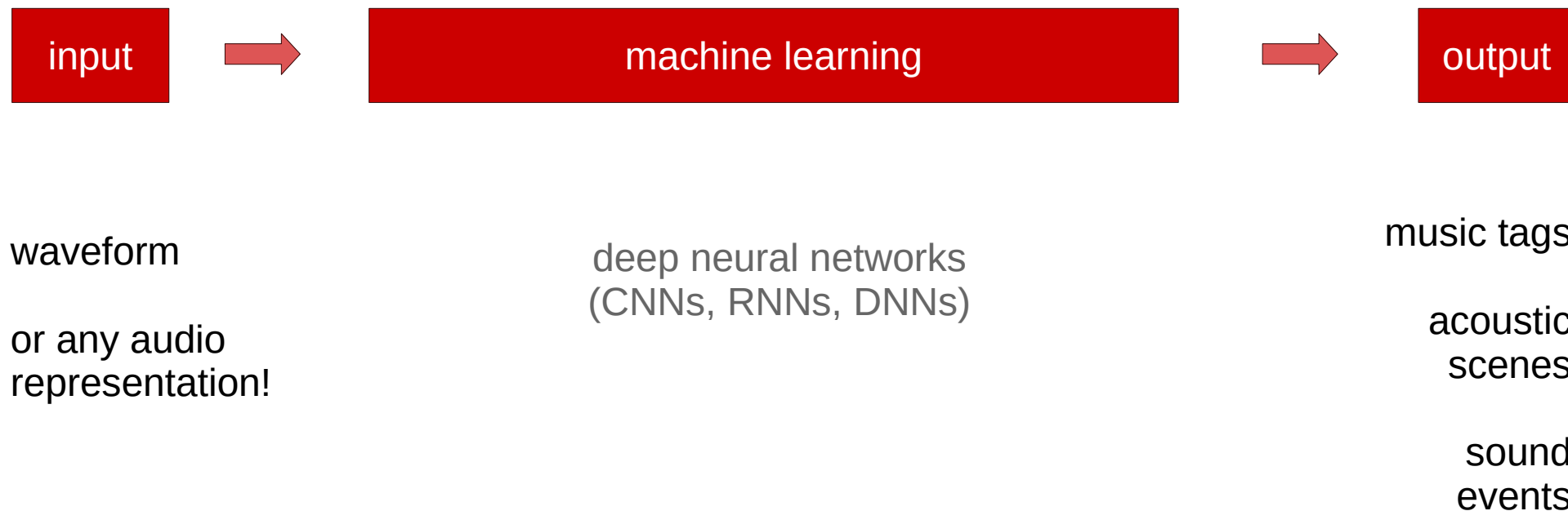


EXCELENCIA  
MARÍA  
DE MAEZTU

# Music and audio tagging



# Deep neural networks



# Research question I

- Deep artificial neural networks can be a **suitable tool** for modeling music and audio computationally.
- Artificial neural networks were **not widely used** for music and audio. Hence, deep learning was still a promise to be explored and it was not clear how researchers would adopt it.

**Which deep learning architectures are most appropriate for (music) audio signals?**

## Research question II

- It exists an **end-to-end learning trend** among deep learning researchers, who are exploring the possibilities of this approach.
- “End-to-end learning for audio is an **impossible** endeavor”. It existed the idea that for end-to-end learning to be viable, much more computing power and training data were required.

**In which scenarios is waveform-based end-to-end learning feasible?**

## Research question III

- Artificial neural networks require a **significant amount of data** to be competitive.

**How much data is required for carrying out competitive deep learning research?**

Which deep learning architectures are most appropriate for (music) audio signals?

In which scenarios is waveform-based end-to-end learning feasible?

How much data is required for carrying out competitive deep learning research?

- Musically Motivated CNNs for music tagging (Chapter III)
- Non-trained CNNs for music and audio tagging (Chapter IV)
- Music tagging at scale (Chapter V)
- Audio tagging with few training data (Chapter VI)
- Conclusions (Chapter VII)

Which deep learning architectures are most appropriate for (music) audio signals?

In which scenarios is waveform-based end-to-end learning feasible?

How much data is required for carrying out competitive deep learning research?

- **Musically Motivated CNNs for music tagging (Chapter III)**
- Non-trained CNNs for music and audio tagging (Chapter IV)
- Music tagging at scale (Chapter V)
- Audio tagging with few training data (Chapter VI)
- Conclusions (Chapter VII)



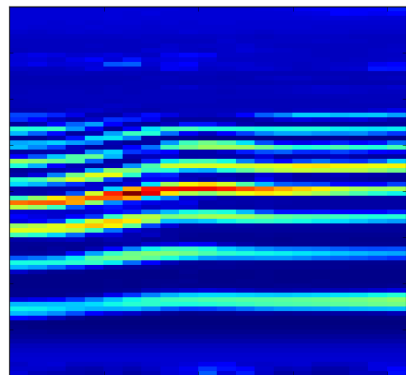
**Which deep learning architectures are most appropriate for (music) audio signals?**

In which scenarios is waveform-based end-to-end learning feasible?

How much data is required for carrying out competitive deep learning research?

- **Musically Motivated CNNs for music tagging (Chapter III)**
- Non-trained CNNs for music and audio tagging (Chapter IV)
- Music tagging at scale (Chapter V)
- Audio tagging with few training data (Chapter VI)
- Conclusions (Chapter VII)

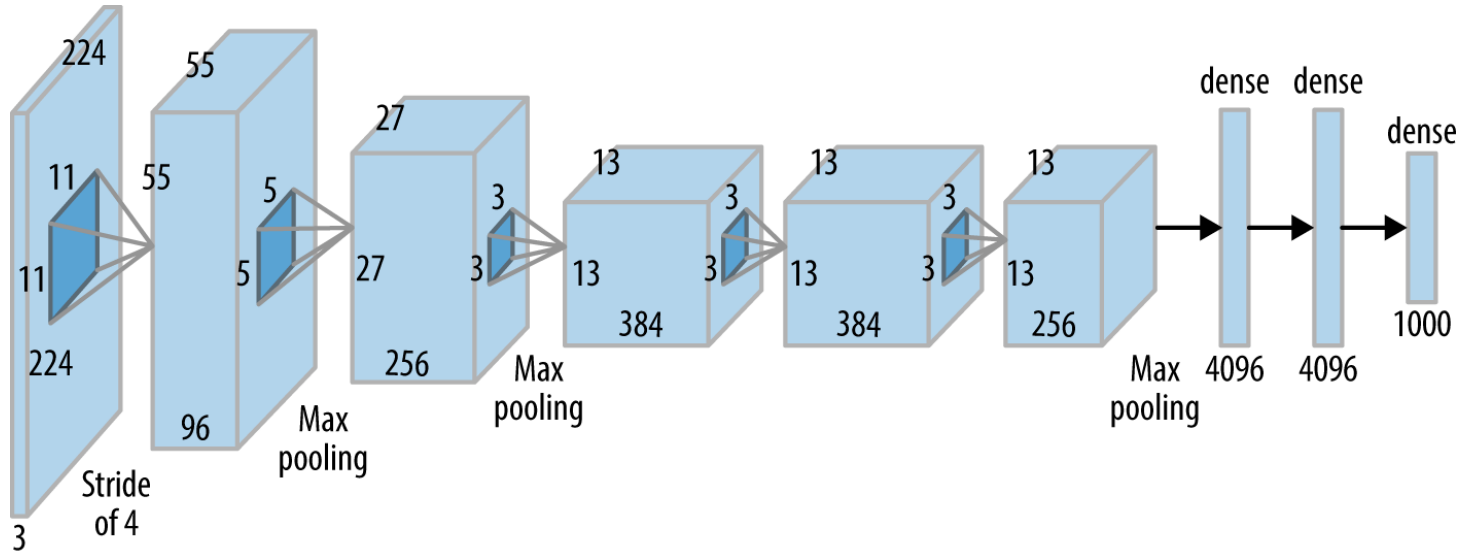
# Most researchers use computer vision architectures



## **Spectrograms are not images**

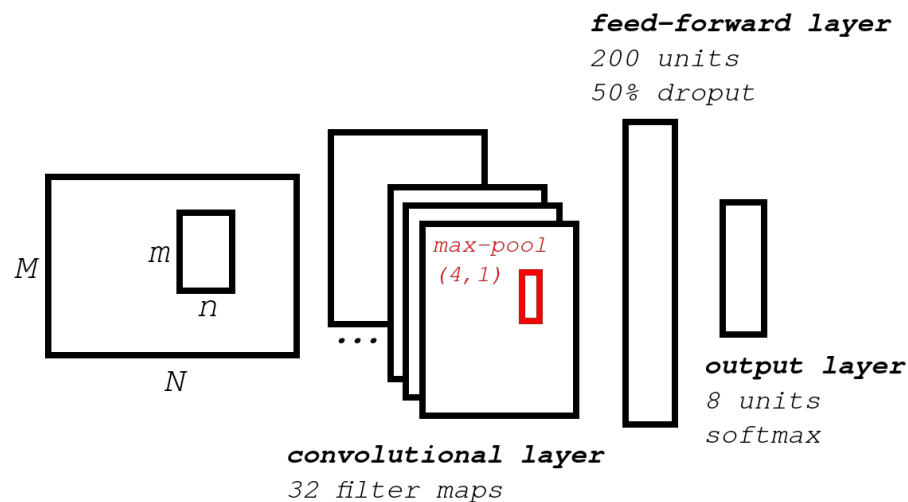
- No special meaning
- Vertical axis: frequency
- Horizontal axis: time

# Most researchers use computer vision architectures



AlexNet (Krizhevsky, 2012)

# Most researchers use computer vision architectures



## Spectrogram input

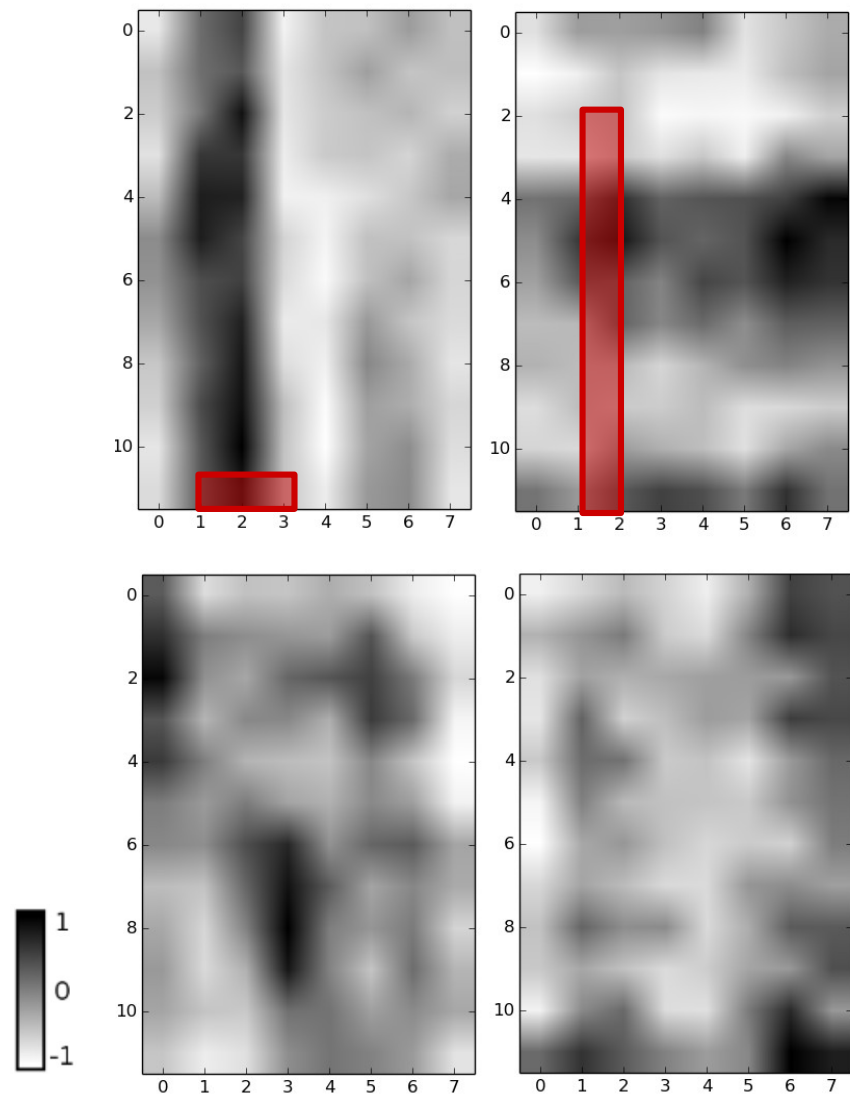
$M=40$  mel bands

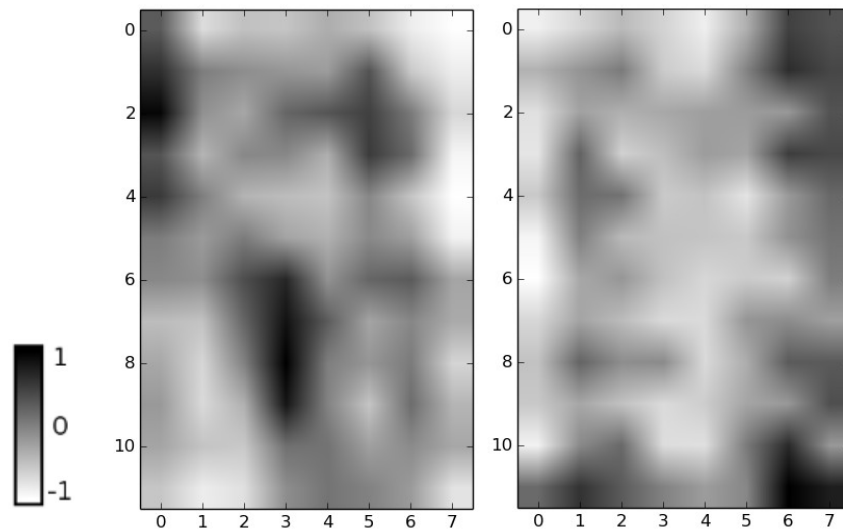
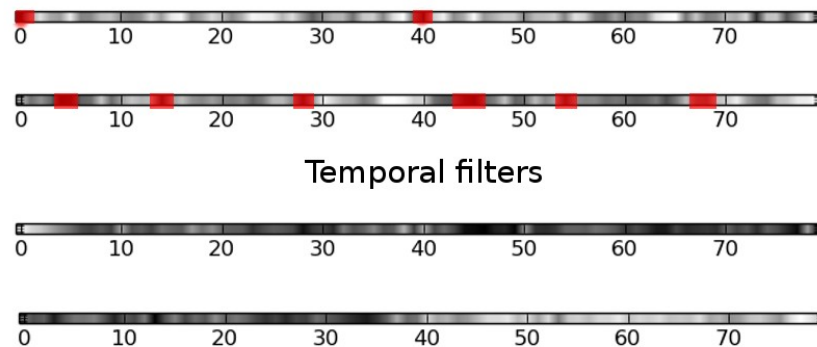
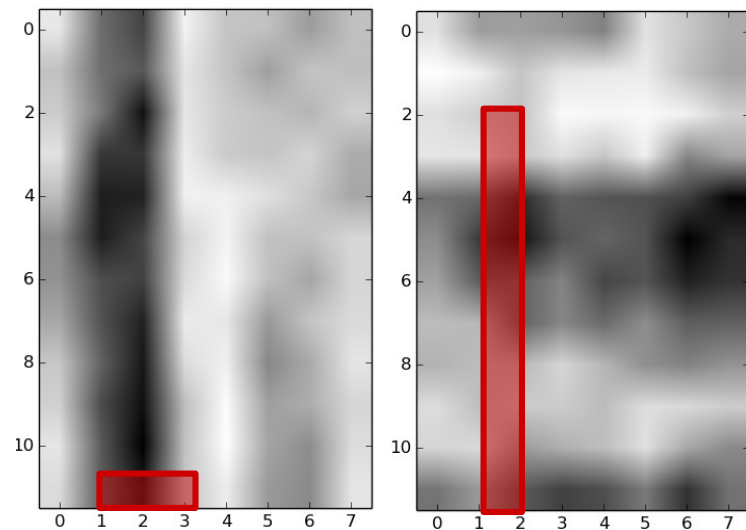
$N= 80$  frames (1.85 sec)

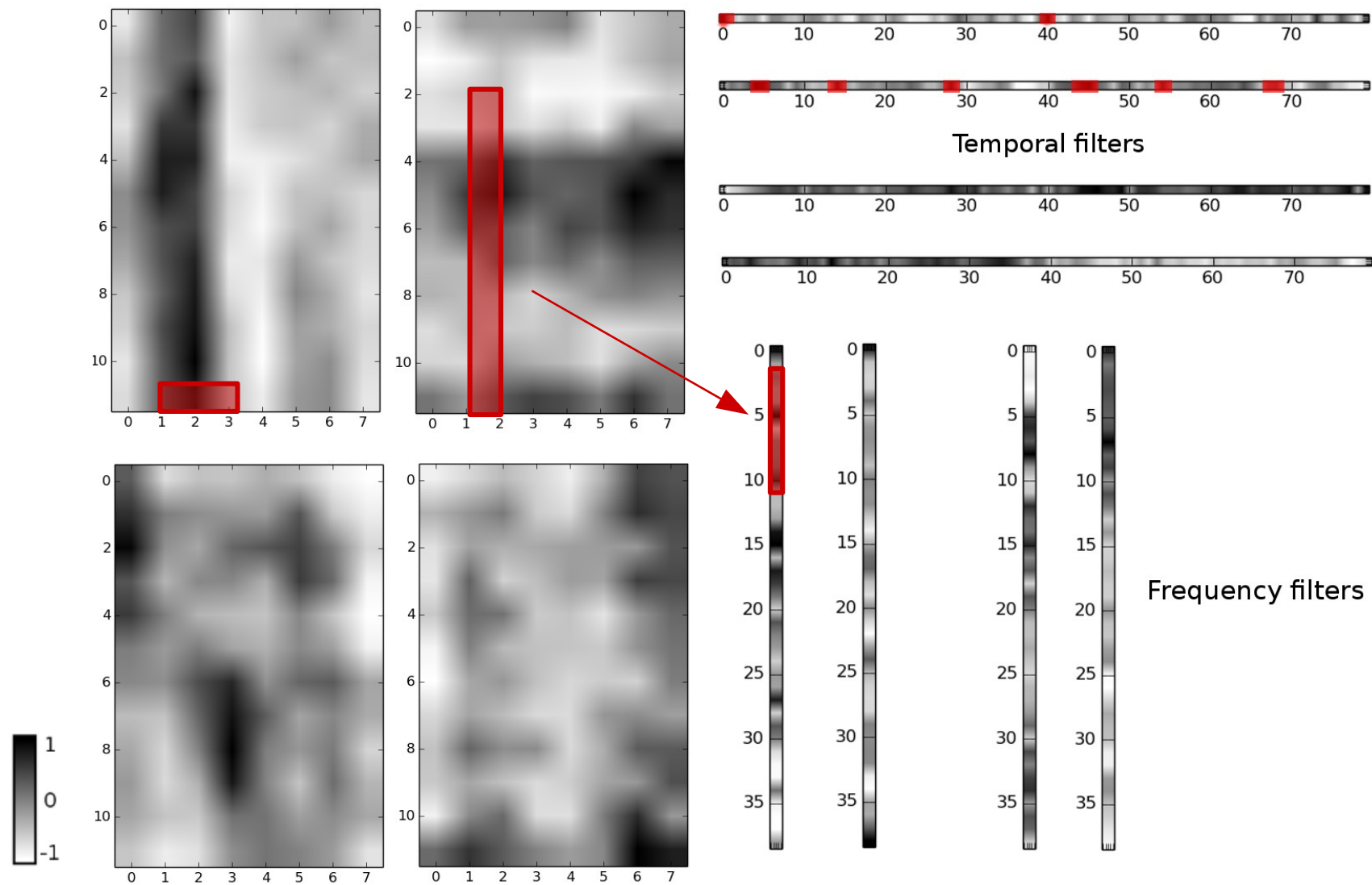
## CNN filter shape

$m=12$  mel bands

$n=8$  frames (0.18 sec)





