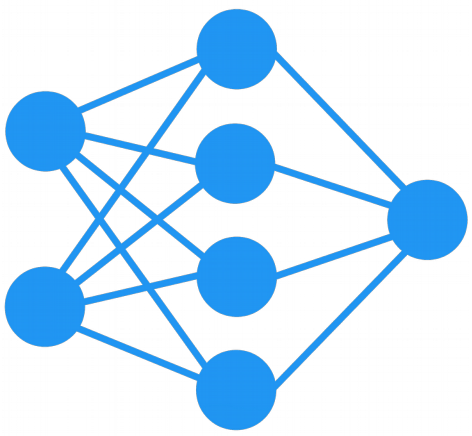
Deep learning in audio research



Igor Kostiuk | 2017

What's the plan?

- Tasks for AI
- Data and where to find it
- Which data is better
- NN architecture comparison
- Links

Audio related AI tasks

- music auto tagging
- urban sound classification
- keyword spotting
- user identification
- etc.

→ in recommender system engines

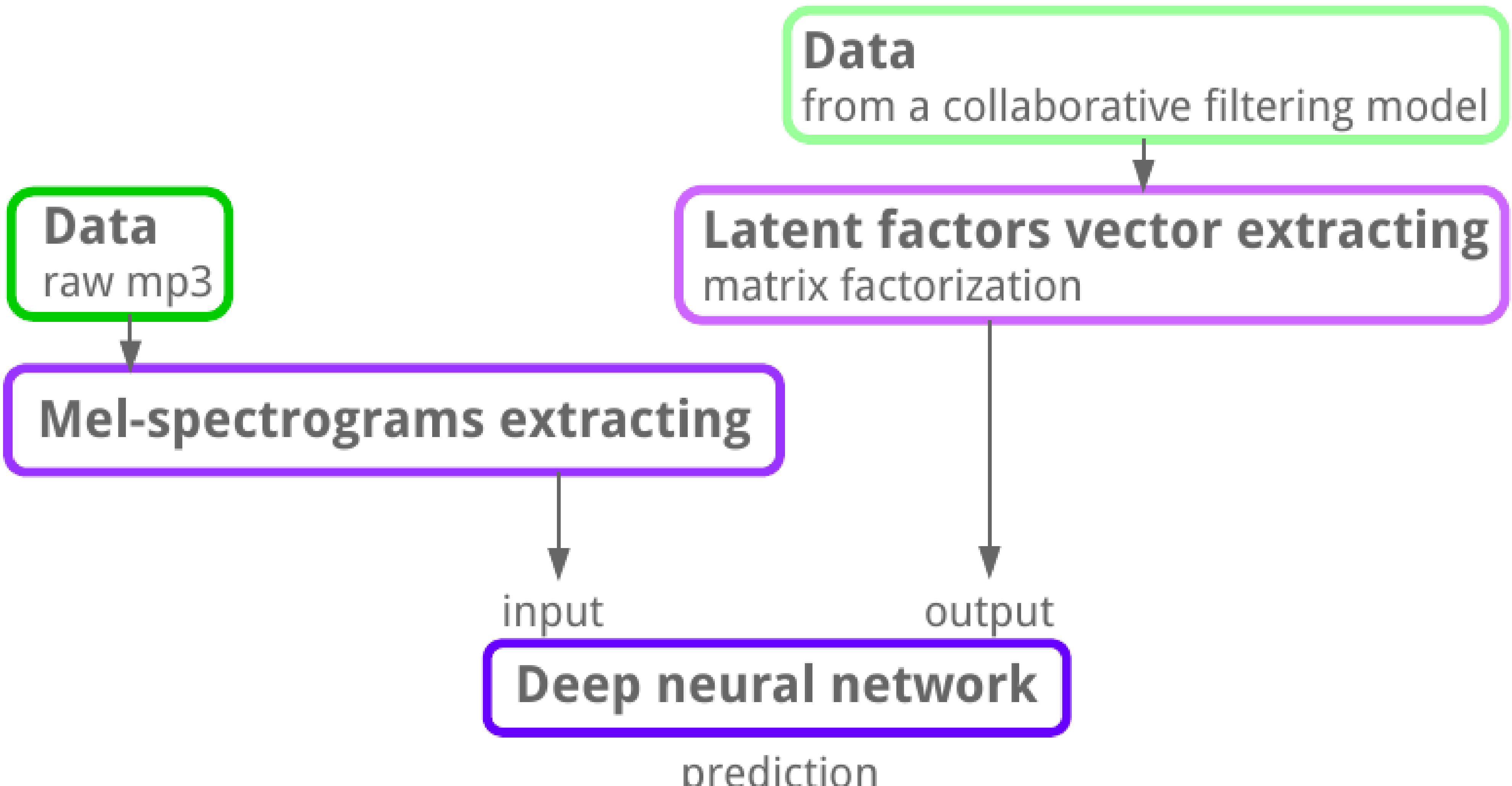
→ in smart phones to detect the environment