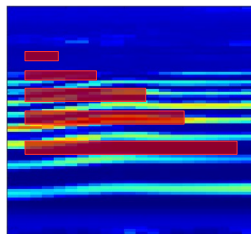


# Musically Motivated CNNs

- Discuss the importance of the CNN filter shapes
- Explore vertical or horizontal CNN filter shapes in the first layer
- Show that can perform similarly with an order of magnitude less number of learnable parameters
  - Task: classify rhythm classes
  - Performance:  $\approx 87\%$  accuracy

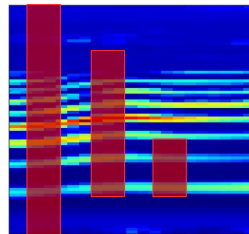


# TemporalCNN: many horizontal filters



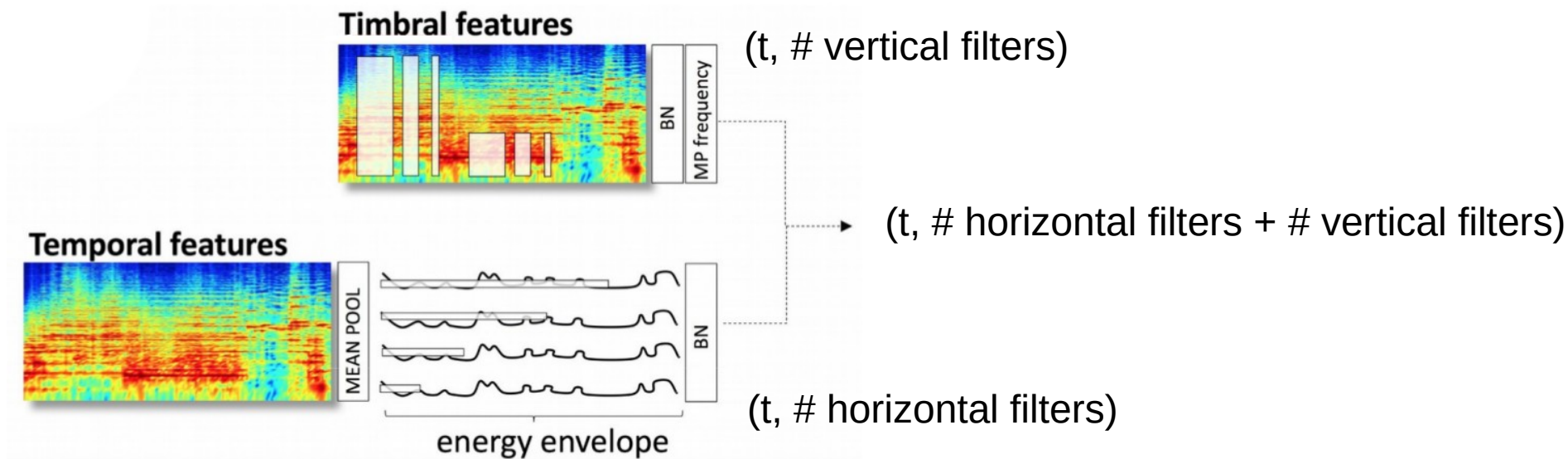
- Investigate using many different filters in the same layer
- Show that can perform the same with an extra order of magnitude less number of trainable parameters
  - Task: classify rhythm classes
  - Improve from  $\approx 87\%$  to  $\approx 92\%$  accuracy

# TimbreCNN: many vertical filters



- Pitch invariant filters: vertical convolution
- Show that can perform the same (if not better) with an order of magnitude less number of trainable parameters
  - Task I: Singing voice phoneme classification
  - Task II: Musical instrument recognition
  - Task III: Music tagging

# A novel design strategy for music CNNs



# How our work contributes to the state-of-the-art?



waveform

or any audio  
representation!

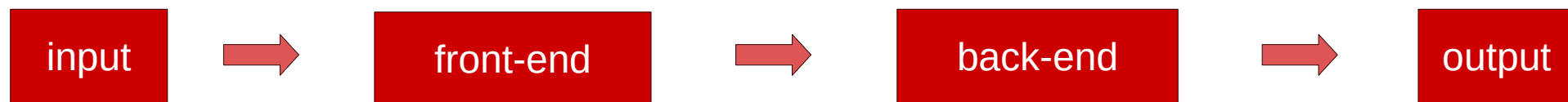
*deep neural networks*

music tags

acoustic  
scenes

sound  
events

# How our work contributes to the state-of-the-art?



waveform

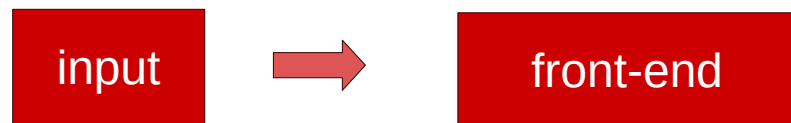
or any audio  
representation!

music tags

acoustic  
scenes

sound  
events

# How our work contributes to the state-of-the-art?



waveform

spectrogram

?

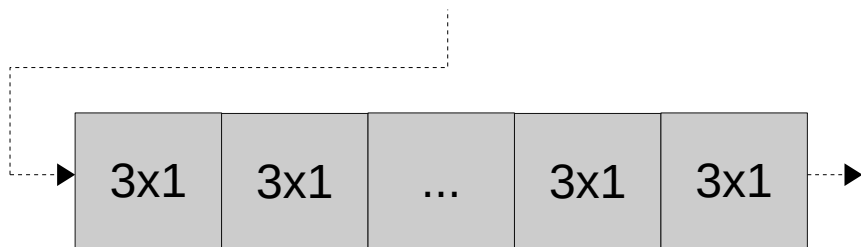
<b>based on domain knowledge?</b>	<b>filters config?</b>
---	----------------------------

<b>input signal?</b>
----------------------

<u><i>waveform</i></u>
------------------------

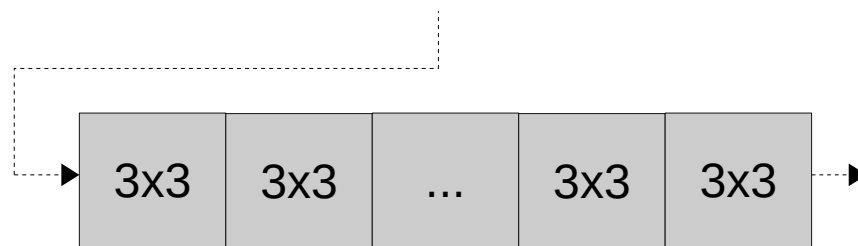
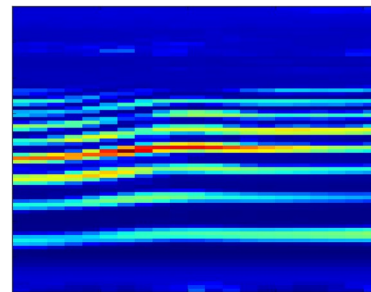
<u><i>spectrogram</i></u>
---------------------------

## Waveform end-to-end learning



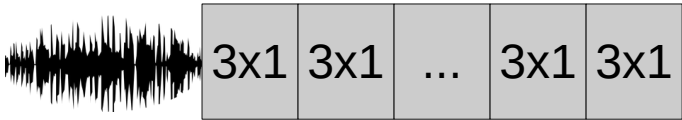
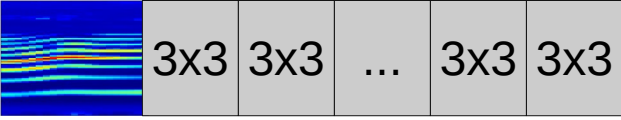
**Sample-level**

## Time-frequency representation *e.g.*: log-mel spectrogram



**VGG**

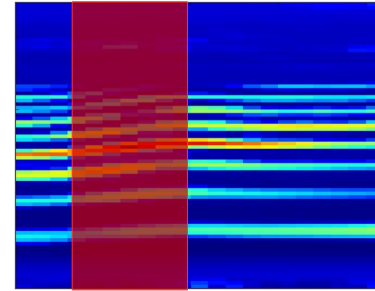


based on domain knowledge?	filters config?	input signal?	
		<u>waveform</u>	<u>spectrogram</u>
no	<u>minimal</u> filter expression	<p data-bbox="825 274 1098 322">sample-level</p> 	<p data-bbox="1625 274 1736 314">VGG</p> 

**Waveform**  
end-to-end learning

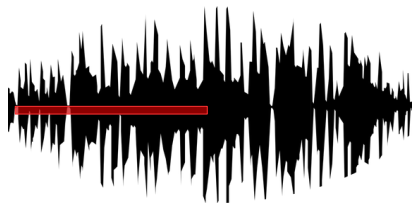


**Time-frequency representation**  
*e.g.*: log-mel spectrogram



## Waveform

end-to-end learning

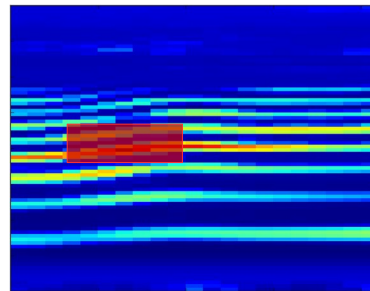


filter length: 512    *window length?*  
stride: 256        *hop size?*

**frame-level**

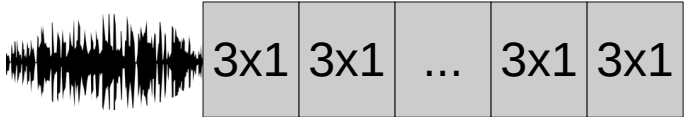
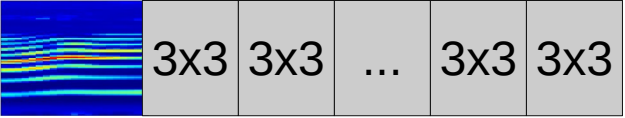
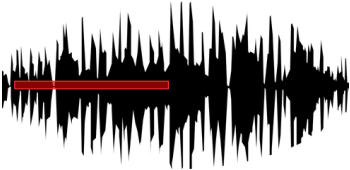

## Time-frequency representation

e.g.: log-mel spectrogram



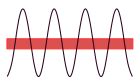
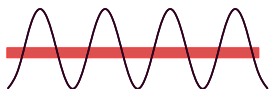
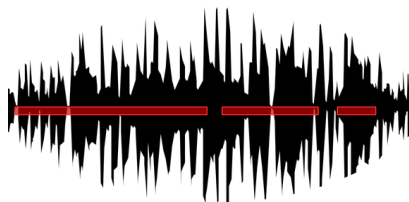
Explicitly tailoring the CNN towards  
learning temporal *or* timbral cues

**vertical or horizontal filters**

based on domain knowledge?	filters config?	input signal?	
		<u>waveform</u>	<u>spectrogram</u>
no	<u>minimal filter expression</u>	<p>sample-level</p> 	<p>VGG</p> 
yes	<u>single filter shape in 1<sup>st</sup> CNN layer</u>	<p>frame-level</p> 	<p>vertical OR horizontal</p> 

## Waveform

end-to-end learning

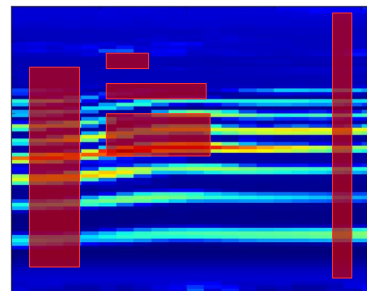


Efficient way  
to represent  
4 periods!

**Frame-level (many shapes!)**

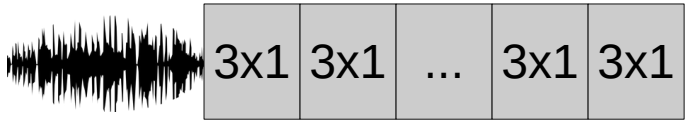
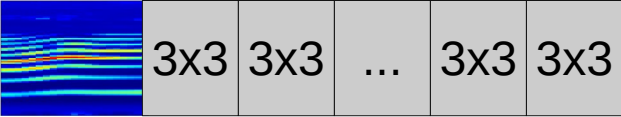
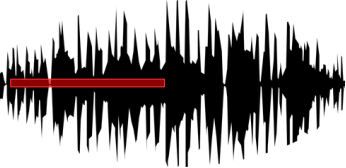
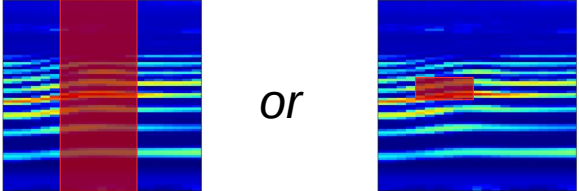
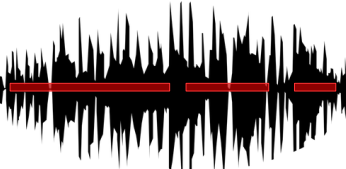
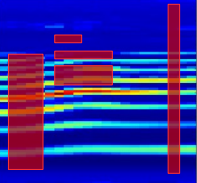
## Time-frequency representation

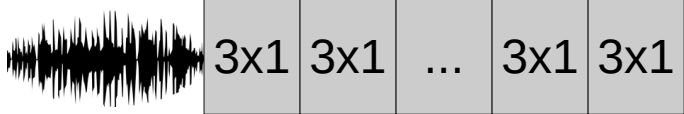
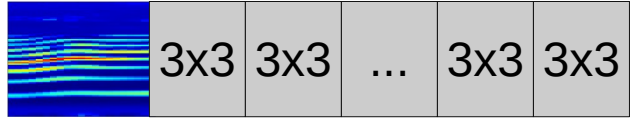
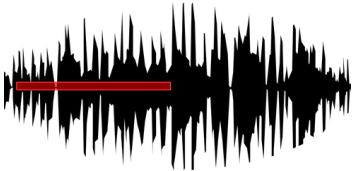
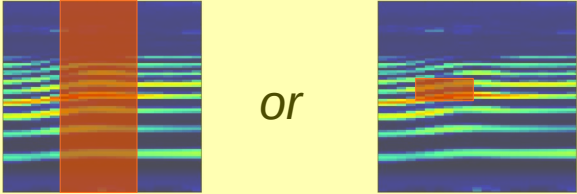
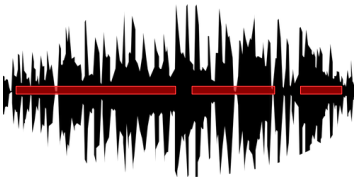
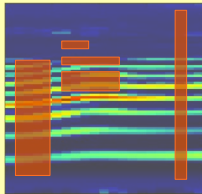
e.g.: log-mel spectrogram



Explicitly tailoring the CNN towards  
learning temporal *and* timbral cues

**Vertical and/or horizontal**

based on domain knowledge?	filters config?	input signal?	
		<u>waveform</u>	<u>spectrogram</u>
no	<u>minimal</u> filter expression	<p>sample-level</p> 	<p>VGG</p> 
yes	<u>single</u> filter shape in 1 <sup>st</sup> CNN layer	<p>frame-level</p> 	<p>vertical OR horizontal</p> 
yes	<u>many</u> filter shapes in 1 <sup>st</sup> CNN layer	<p>frame-level</p> 	<p>vertical AND/OR horizontal</p> 

based on domain knowledge?	filters config?	input signal?	
		<u>waveform</u>	<u>spectrogram</u>
no	<u>minimal</u> filter expression	<p>sample-level</p>  <p>A black waveform is shown on the left. To its right is a sequence of five gray boxes, each labeled '3x1'. The second and fourth boxes are separated by an ellipsis '...'.</p>	<p>VGG</p>  <p>A spectrogram is shown on the left. To its right is a sequence of six gray boxes, each labeled '3x3'. The third box is separated by an ellipsis '...'.</p>
yes	<u>single</u> filter shape in 1 <sup>st</sup> CNN layer	<p>frame-level</p>  <p>A black waveform is shown. A single red horizontal bar is drawn across the middle of the waveform, representing a single frame-level filter.</p>	<p>vertical OR horizontal</p>  <p>Two spectrograms are shown side-by-side, separated by the word 'or'. The left spectrogram has a red vertical bar, and the right spectrogram has a red horizontal bar.</p>
yes	<u>many</u> filter shapes in 1 <sup>st</sup> CNN layer	<p>frame-level</p>  <p>A black waveform is shown. Three red horizontal bars are drawn across the waveform at different time intervals, representing multiple frame-level filters.</p>	<p>vertical AND/OR horizontal</p>  <p>A spectrogram is shown with several red rectangles of different sizes and orientations (vertical and horizontal) overlaid on it, representing multiple filter shapes.</p>

# CNN front-ends for audio classification

**Sample-level:** Lee et al., 2017 – **Sample-level Deep Convolutional Neural Networks for Music Auto-tagging Using Raw Waveforms** in *Sound and Music Computing Conference (SMC)*

**VGG:** Choi et al., 2016 – **Automatic tagging using deep convolutional neural networks** in *Proceedings of the ISMIR (International Society of Music Information Retrieval) Conference*

**Frame-level (single shape):** Dieleman et al., 2014 – **End-to-end learning for music audio** in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*

**Vertical:** Lee et al., 2009 – **Unsupervised feature learning for audio classification using convolutional deep belief networks** in *Advances in Neural Information Processing Systems (NIPS)*

**Horizontal:** Schluter & Bock, 2014 – **Improved musical onset detection with convolutional neural networks** in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*

**Frame-level (many shapes):** Zhu et al., 2016 – **Learning multiscale features directly from waveforms** in *arXiv:1603.09509*

**Vertical and horizontal (many shapes):** Pons, et al., 2016 – **Experimenting with musically motivated convolutional neural networks** in *14th International Workshop on Content-Based Multimedia Indexing*



Which deep learning architectures are most appropriate for (music) audio signals?

In which scenarios is waveform-based end-to-end learning feasible?

How much data is required for carrying out competitive deep learning research?

- Musically Motivated CNNs for music tagging (Chapter III)
- Non-trained CNNs for music and audio tagging (Chapter IV)
- Music tagging at scale (Chapter V)
- Audio tagging with few training data (Chapter VI)
- Conclusions (Chapter VII)

Which deep learning architectures are most appropriate for (music) audio signals?

In which scenarios is waveform-based end-to-end learning feasible?

How much data is required for carrying out competitive deep learning research?

- Musically Motivated CNNs for music tagging (Chapter III)
- **Non-trained CNNs for music and audio tagging (Chapter IV)**
- Music tagging at scale (Chapter V)
- Audio tagging with few training data (Chapter VI)
- Conclusions (Chapter VII)

**Which deep learning architectures are most appropriate for (music) audio signals?**

**In which scenarios is waveform-based end-to-end learning feasible?**

**How much data is required for carrying out competitive deep learning research?**

- Musically Motivated CNNs for music tagging (Chapter III)
- **Non-trained CNNs for music and audio tagging (Chapter IV)**
- Music tagging at scale (Chapter V)
- Audio tagging with few training data (Chapter VI)
- Conclusions (Chapter VII)

Which deep learning architectures are most appropriate for (music) audio signals?

In which scenarios is waveform-based end-to-end learning feasible?

How much data is required for carrying out competitive deep learning research?

- Musically Motivated CNNs for music tagging (Chapter III)
- **Non-trained CNNs for music and audio tagging (Chapter IV)**
- Music tagging at scale (Chapter V)
- Audio tagging with few training data (Chapter VI)
- Conclusions (Chapter VII)

# Methodology

**Goal?** Compare different (randomly weighted) architectures

**Method?** Features (embeddings of random CNN)  
+ classifier

Compare classification accuracies when  
using different (randomly weighted) architectures

# Methodology

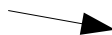
---

## On Random Weights and Unsupervised Feature Learning

---

Andrew M. Saxe, Pang Wei Koh, Zhenghao Chen,  
Maneesh Bhand, Bipin Suresh, and Andrew Y. Ng  
Stanford University  
Stanford, CA 94305

{asaxe, pangwei, zhenghao, mbhand, bipins, ang}@cs.stanford.edu



### 4 Fast architecture selection

When we plot the classification performance of random-weight architectures against trained-weight architectures, a distinctive trend emerges: we see that architectures which perform well with random weights also tend to perform well with pretrained and finetuned weights, and vice versa (Fig. 5). Intuitively, our analysis in Section 2 suggests that random-weight performance is not truly random but should correlate with the corresponding trained-weight performance, as both are linked to intrinsic properties of the architecture. Indeed, this happens in practice.

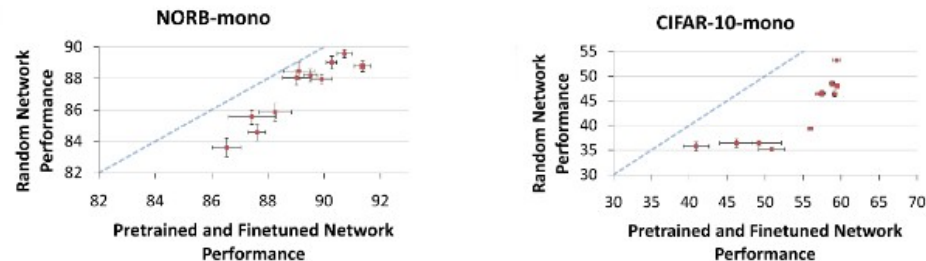
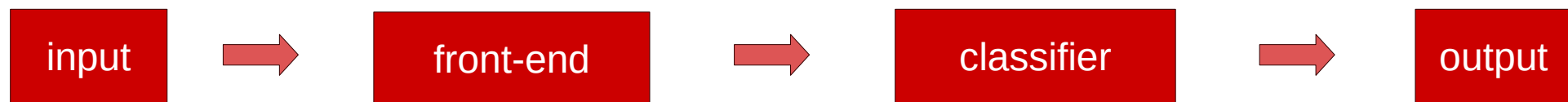


Figure 5: Classification performance of random-weight networks vs pretrained and finetuned networks. Left: NORB-mono. Right: CIFAR-10-mono (Error bars represent a 95% confidence interval about the mean)


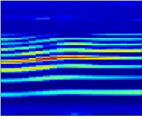
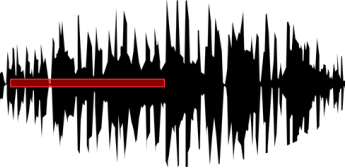
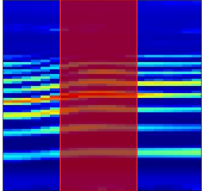
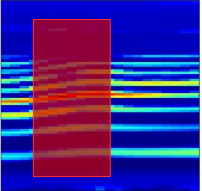
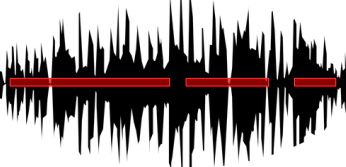
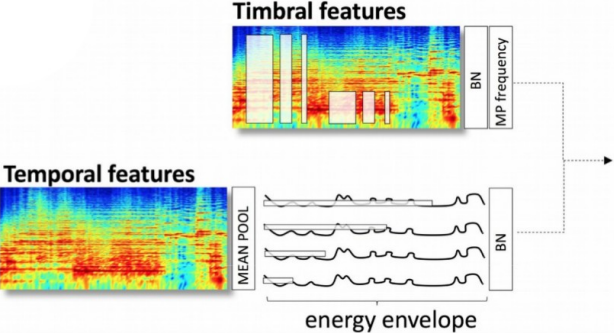


waveform

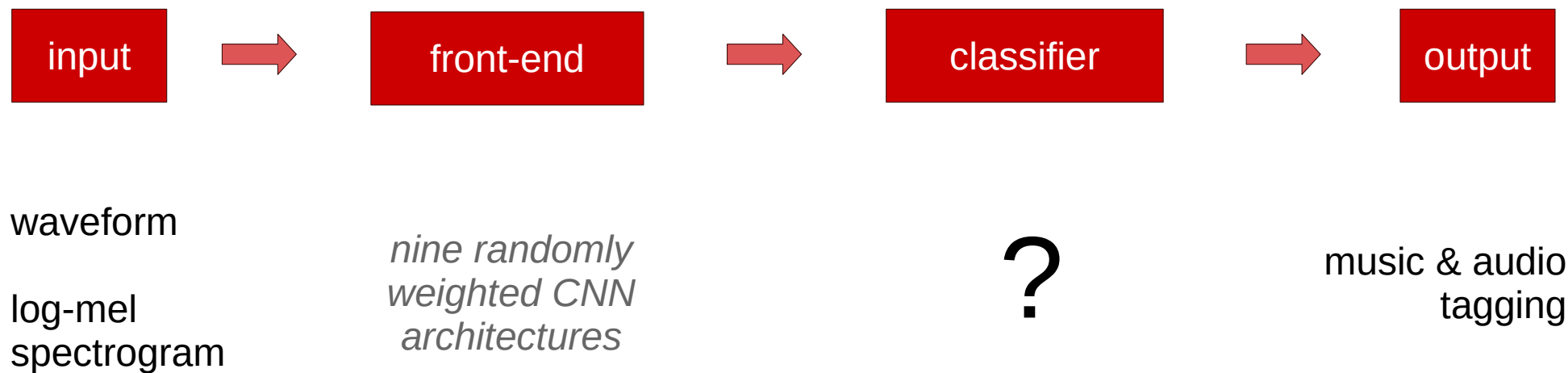
log-mel  
spectrogram

?

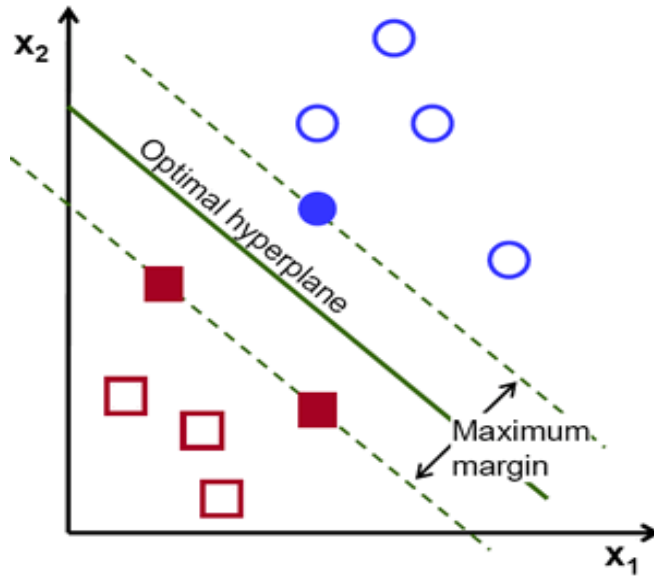
music & audio  
tagging

based on domain knowledge?	filters config?	input signal?	
		<u>waveform</u>	<u>log-mel spectrogram</u>
no	<u>minimal</u> filter expression	<p>sample-level</p>  <p>3x1 3x1 ... 3x1 3x1</p>	<p>VGG</p>  <p>3x3 3x3 ... 3x3 3x3</p>
yes	<u>single</u> filter shape in 1 <sup>st</sup> CNN layer	<p>frame-level</p> 	<p>7x96</p>  <p>7x86</p> 
yes	<u>many</u> filter shapes in 1 <sup>st</sup> CNN layer	<p>frame-level</p> 	 <p>Timbral features</p> <p>Temporal features</p> <p>energy envelope</p>

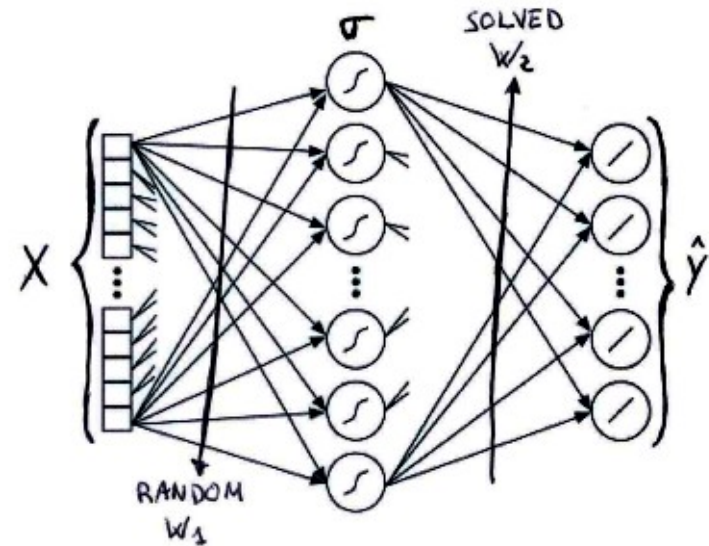




# Studied classifiers: SVM and ELM classifiers

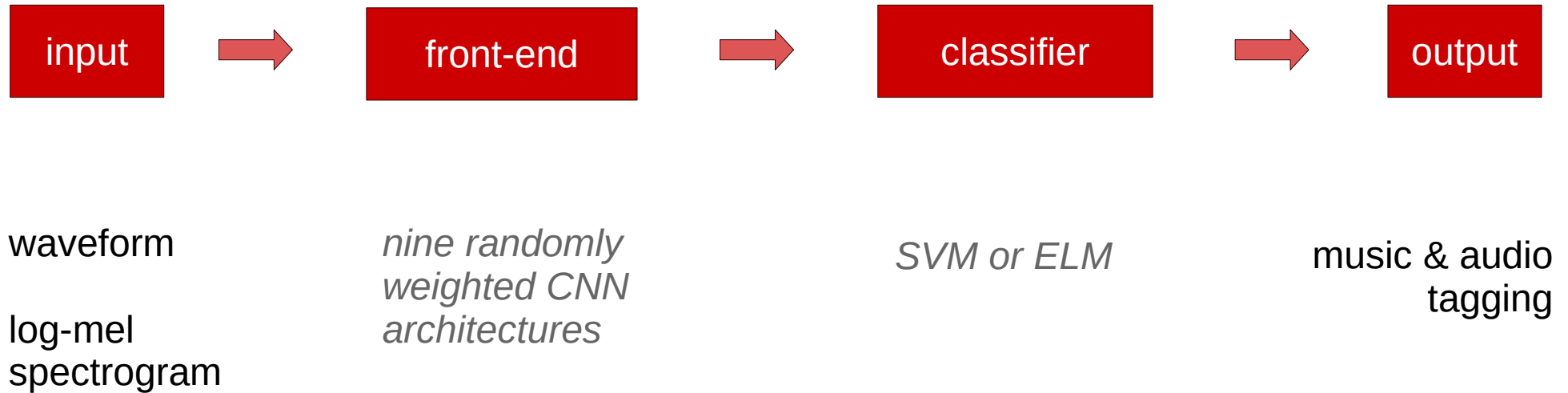


SVM: support  
vector machine



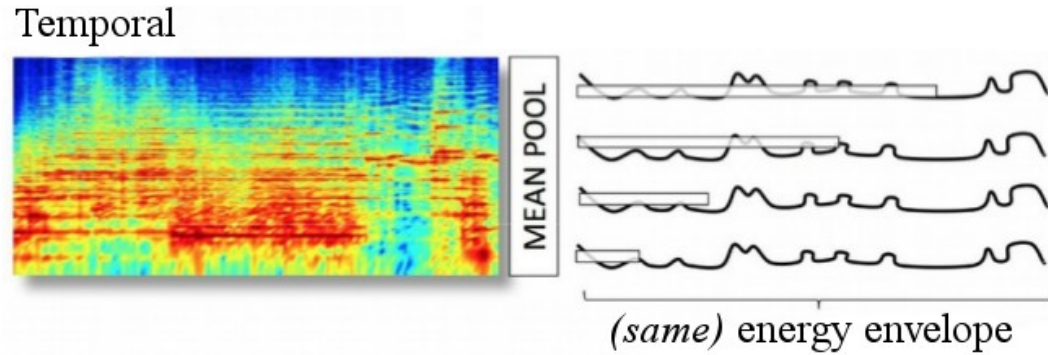
ELM: extreme  
learning machine

# The deep learning pipeline: output

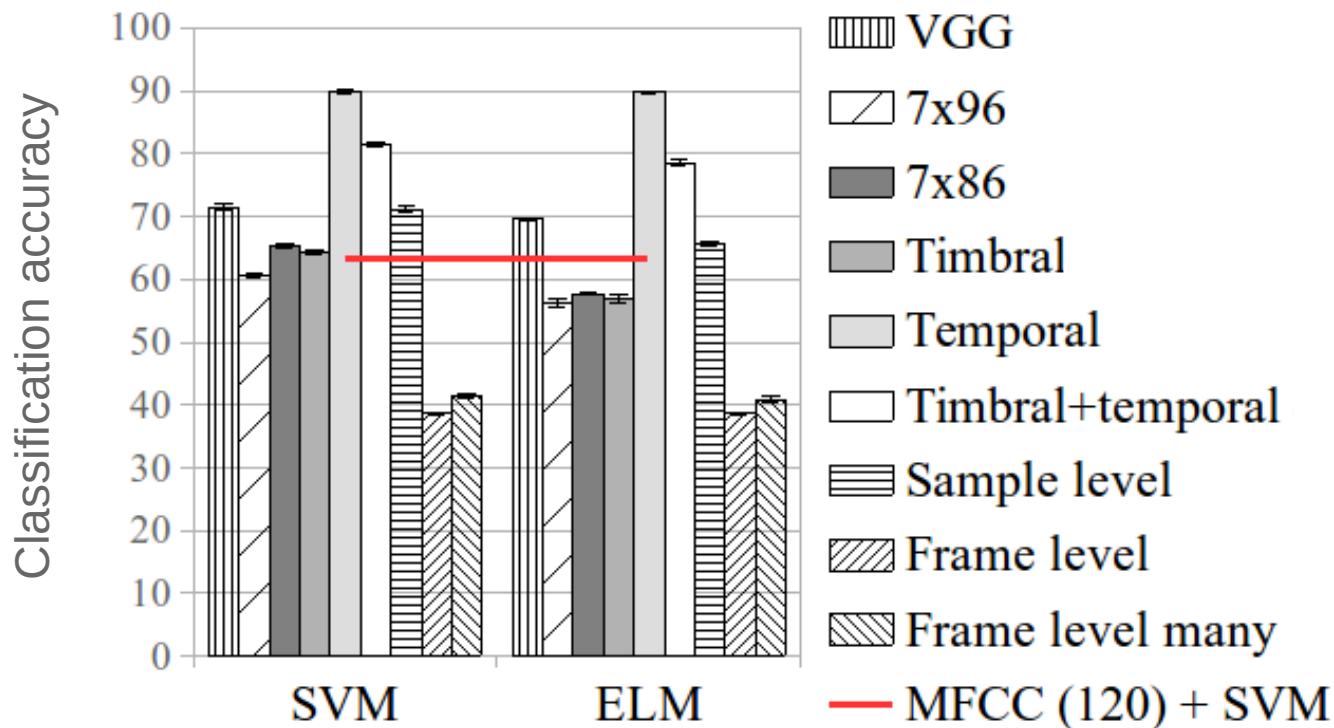


# Random CNN features: Extended Ballroom

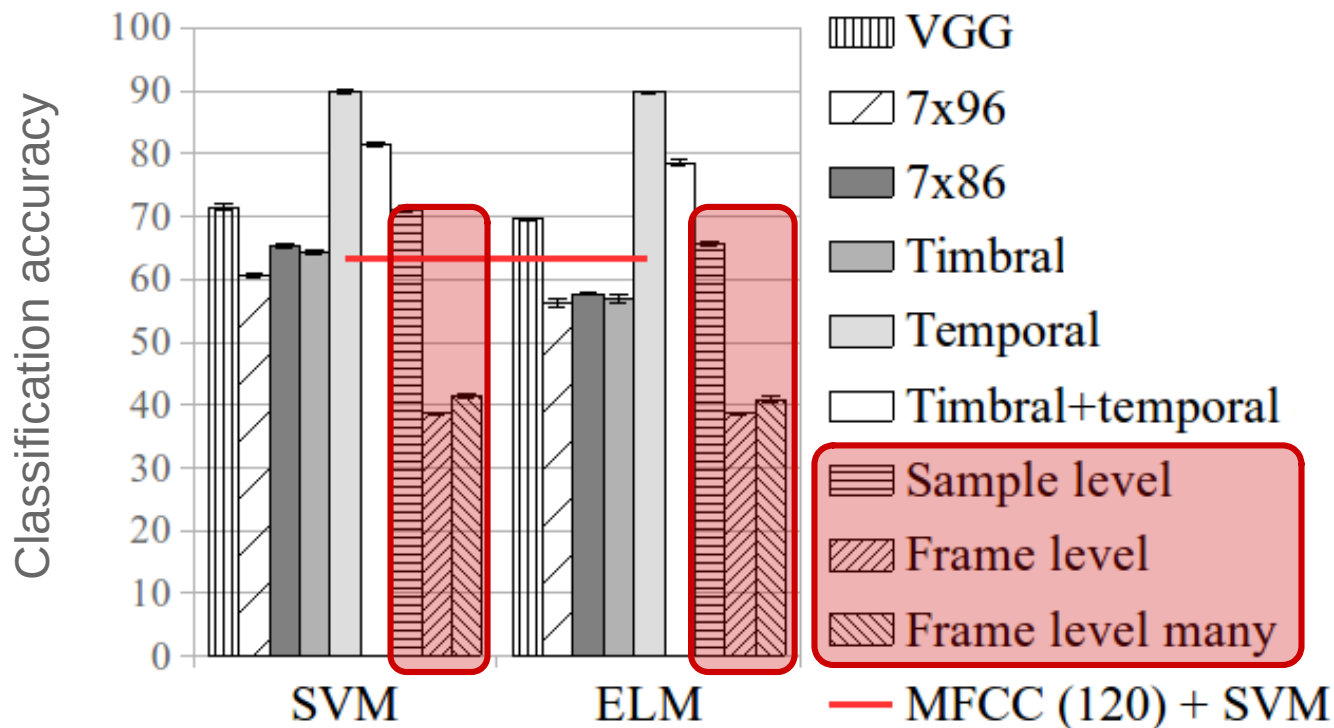
# Do you remember the *temporal* CNN?