→ in smart house systems

Binary
Jay-Z
The Notorious B.I.G.
Ratatat
Lil Scrappy
Company Flow
Kanye West
Slum Village
Ignition
DMX / Obie Trice
Ying Yang Twins ft. Pitbull
Wax Tailor
Common
Black Eyed Peas
Collie Buddz
DMX
Girl Talk
Linkin Park
Rick Ross
Jay-Z
Buju Banton
California Swag District
Common / Kanye West
Prince & The New Power Generation
Eminem / Dina Rae
Daddy Yankee / Randy
Gym Class Heroes
Aerosmith
Eric Clapton
Warren Zevon
Lonesome River Band
Love
Fleetwood Mac
The Velvet Underground
The Police
Kings Of Leon
Gipsy Kings
Ruben Blades
Barry Manilow
Estopa
The Police
Bee Gees
The Four Seasons
Weezer
Crosby, Stills, Nash & Young
The Black Keys
PAULA COLE
The Monkees
Mannie Fresh / Lil Wayne
Beyonc
Sugarland
Taylor Swift
Leona Lewis
Kelly Clarkson
Avril Lavigne
Ashlee Simpson
Lil Wayne / Shanell
Miley Cyr
Mariah Carey
Britney Spears
Owl City
Mariah Carey
Madonna
Aqua
DeBarge
DeLuna
Portishead
Lindsay Lohan
Rudeboy
Timbaland / Keri Hilson
Miike Snow
Cut Copy
Basshunter
Chromeo
Daft Punk
Basshunter
Alaska Y Dinarar
LCD Soundsys
Imogen Heap
Vitalic
Eric Prydz
Irene Cara
Gorillaz
Filo + Peri
Airwave
Gielen
DHT Feat. Edme
Perfect Stranger
Cut Copy
The Prodigy
Sub Focus

Deep Learning approach

Predicting listening preferences from audio signals by training a regression model to predict the latent representations of songs that were obtained from a collaborative filtering model.



Data

- 101 - Dog
- 102 - Rooster
- 103 - Pig
- 104 - Cow
- 105 - Frog
- 106 - Cat
- 107 - Hen
- 108 - Insects
- 109 - Sheep
- 110 - Crow
- 201 - Rain
- 202 - Sea waves
- 203 - Crackling fire
- 204 - Crickets
- 205 - Chirping birds
- 206 - Water drops
- 207 - Wind
- 208 - Pouring water
- 209 - Toilet flush
- 210 - Thunderstorm
- 301 - Crying baby
- 302 - Sneezing