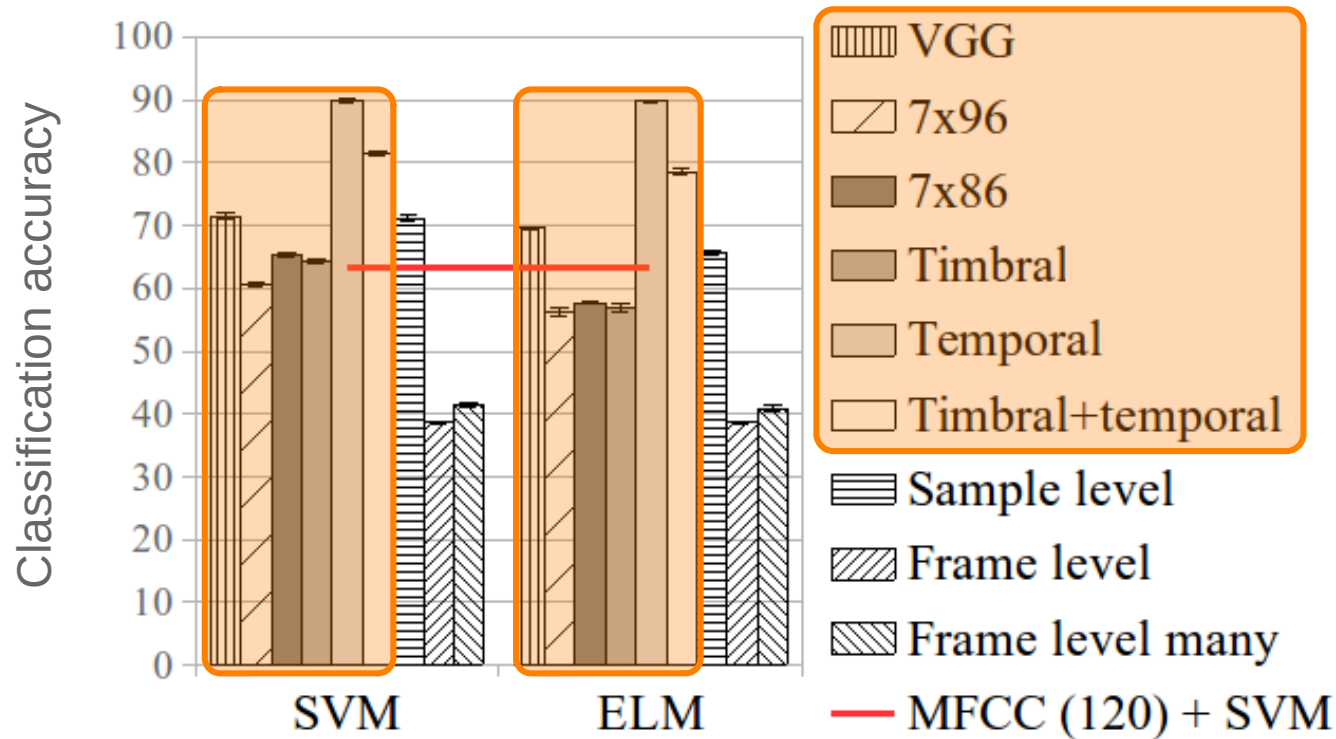
# Random CNN features: Extended Ballroom



# Random CNN features: Extended Ballroom



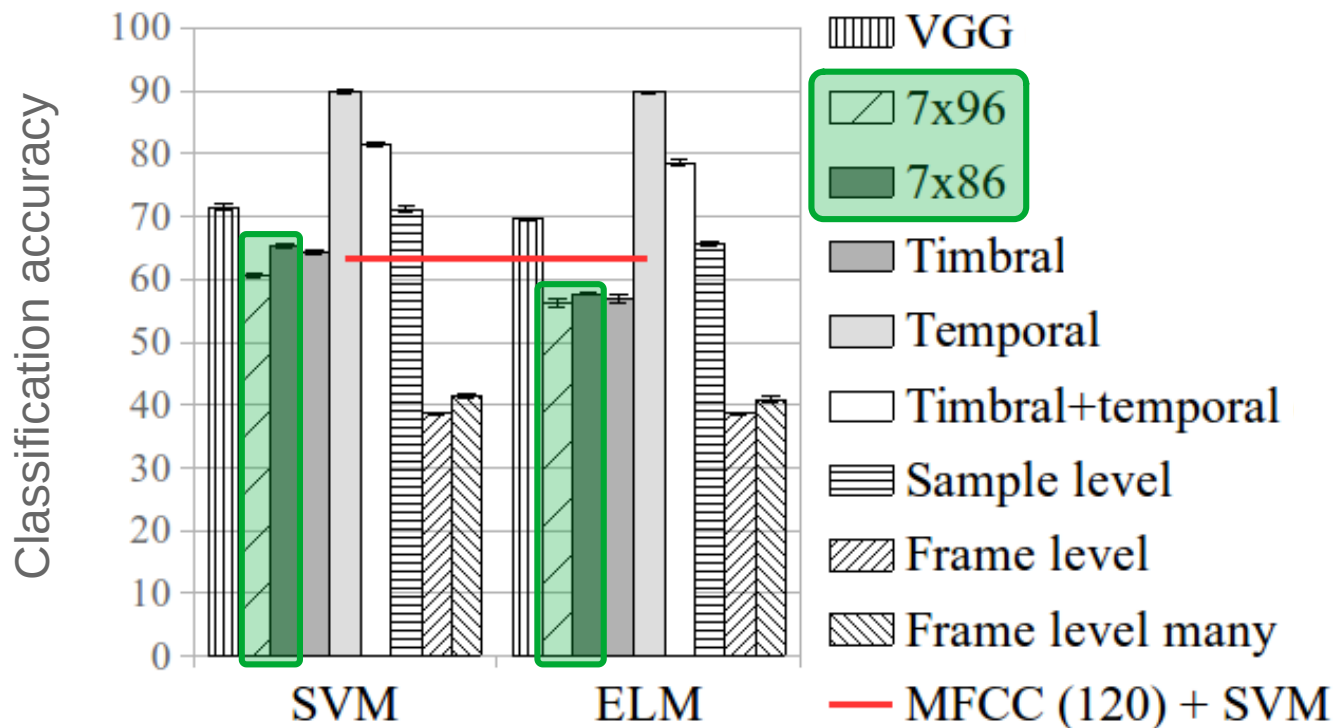
# Random CNN features: Extended Ballroom



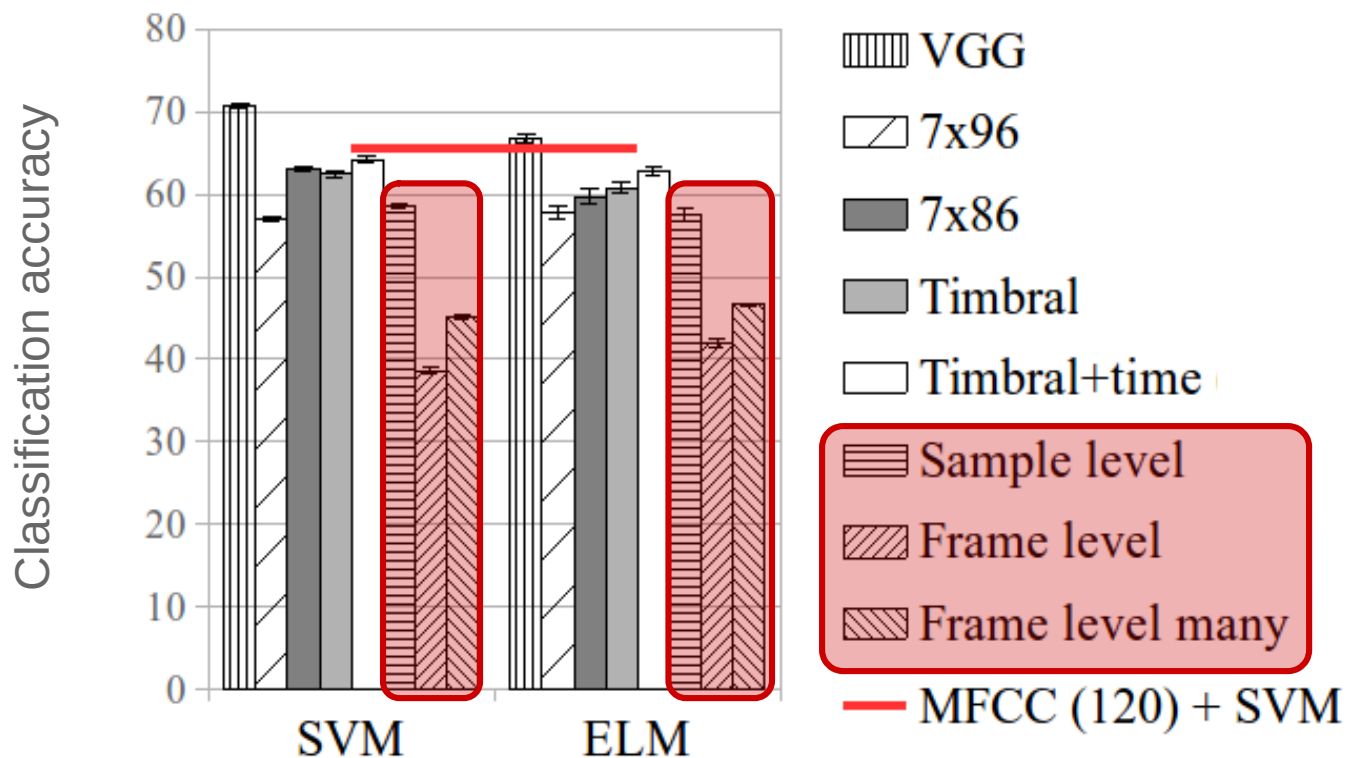
89.82 % (best random CNN) < 93.7 % (SOTA)



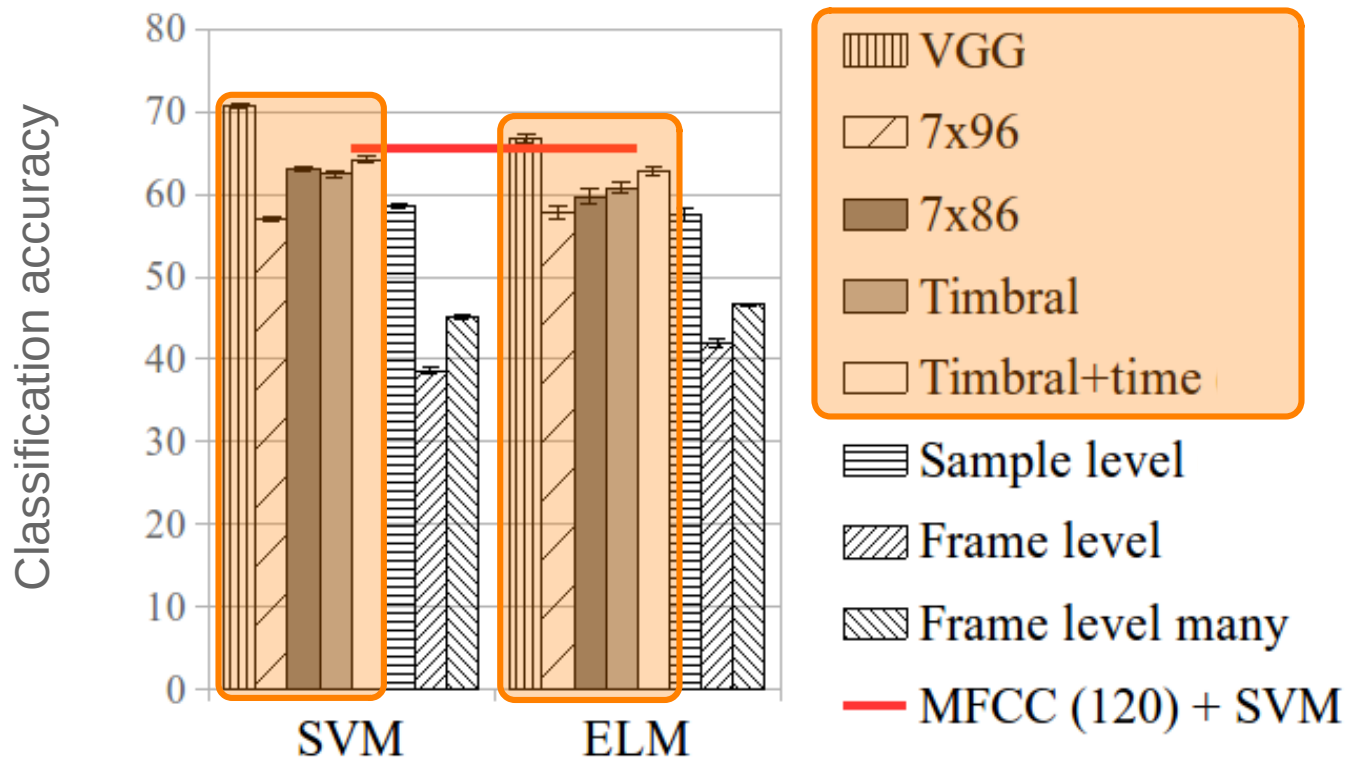
# Random CNN features: Extended Ballroom



# Random CNN features: Urban Sound 8k

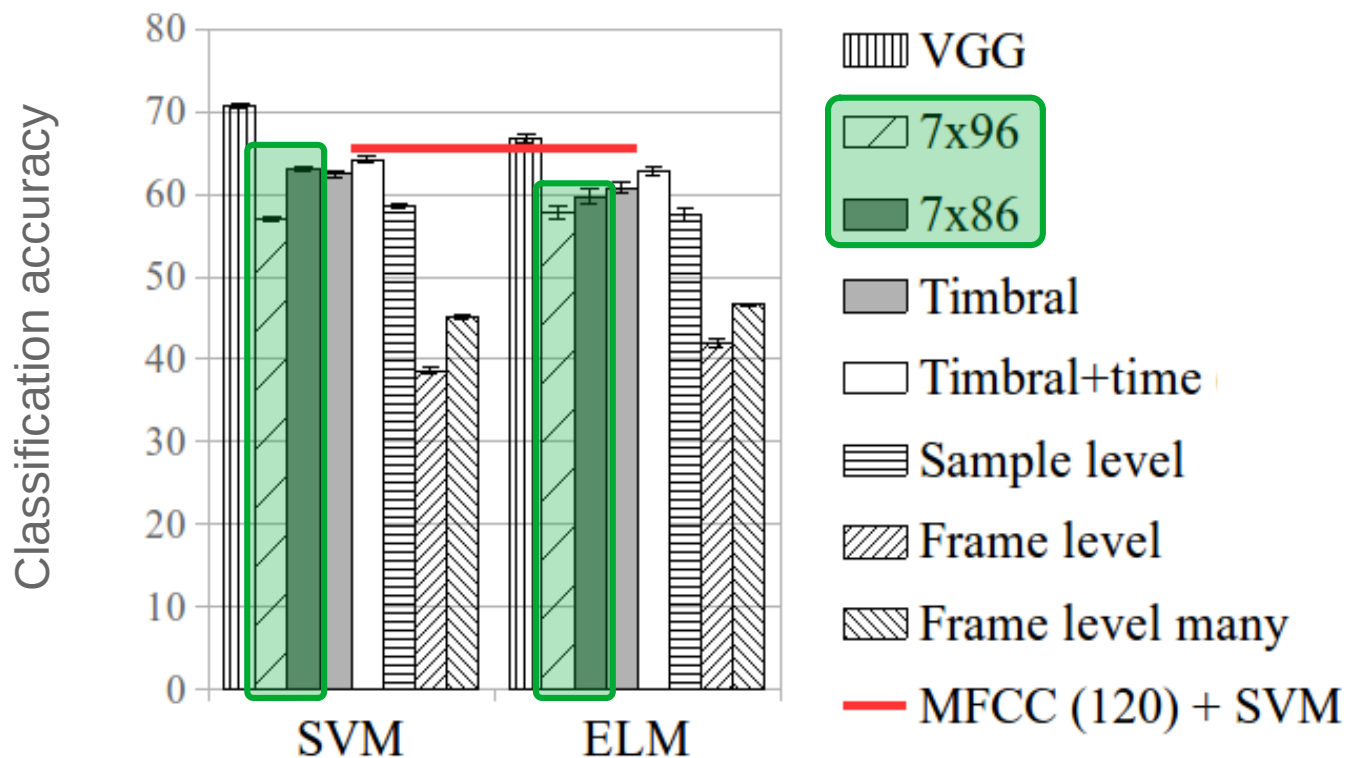


# Random CNN features: Urban Sound 8k



70.74 % (best random CNN) < 73 % (SOTA)

# Random CNN features: Urban Sound 8k



# Summary

- **Waveform front-ends:** sample-level >> frame-level many > frame-level
- **Spectrogram front-ends:** allowing pitch-shifting is beneficial (7x86>7x96)
- **Music tagging:** using prior music domain knowledge can be useful
- **Audio tagging:** the VGG, a computer vision architecture, achieves the best results

Which deep learning architectures are most appropriate for (music) audio signals?

In which scenarios is waveform-based end-to-end learning feasible?

How much data is required for carrying out competitive deep learning research?

- Musically Motivated CNNs for music tagging (Chapter III)
- Non-trained CNNs for music and audio tagging (Chapter IV)
- Music tagging at scale (Chapter V)
- Audio tagging with few training data (Chapter VI)
- Conclusions (Chapter VII)

Which deep learning architectures are most appropriate for (music) audio signals?

In which scenarios is waveform-based end-to-end learning feasible?

How much data is required for carrying out competitive deep learning research?

- Musically Motivated CNNs for music tagging (Chapter III)
- Non-trained CNNs for music and audio tagging (Chapter IV)
- **Music tagging at scale (Chapter V)**
- Audio tagging with few training data (Chapter VI)
- Conclusions (Chapter VII)

Which deep learning architectures are most appropriate for (music) audio signals?

In which scenarios is waveform-based end-to-end learning feasible?

**How much data is required for carrying out competitive deep learning research?**

- Musically Motivated CNNs for music tagging (Chapter III)
- Non-trained CNNs for music and audio tagging (Chapter IV)
- **Music tagging at scale (Chapter V)**
- Audio tagging with few training data (Chapter VI)
- Conclusions (Chapter VII)



Which deep learning architectures are most appropriate for (music) audio signals?

In which scenarios is waveform-based end-to-end learning feasible?

How much data is required for carrying out competitive deep learning research?

- Musically Motivated CNNs for music tagging (Chapter III)
- Non-trained CNNs for music and audio tagging (Chapter IV)
- **Music tagging at scale (Chapter V)**
- Audio tagging with few training data (Chapter VI)
- Conclusions (Chapter VII)

MagnaTT  
**25k**  
songs

Million song dataset  
**250k**  
songs

MagnaTT  
**25k**  
songs

Million song dataset  
**250k**  
songs

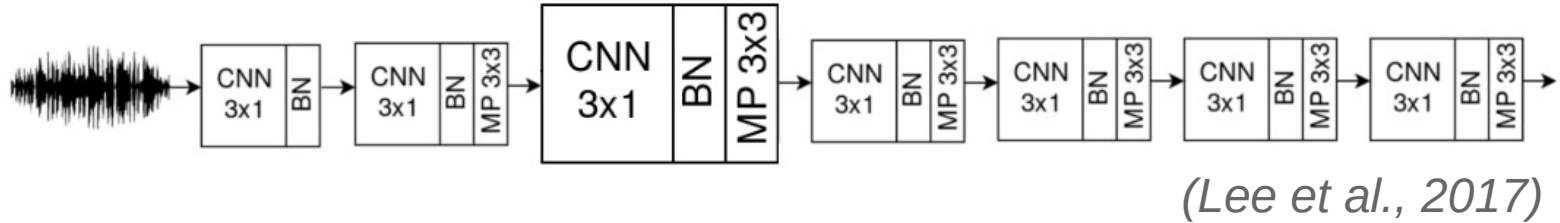
**1M**  
songs

Which deep learning models  
perform best at scale?

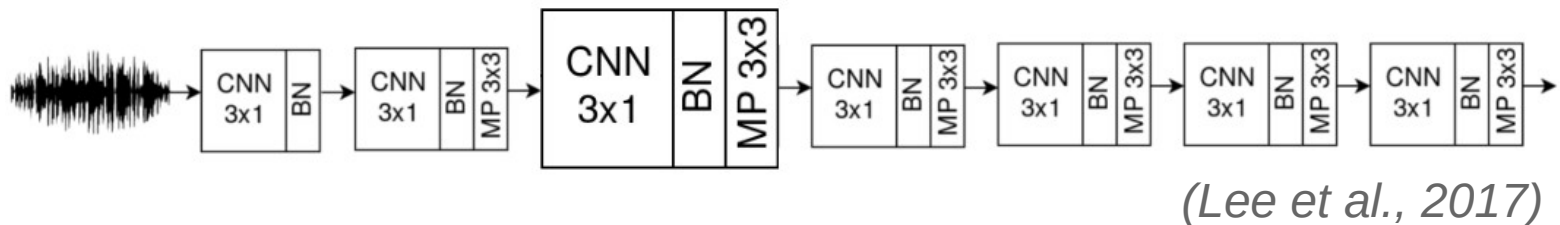
# Which deep learning models perform best at scale?

**waveform-based** model – generic CNN architecture

# Which deep learning models perform best at scale?

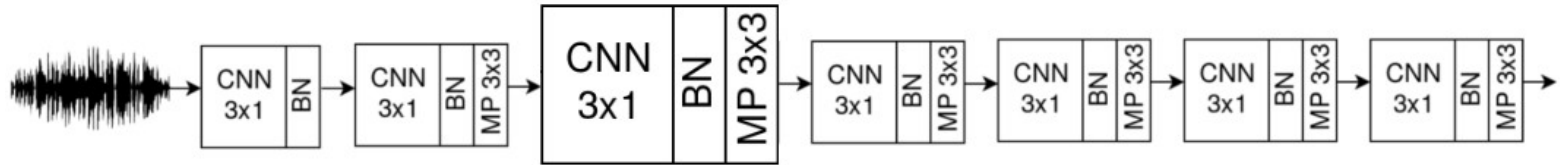


# Which deep learning models perform best at scale?

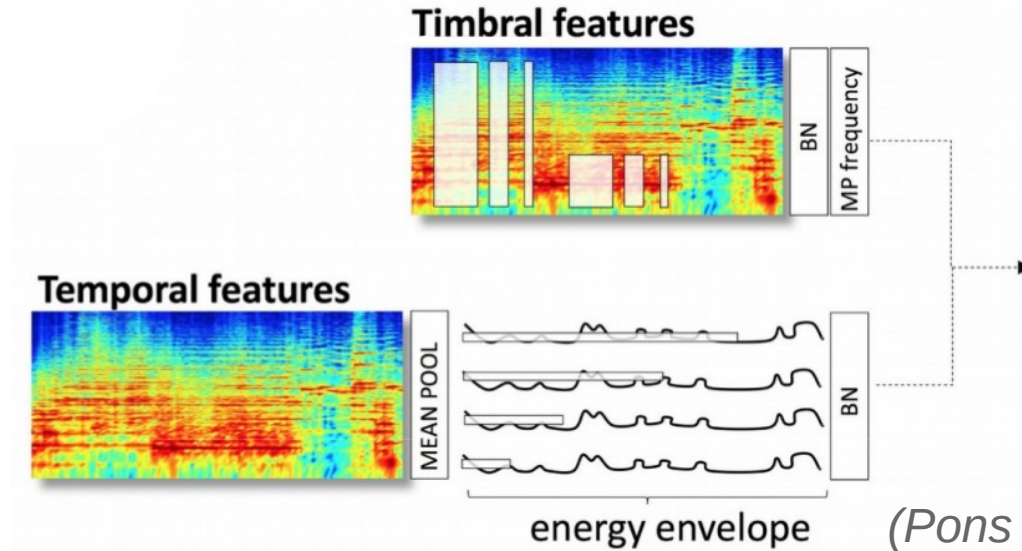


**spectrogram-based** model – CNN architecture for music

# Which deep learning models perform best at scale?



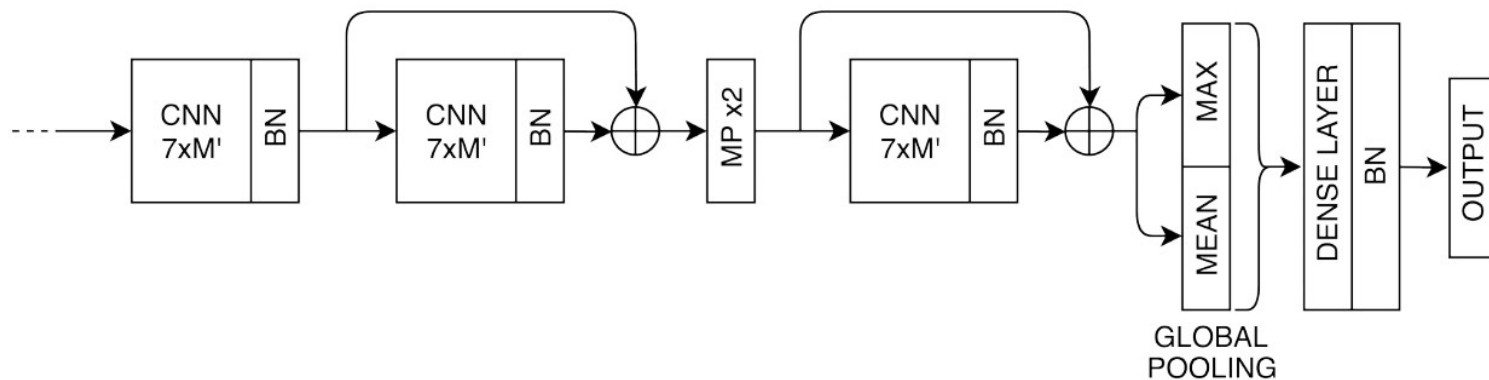
(Lee et al., 2017)



(Pons et al., 2016)



# Same back-end: to allow a fair comparison



MagnaTT  
**25k**  
songs

Million song dataset  
**250k**  
songs

**1M**  
songs

spectrograms > waveforms

MagnaTT  
**25k**  
songs

Million song dataset  
**250k**  
songs

**1M**  
songs

spectrograms ? waveforms

spectrograms > waveforms

MagnaTT  
**25k**  
songs

Million song dataset  
**250k**  
songs

**1M**  
songs

<i>Models</i>	<i>train size</i>	<i>ROC AUC</i>	<i>PR AUC</i>
Baseline	1.2M	91.61%	54.27%
Waveform	1M	<b>92.50%</b>	<b>61.20%</b>
Spectrogram	1M	92.17%	59.92%
Waveform	500k	91.16%	56.42%
Spectrogram	500k	91.61%	58.18%
Waveform	100k	90.27%	52.76%
Spectrogram	100k	90.14%	52.67%

waveforms > spectrograms

spectrograms > waveforms

MagnaTT  
**25k**  
songs

Million song dataset  
**250k**  
songs

**1M**  
songs

Which deep learning architectures are most appropriate for (music) audio signals?

In which scenarios is waveform-based end-to-end learning feasible?

How much data is required for carrying out competitive deep learning research?

- Musically Motivated CNNs for music tagging (Chapter III)
- Non-trained CNNs for music and audio tagging (Chapter IV)
- Music tagging at scale (Chapter V)
- Audio tagging with few training data (Chapter VI)
- Conclusions (Chapter VII)

Which deep learning architectures are most appropriate for (music) audio signals?

In which scenarios is waveform-based end-to-end learning feasible?

How much data is required for carrying out competitive deep learning research?

- Musically Motivated CNNs for music tagging (Chapter III)
- Non-trained CNNs for music and audio tagging (Chapter IV)
- Music tagging at scale (Chapter V)
- **Audio tagging with few training data (Chapter VI)**
- Conclusions (Chapter VII)



Which deep learning architectures are most appropriate for (music) audio signals?

In which scenarios is waveform-based end-to-end learning feasible?

**How much data is required for carrying out competitive deep learning research?**

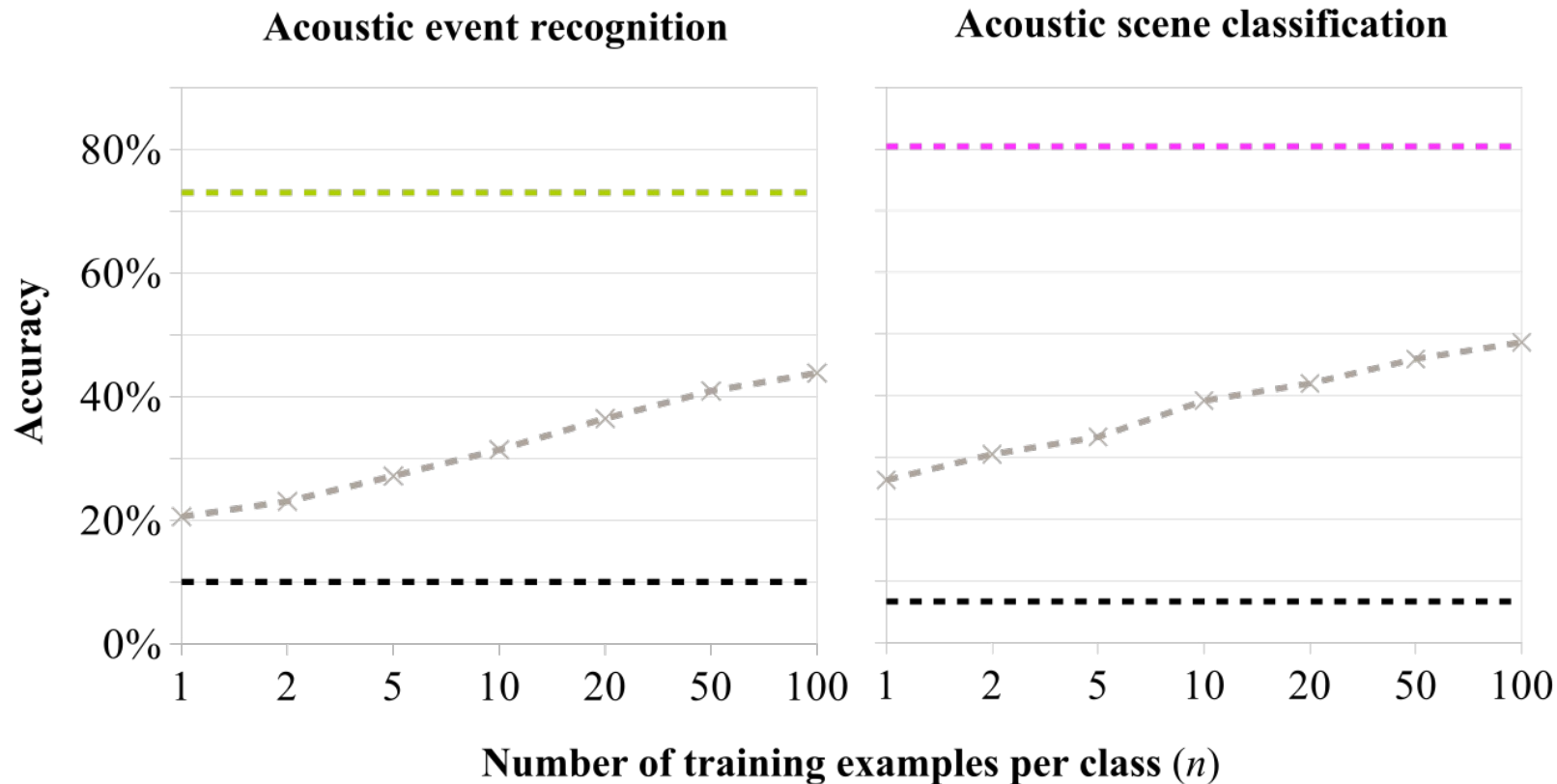
- Musically Motivated CNNs for music tagging (Chapter III)
- Non-trained CNNs for music and audio tagging (Chapter IV)
- Music tagging at scale (Chapter V)
- **Audio tagging with few training data (Chapter VI)**
- Conclusions (Chapter VII)

# Audio tagging with few data: how?

- **Strong regularization**
  - Will show the limitations of the standard deep learning pipeline
- **Prototypical networks**
  - A distance-based classifier that operates over a learn latent space
- **Transfer learning**
  - Enables to leverage external sources of audio data