https://github.com/karoldvl/ESC-50

https://serv.cusp.nyu.edu/projects/urbansound dataset/urbansound8k.html

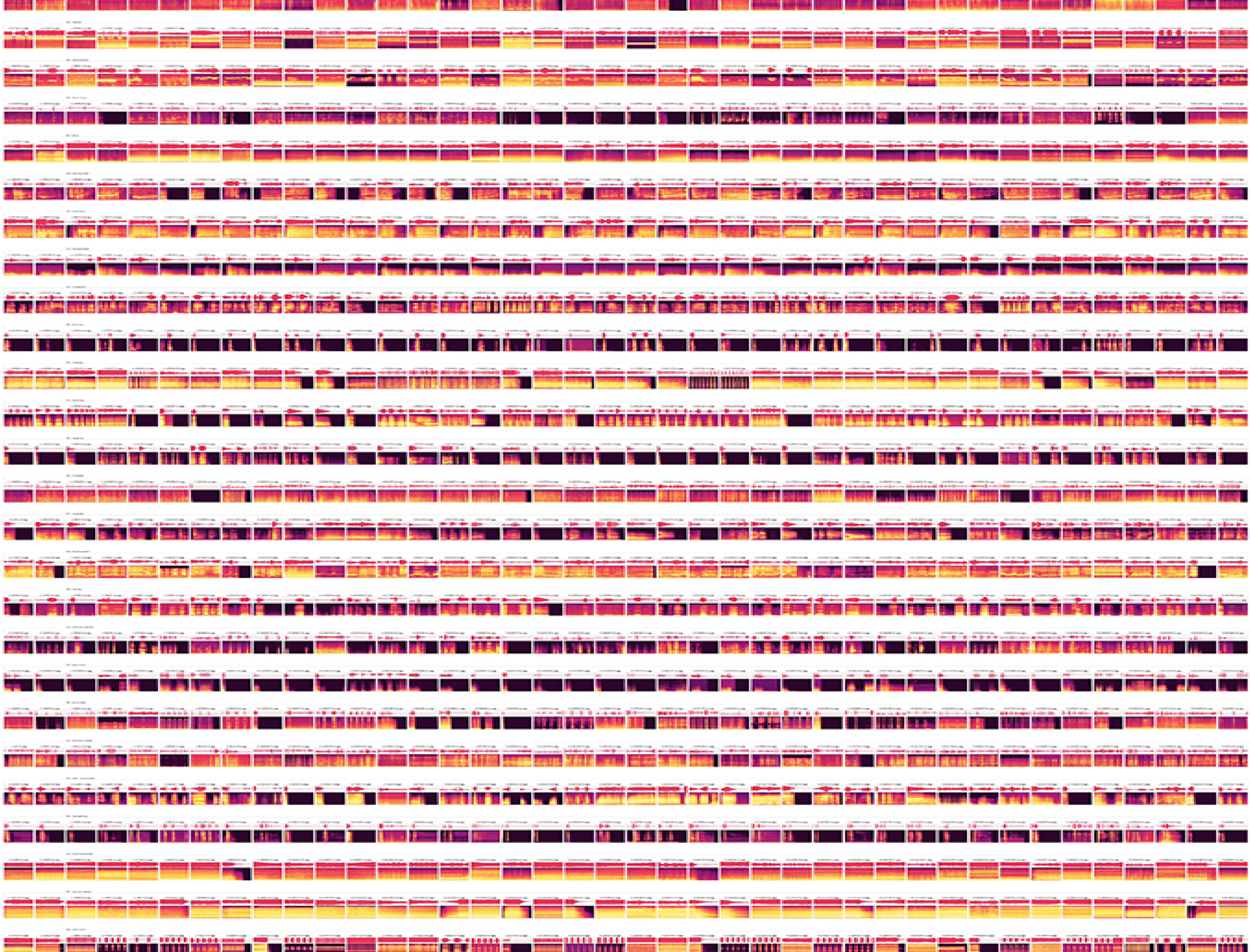
https://labrosa.ee.columbia.edu/millionsong/

https://en.wikipedia.org/wiki/List_of_datasets _for_machine_learning_research#Sound_data

The **ESC-50** dataset is a public labeled set of 2000 environmental recordings (50 classes, 40 clips per class, 5 seconds per clip) suitable for environmental sound classification tasks.

Animals

- 101 - Dog
- 102 - Rooster
- 103 - Pig
- 104 - Cow
- 105 - Frog
- 106 - Cat
- 107 - Hen
- 108 - Insects (flying)
- 109 - Sheep
- 110 - Crow



Data

The Echo Nest Taste Profile Subset

http://labrosa.ee.columbia.edu/millionsong/tasteprofile

b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBSUJE12A6D4F8CF5	2
b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBVFZR12A6D4F8AE3	1
b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBXALG12A8C13C108	1
b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBXHDL12A81C204C0	1
b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBYHAJ12A6701BF1D	1
b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOCNMUH12A6D4F6E6D	1
b80344d063b5ccb3212f76538f3d9e43d87dca9e	SODACBL12A8C13C273	1
b80344d063b5ccb3212f76538f3d9e43d87dca9e	SODDNQT12A6D4F5F7E	5

Taste Profile subset is big. Some numbers:

1,019,318 unique users
384,546 unique MSD songs
48,373,586 user - song - play count triplets