# Methodology

The MFCC's + nearest neighbor baseline case



Regularized models

Prototypical networks

Transfer learning

Regularized models

Prototypical networks

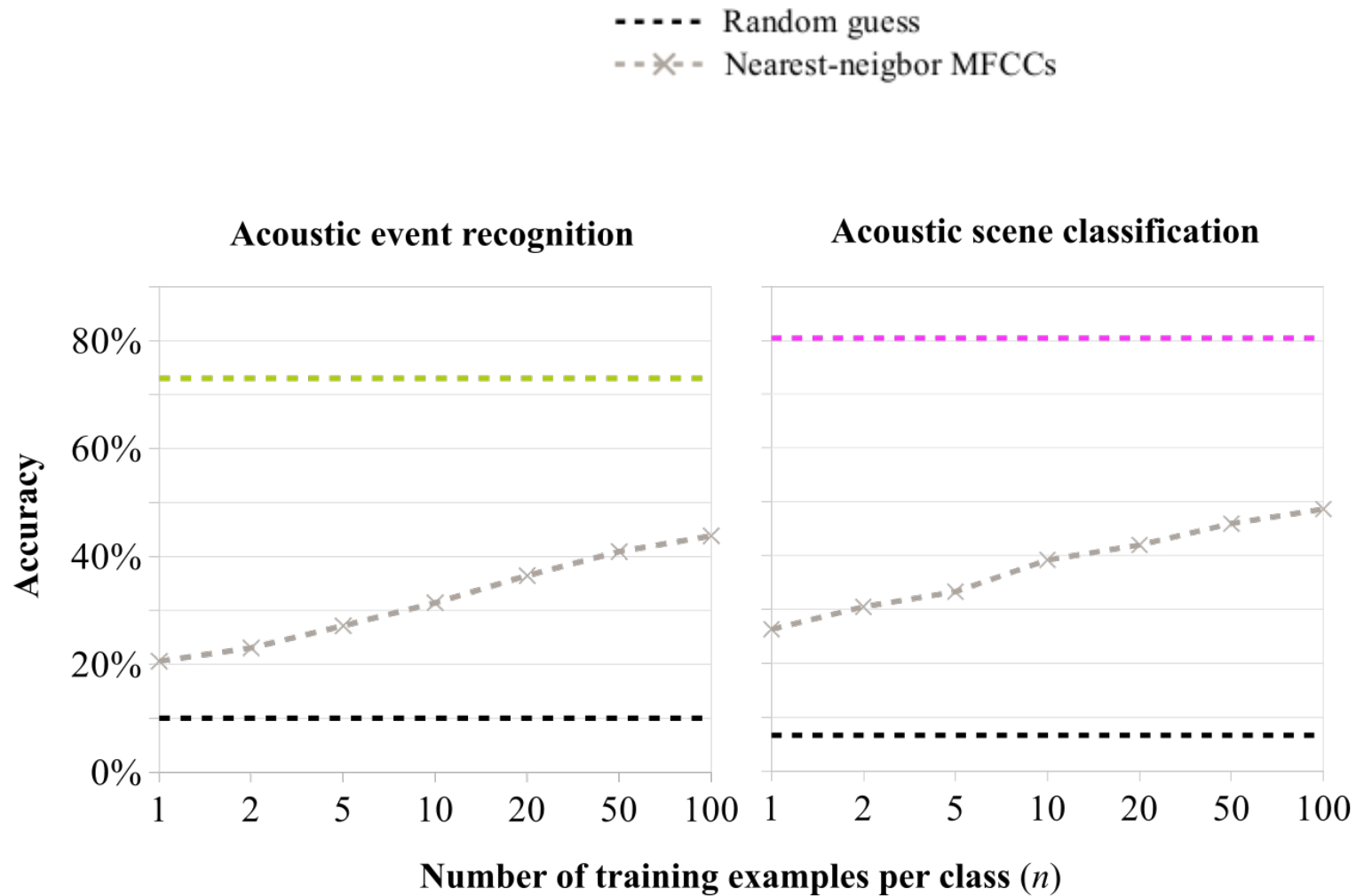
Transfer learning

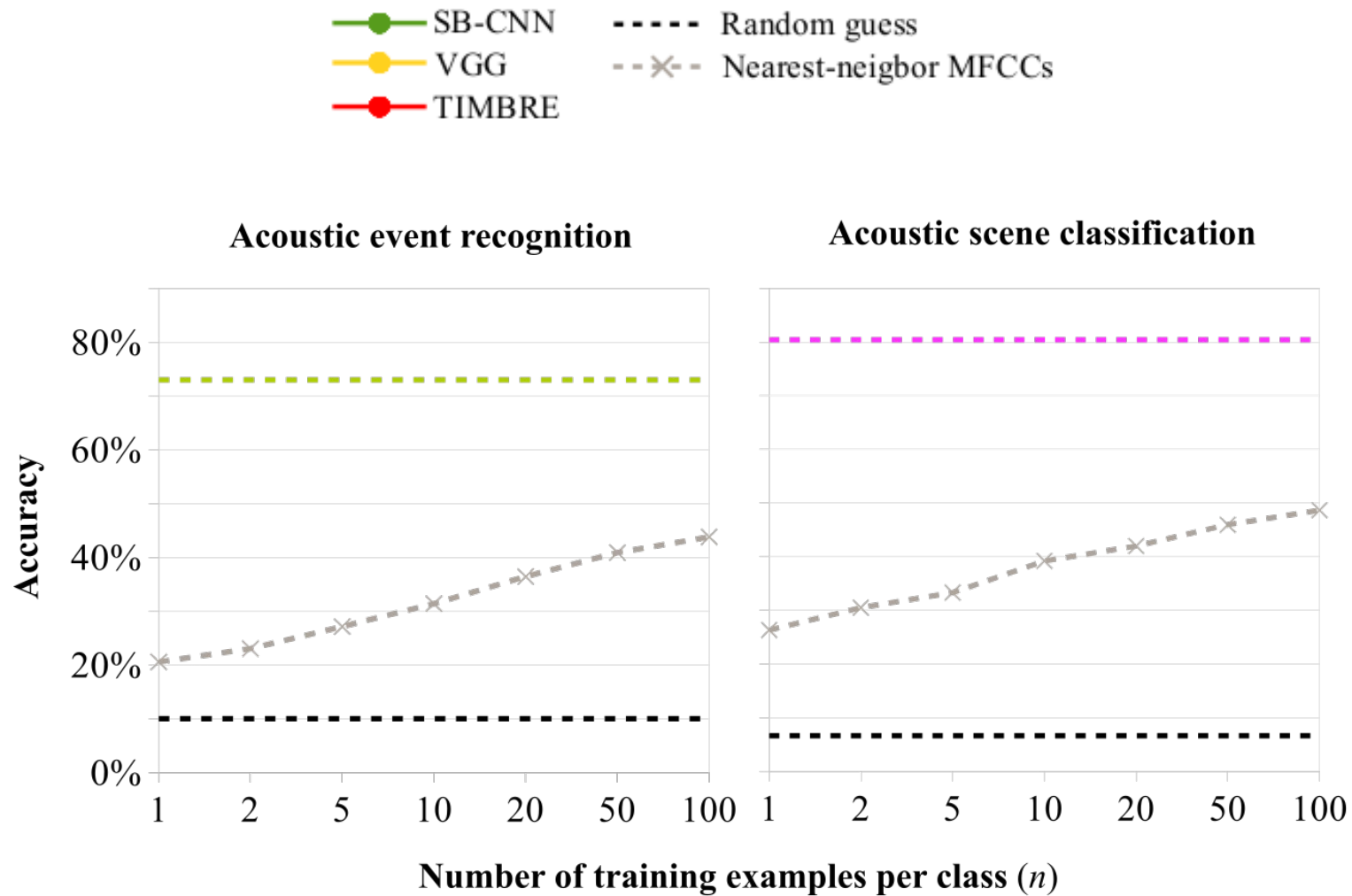
## **Regularized models**

# Regularized models

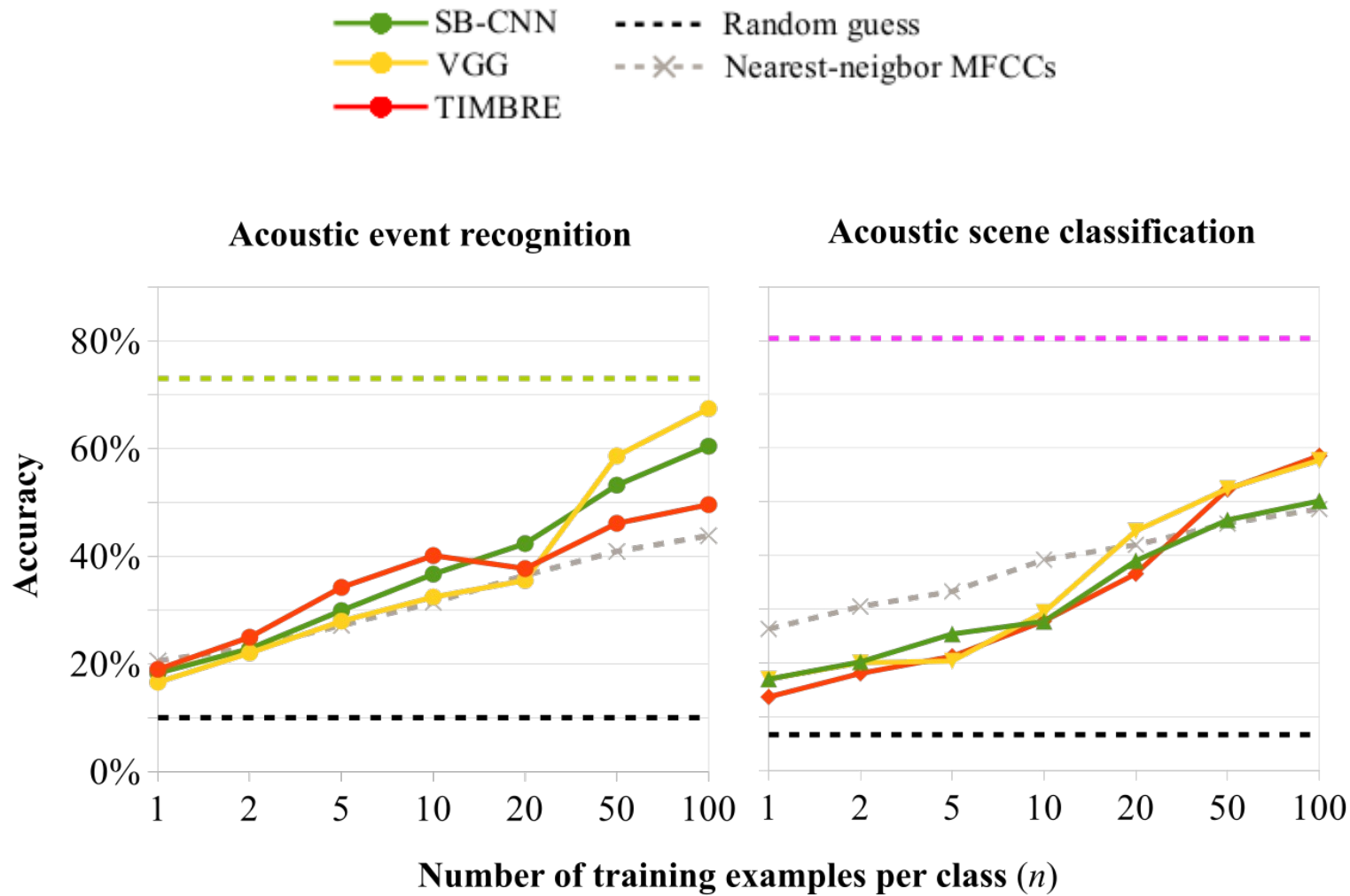
**Input:** log-mel spectrogram of 128 bins x 3 sec (128 frames)

- **SB-CNN: 250k** parameters
  - Inspired by AlexNet's computer vision architecture
  - *3 CNN layers (5x5) with max-pool + dense layer + softmax*
- **VGG: 50k** parameters
  - yet another computer vision architecture
  - *5 CNN layers (3x3) with max-pool (2x2) + softmax*
- **TIMBRE: 10k** parameters
  - The smallest CNN one can imagine for learning timbral traces
  - *1 CNN layer (vertical filters 108x7) with maxpool + softmax*









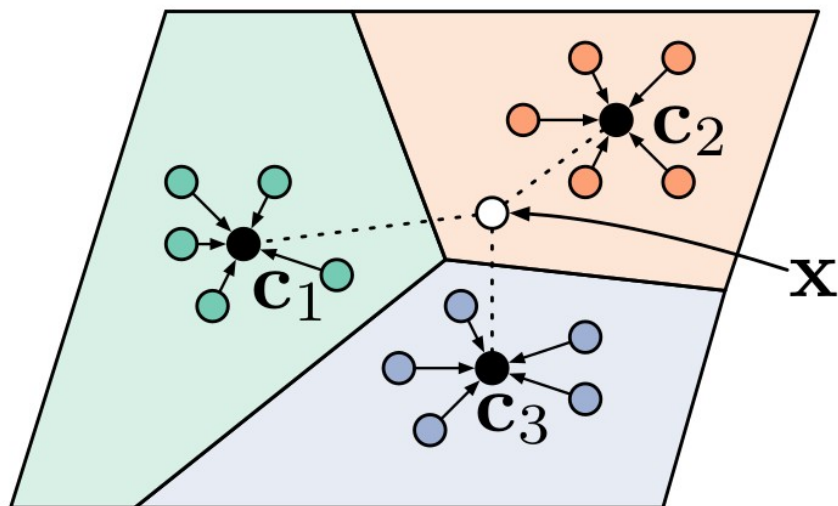
Regularized models

Prototypical networks

Transfer learning

# Prototypical networks

# Prototypical networks



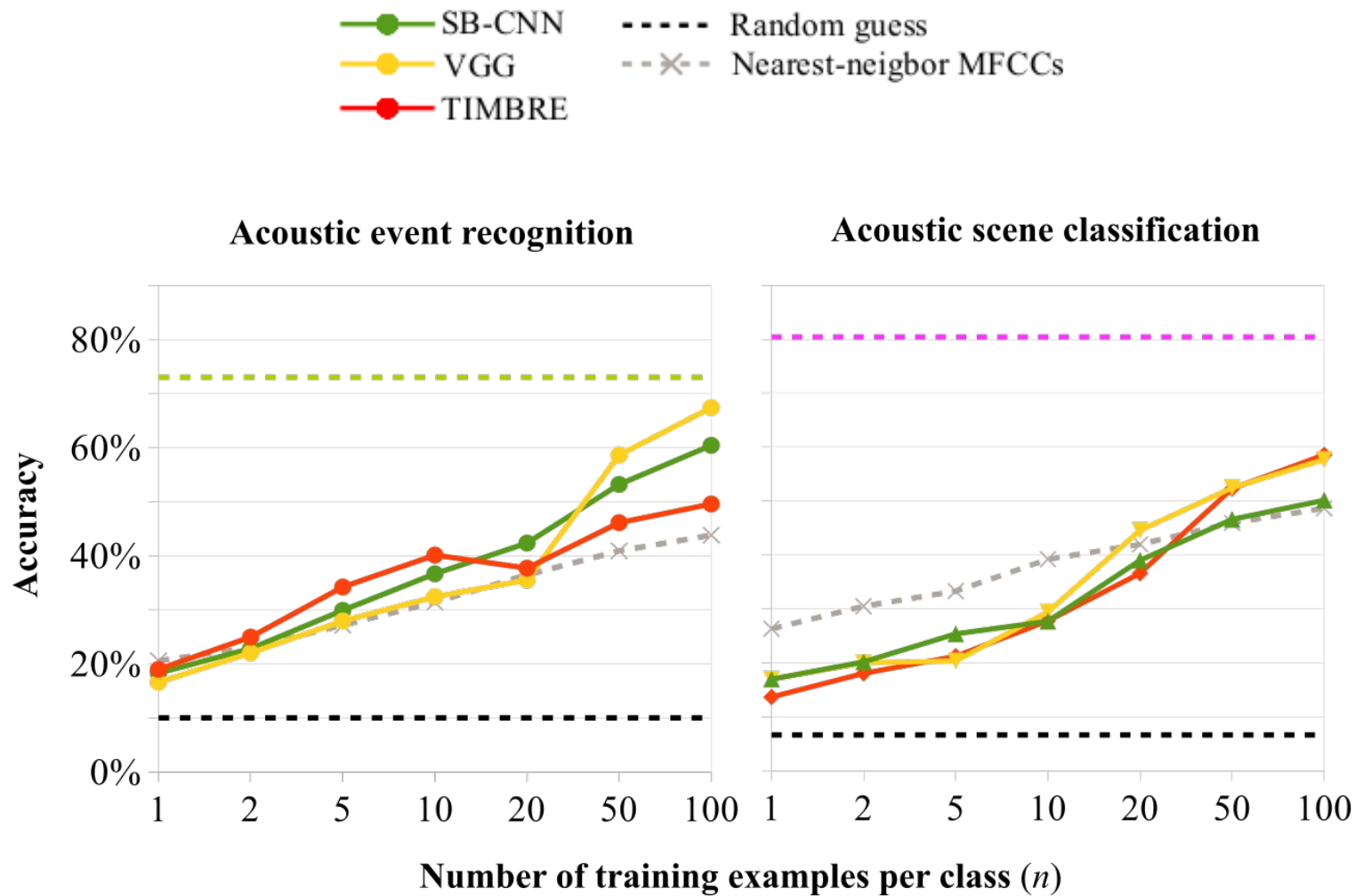
In our experiments:  
a VGG parametrizes  $f_\phi(\cdot)$

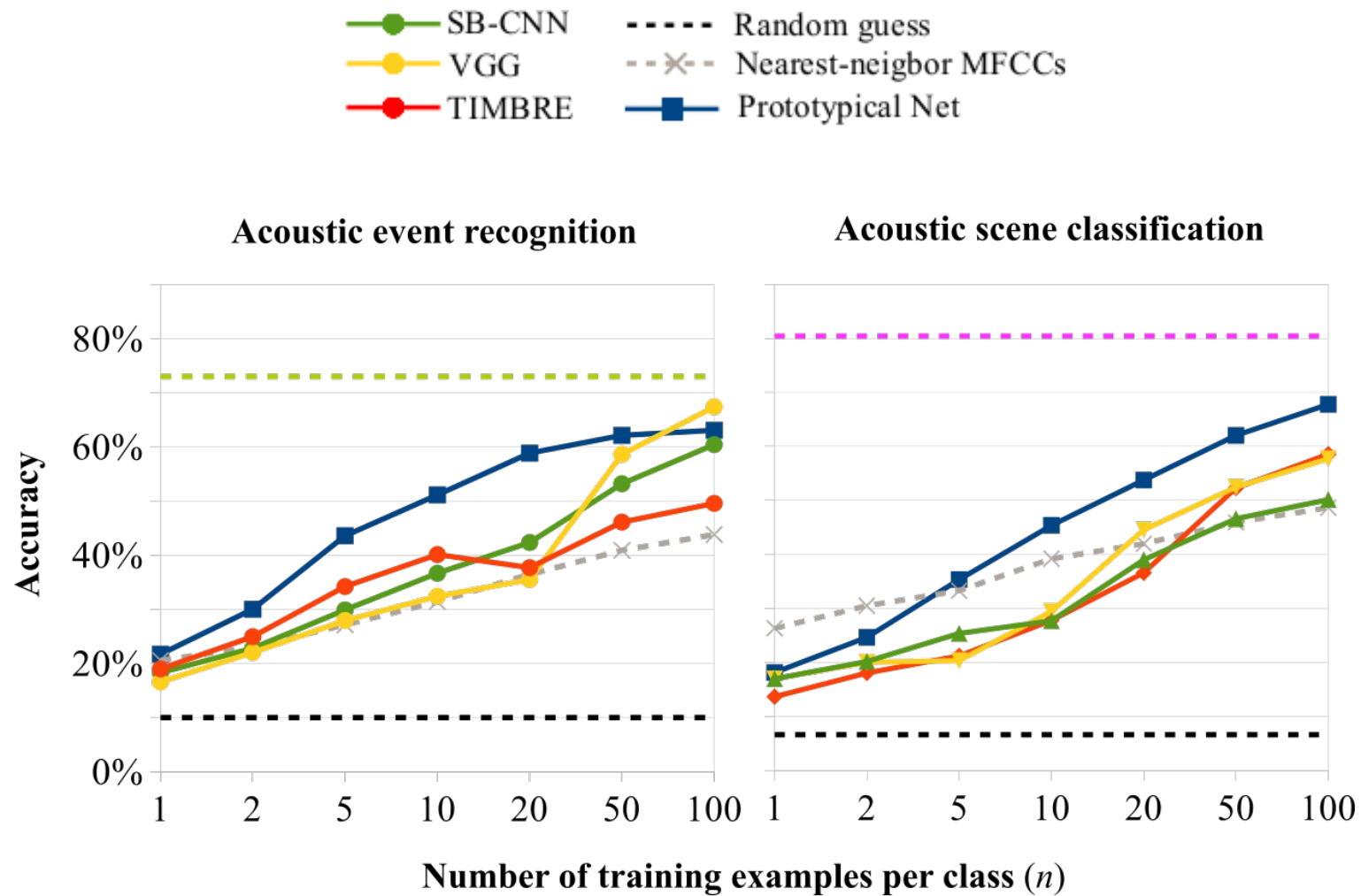
**0.** Compute a prototype per class ( $k$ ):

$$c_k = \mu_k = \frac{1}{|S_k|} \sum_{x_i \in S_k} f_\phi(x_i)$$

**1. Learning  $f_\phi(\cdot)$ :** to separate classes in the **embedding space of size 10**.

**2. Classification:** distribution based on a softmax over distances to the prototypes in the embedding space.





Regularized models

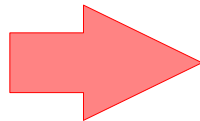
Prototypical networks

Transfer learning

# **Transfer learning**

# Transfer learning

**pretrain with  
source task**



**finetune with  
target task(s)**

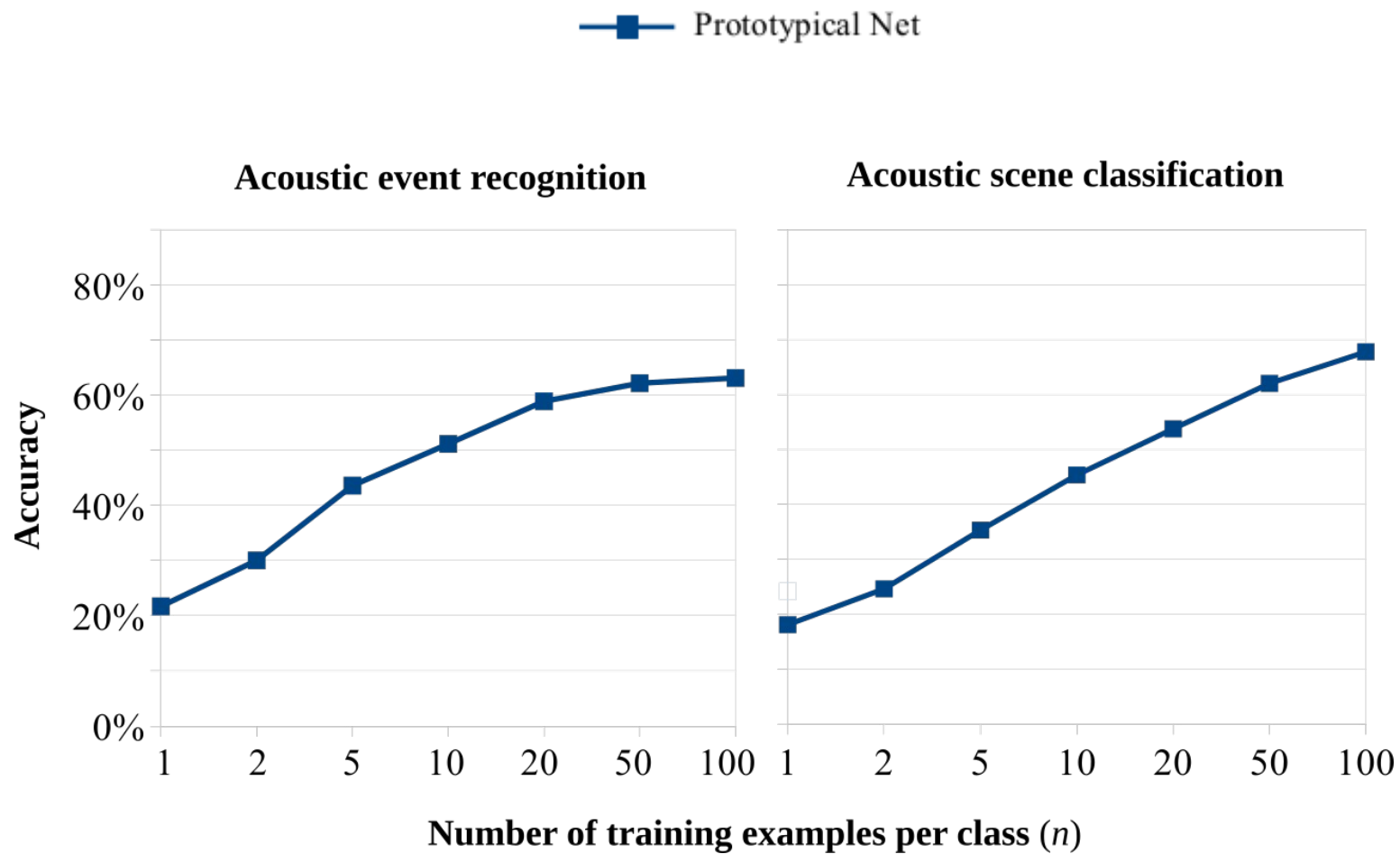
**AudioSet dataset**  
(acoustic event recognition)  
2M Youtube audios

**US8K dataset**  
(acoustic event recognition)

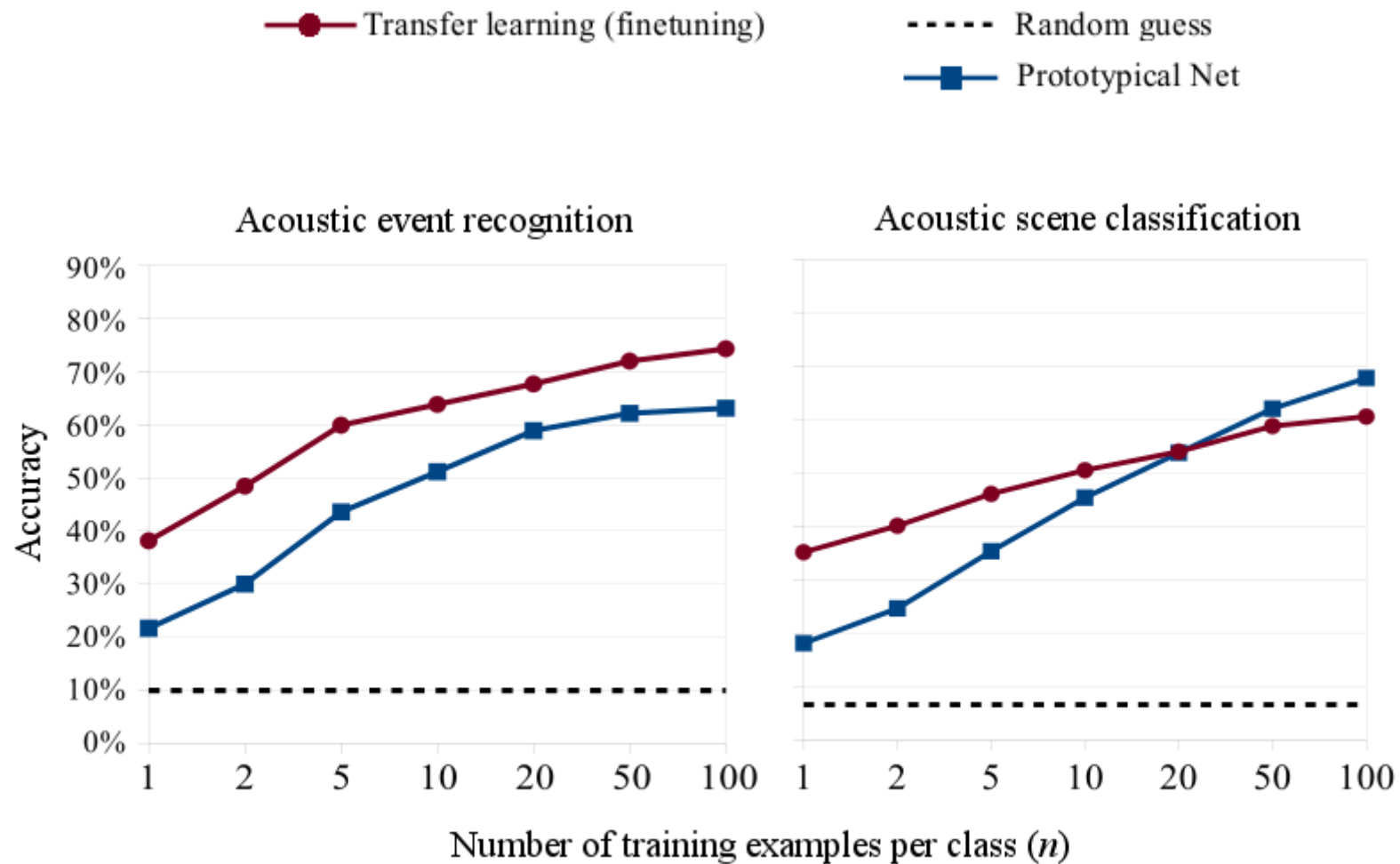
**ASC-TUT dataset**  
(acoustic scene classification)

Pre-trained **VGGish** on AudioSet:  
6 CNN layers (3×3)  
with max-pool layers (2×2) +  
3 dense layers (4096, 4096, 128)

**Finetuning of classifier:**  
dense softmax layer







# Summary

- **Strong regularization**
  - To realize the limitations of the standard deep learning pipeline
- **Prototypical networks**
  - A distance-based classifier that operates over a learn latent space
  - Particularly useful when
    - No additional “similar” data is accessible
- **Transfer learning**
  - Enables to leverage external sources of audio data

Which deep learning architectures are most appropriate for(music) audio signals?

In which scenarios is waveform-based end-to-end learning feasible?

How much data is required for carrying out competitive deep learning research?

- Musically Motivated CNNs for music tagging (Chapter III)
- Non-trained CNNs for music and audio tagging (Chapter IV)
- Music tagging at scale (Chapter V)
- Audio tagging with few training data (Chapter VI)
- Conclusions (Chapter VII)

Which deep learning architectures are most appropriate for(music) audio signals?

In which scenarios is waveform-based end-to-end learning feasible?

How much data is required for carrying out competitive deep learning research?

- Musically Motivated CNNs for music tagging (Chapter III)
- Non-trained CNNs for music and audio tagging (Chapter IV)
- Music tagging at scale (Chapter V)
- Audio tagging with few training data (Chapter VI)
- **Conclusions (Chapter VII)**

# Research question I

**Which deep learning architectures are most appropriate for (music) audio signals?**

## **Music tagging:**

- Musically motivated CNNs perform similarly (if not better) than its counterparts
- Intuitive design strategy that allows interpretable CNNs
- It allows designing compact CNNs

## **Audio tagging:**

- A computer vision architecture, VGG, achieves the best results
- Potentially because is flexible and general audio is very diverse

## Research question II

### In which scenarios is waveform-based end-to-end learning feasible?

**Initial hypothesis:** when large computing power and big training datasets are accessible.

- Large datasets are required for waveform-based > spectrogram-based ones.
- With the appropriate methodology, one can do conclusive research with small datasets and with not much hardware resources.

## Research question III

**How much data is required for carrying out competitive deep learning research?**

- Large datasets are required for developing state-of-the-art models.
- With the appropriate methodology, one can do conclusive research with small datasets and with not much hardware resources.

# Publications, code & awards

- Jordi Pons & Xavier Serra. **musicnn: pre-trained convolutional neural networks for music audio tagging**. LBD-ISMIR, 2019.
  - <https://github.com/jordipons/musicnn>
- Jordi Pons, Joan Serrà & Xavier Serra. **Training neural audio classifiers with few data**. ICASSP, 2019.
  - Oral presentation.
  - <https://github.com/jordipons/neural-classifiers-with-few-audio>
- Jordi Pons & Xavier Serra. **Randomly weighted CNNs for (music) audio classification**. ICASSP, 2019.
  - <https://github.com/jordipons/elmarc>
- Jordi Pons, Oriol Nieto, Matthew Prockup, Erik M. Schmidt, Andreas F. Ehmann & Xavier Serra. **End-to-end learning for music audio tagging at scale**. ISMIR, 2018.
  - Best student paper award
  - <https://github.com/jordipons/music-audio-tagging-at-scale-models>
- Jordi Pons, Rong Gong & Xavier Serra. **Score-informed syllable segmentation for a capella singing voice with convolutional neural networks**. ISMIR, 2017.
  - <https://github.com/ronggong/jingjuSyllabicSegmentation>
- Jordi Pons, Olga Slizovskaia, Rong Gong, Emilia Gómez & Xavier Serra. **Timbre Analysis of Music Audio Signals with Convolutional Neural Networks**. EUSIPCO, 2017.
  - <https://github.com/jordipons/EUSIPCO2017>
  - Oral presentation
- Jordi Pons & Xavier Serra. **Designing efficient architectures for modeling temporal features with convolutional neural networks**. ICASSP, 2017.
  - <https://github.com/jordipons/ICASSP2017>
- Jordi Pons, Thomas Lidy & Xavier Serra. **Experimenting with musically motivated convolutional neural networks**. CBMI, 2016.
  - Best paper award
  - <https://github.com/jordipons/CBMI2016>



Pronounced as "musician", musicnn is a set of pre-trained deep convolutional neural networks for music audio tagging.

[Edit](#)[Manage topics](#)

📦 369 commits

🌿 1 branch

📦 0 releases

👤 1 contributor

📦 ISC

Branch: master ▾

[New pull request](#)[Create new file](#)[Upload files](#)[Find file](#)[Clone or download ▾](#)

jordipons Update extractor.py

Latest commit 96dd51f 26 days ago

[audio](#)

Notebook functioning with basic demo

4 months ago

[images](#)

Debugging functions, and renaming tagging\_example.ipynb

3 months ago

[musicnn](#)

Update extractor.py

26 days ago

[.gitignore](#)

Adding new MSD model and VGGs

4 months ago

[DOCUMENTATION.md](#)

Update DOCUMENTATION.md

3 months ago

[FAQs.md](#)

Update FAQs.md

2 months ago

[LICENSE.md](#)

Minor fixes for pushing to PyPI

4 months ago

[MANIFEST.in](#)

Update MANIFEST.in

3 months ago

[README.md](#)

Update README.md

3 months ago

[musicnn\\_example.ipynb](#)

Updating notebooks

3 months ago

[setup.py](#)

Ready to upload v.0.1.0 to PyPI

3 months ago

[tagging\\_example.ipynb](#)

Debugging functions, and renaming tagging\_example.ipynb

3 months ago

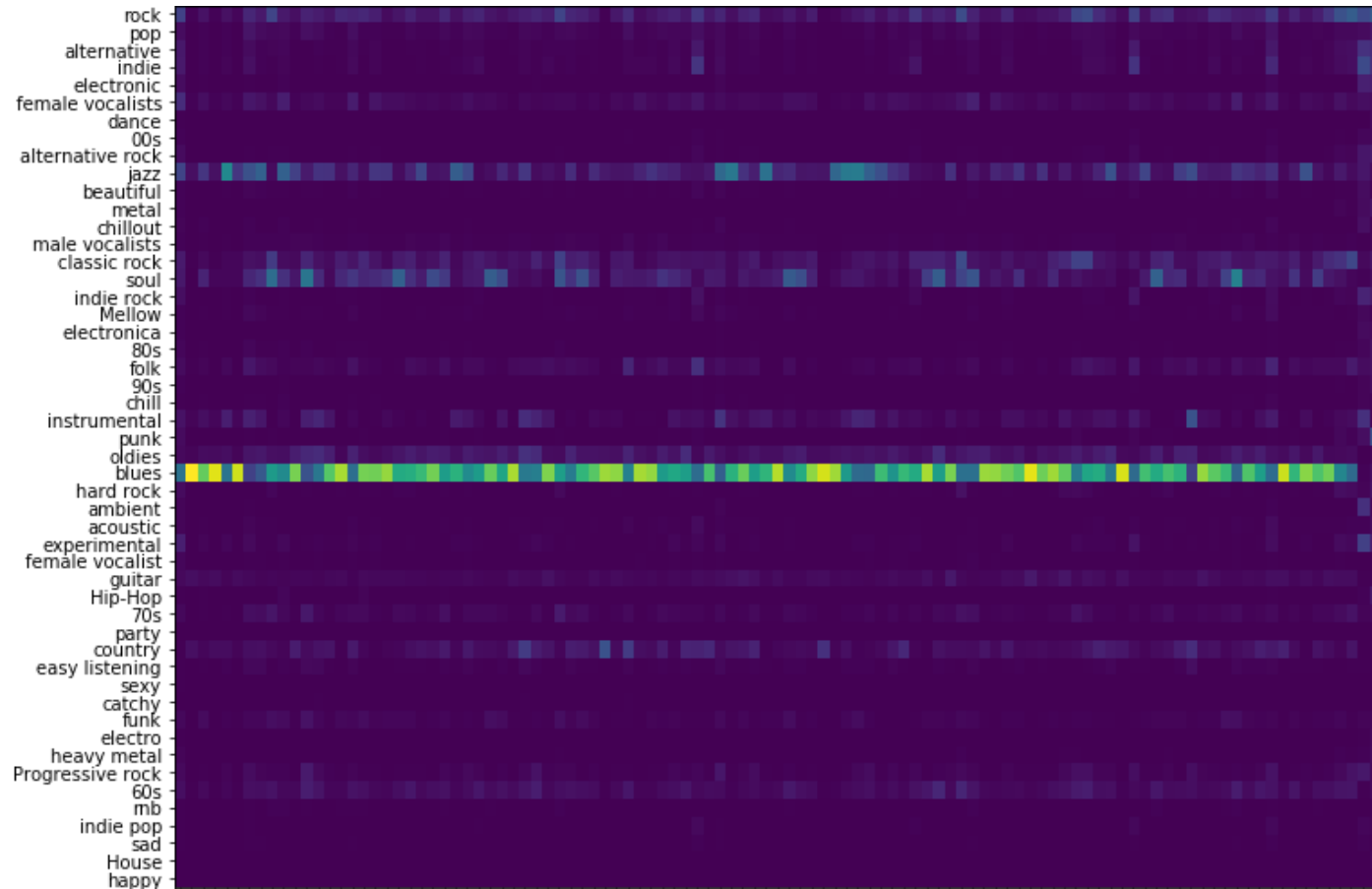
[vgg\\_example.ipynb](#)

Updating notebooks

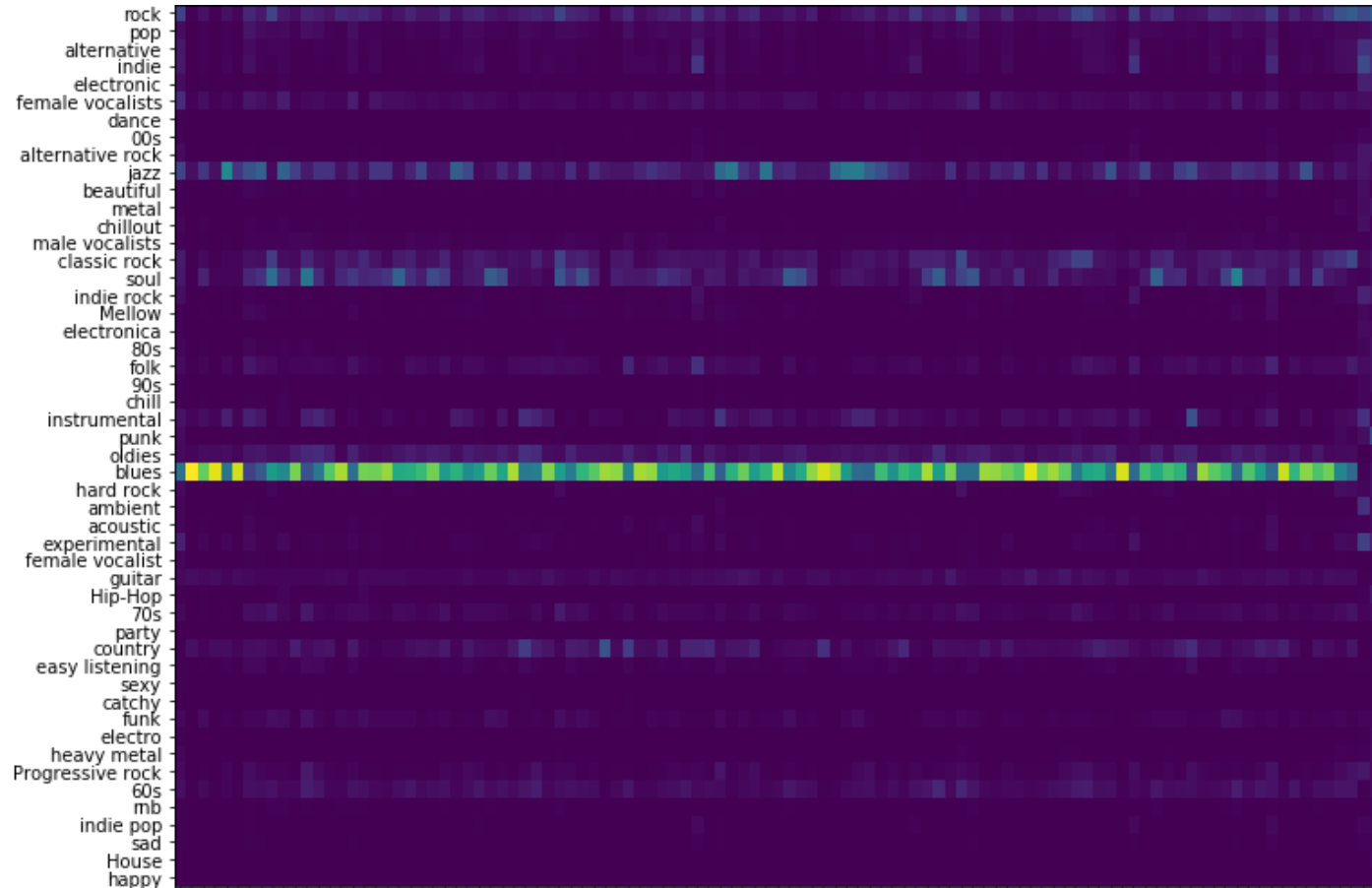
3 months ago

pip install musicnn

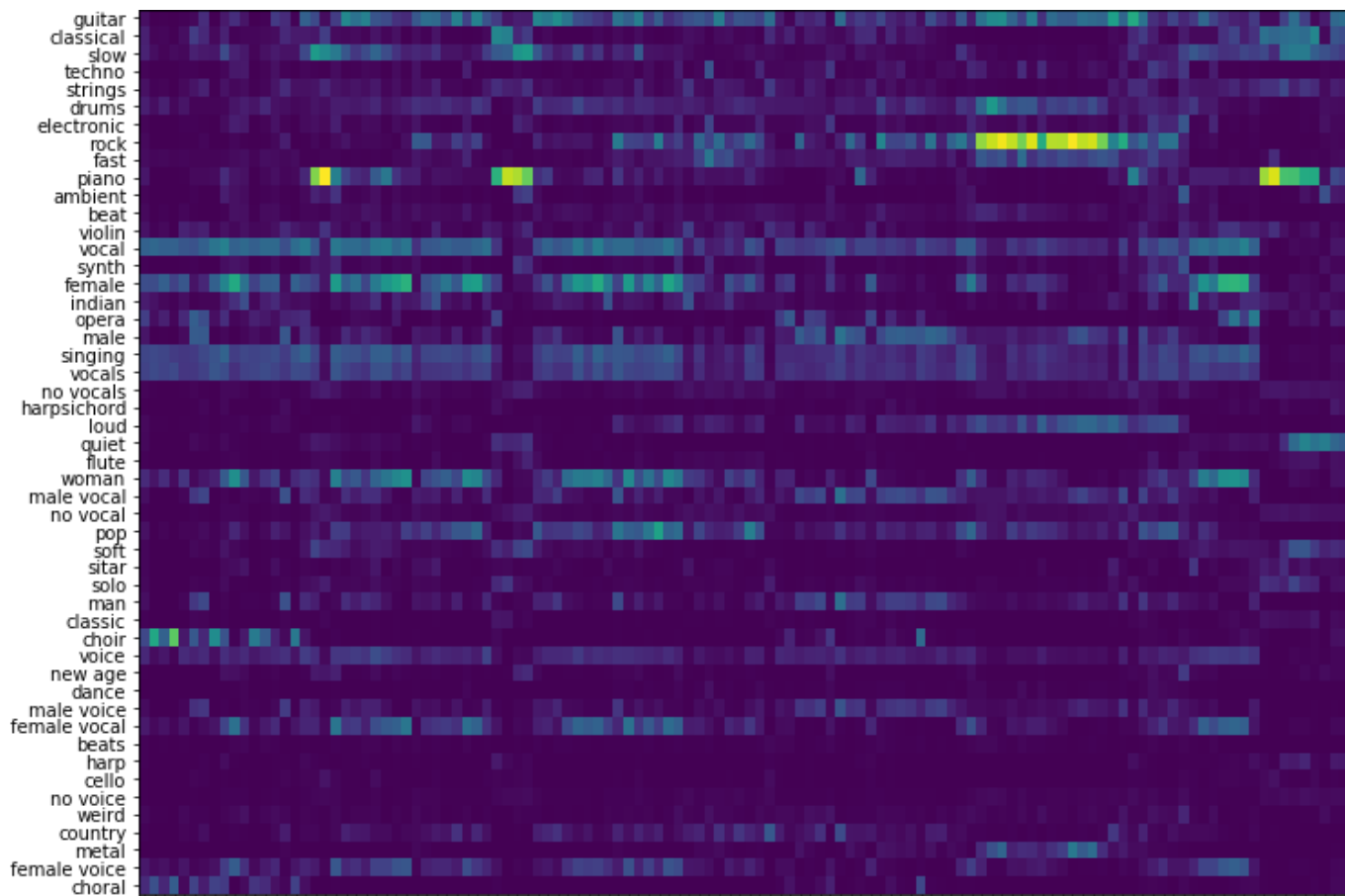
# Muddy Waters: Screamin and Cryin'



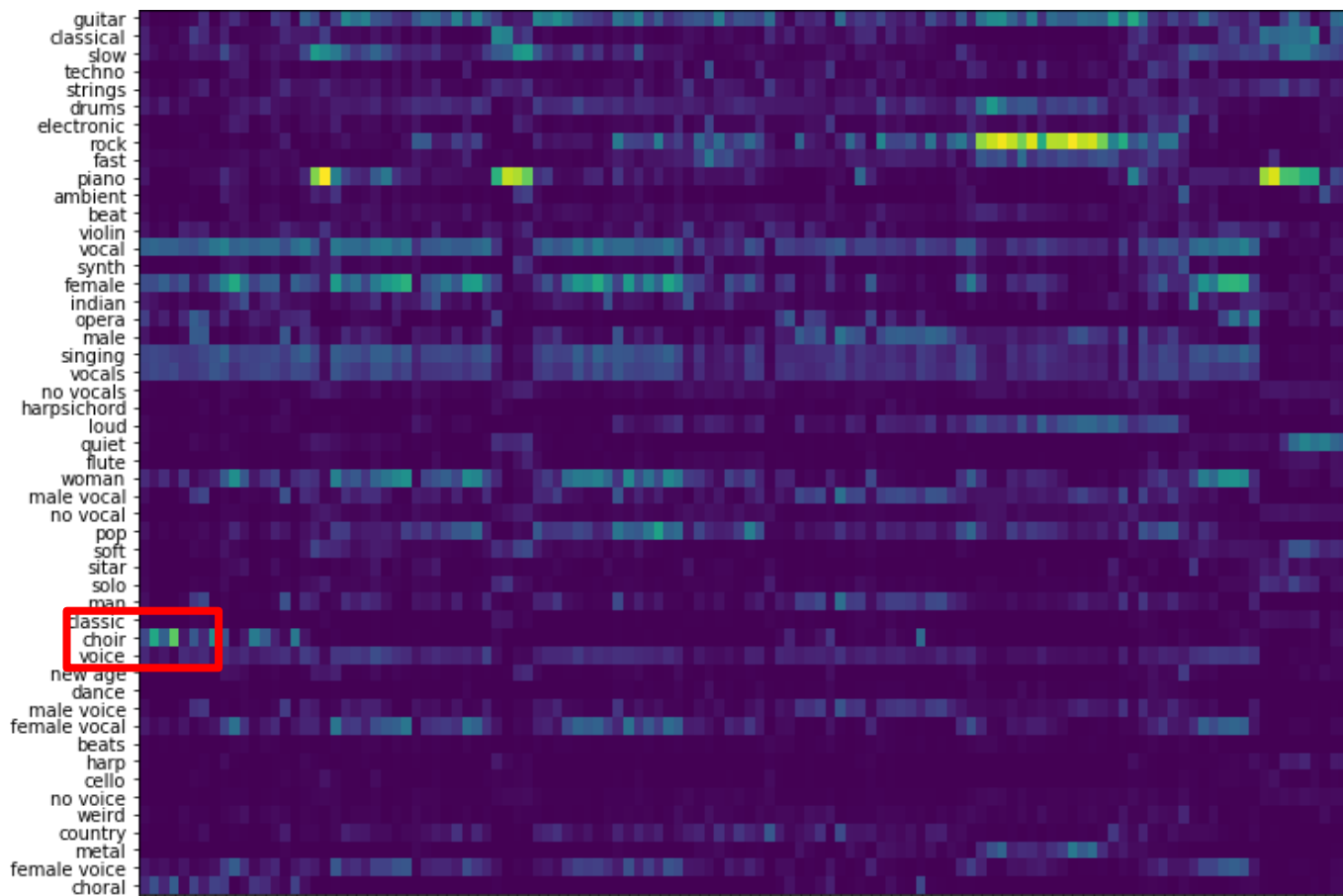
# Muddy Waters: Screamin and Cryin'



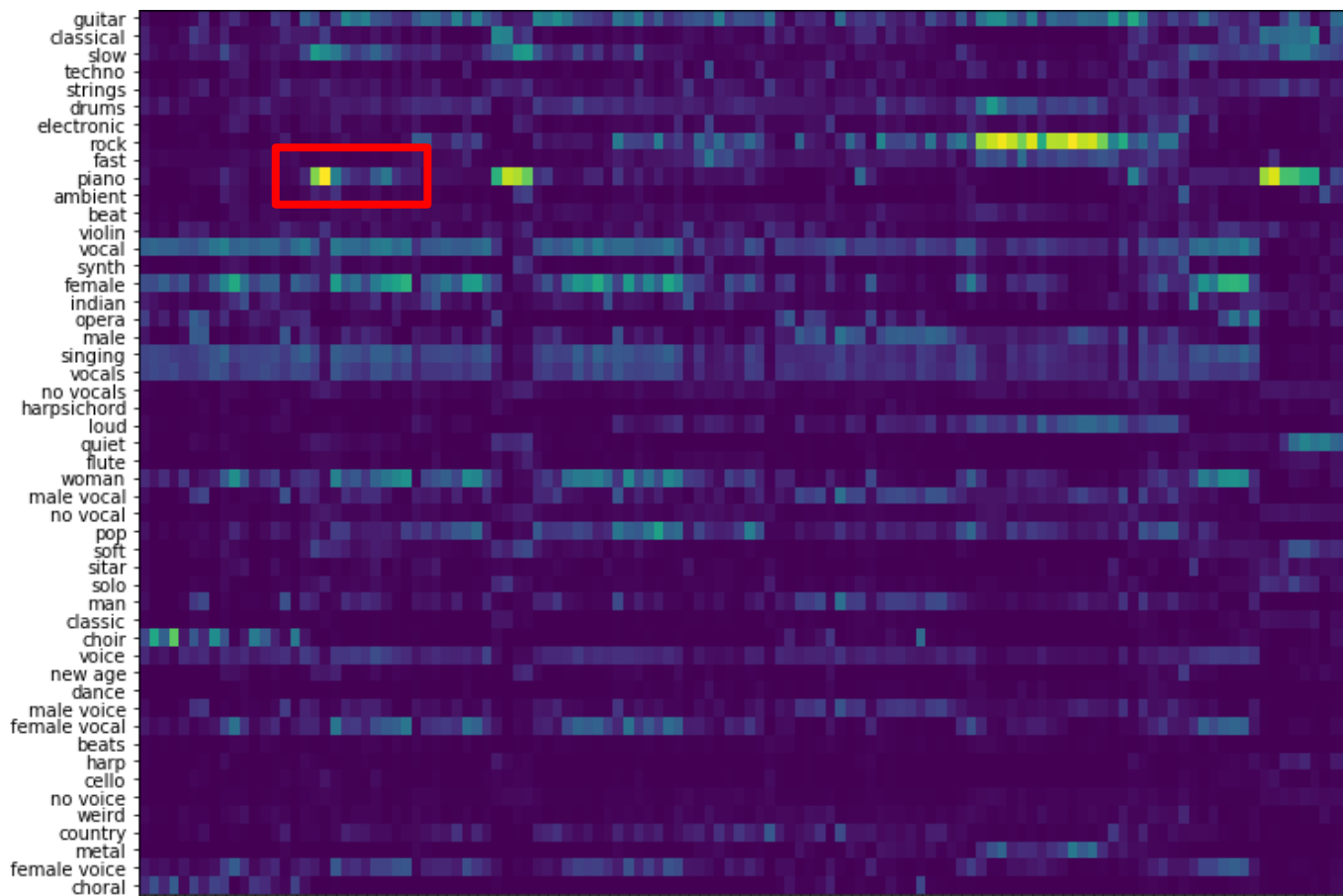
# Queen: Bohemian Rhapsody



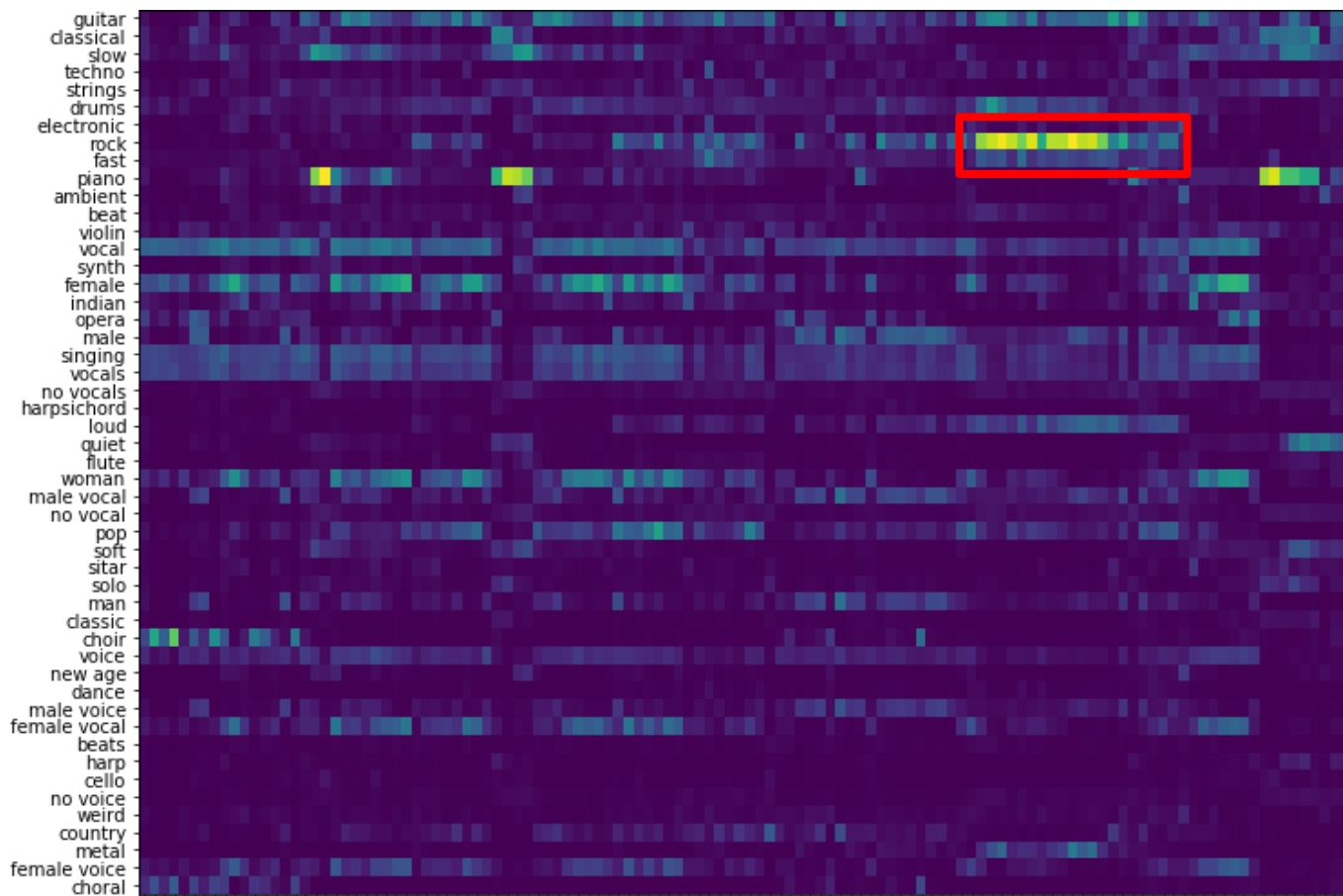
# Queen: Bohemian Rhapsody



# Queen: Bohemian Rhapsody



# Queen: Bohemian Rhapsody



# Thank you, and also thanks to all my collaborators!



**M**usic  
**T**echnology  
**G**roup



EXCELENCIA  
MARÍA  
DE MAEZTU

**pandora**®

*Telefónica*



# Correspondences between trained and non-trained CNNs

- **Waveform front-ends:** sample-level  $\gg$  frame-level many  $>$  frame-level
  - (Lee et al., 2017): the original sample-level CNN paper results.
  - (Pons et al, 2018): at Pandora I was informally experimenting with those.
  - (van den Oord, 2016): the original Wavenet is a sample-level CNN.
- **Spectrogram front-ends:** allowing pitch-shifting is beneficial ( $7 \times 86 > 7 \times 96$ )
  - (Pons et al, 2016): We explicitly measured this trend.
  - (Oramas et al, 2017): They also explicitly measured this trend.
- **Music tagging:** using prior music domain knowledge can be useful
- **Audio tagging:** the VGG, a computer vision architecture, achieves the best results