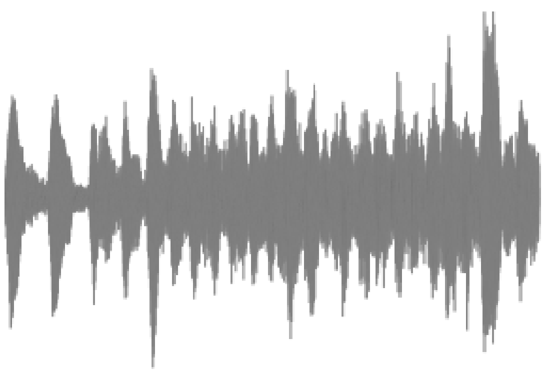
Data retrieval

https://www.7digital.com/

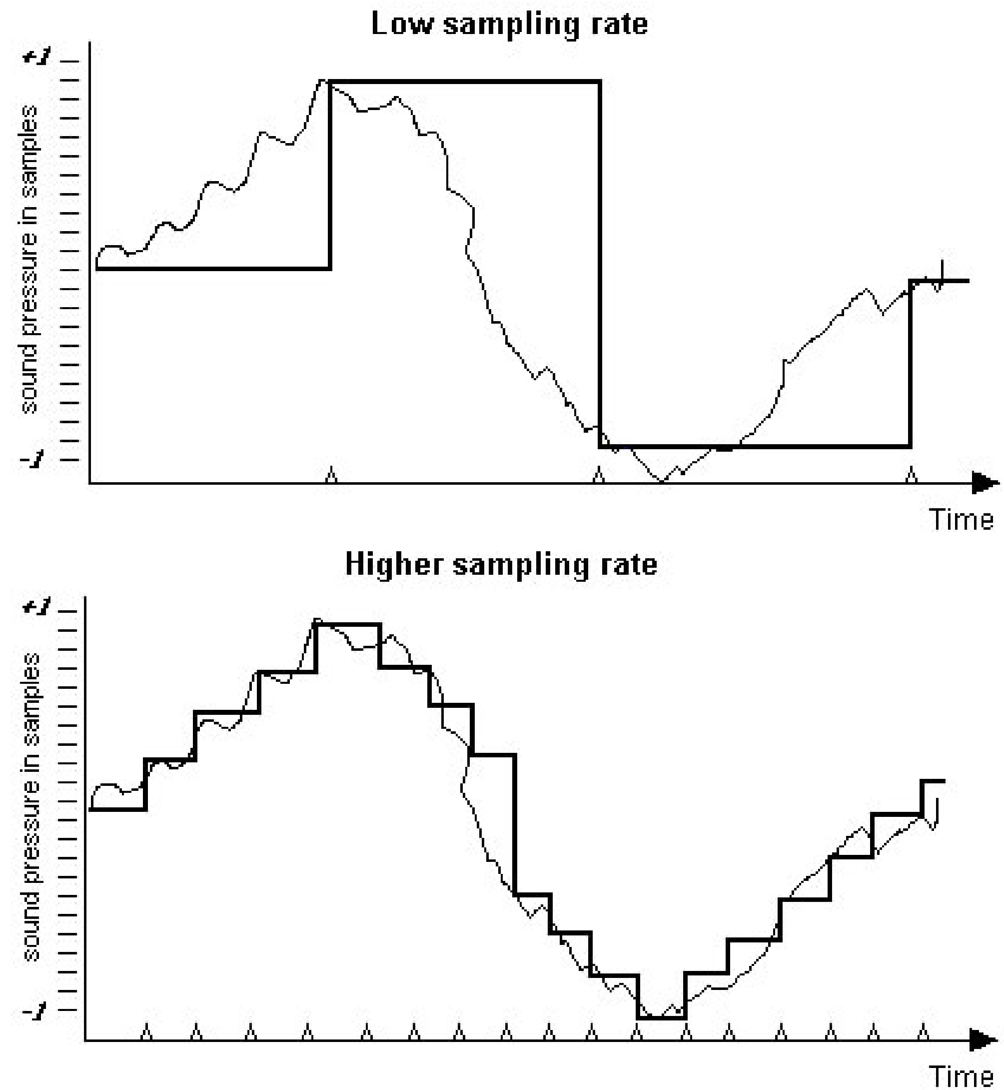
We are able to attain 29 second audio clips for over 99% of the dataset.

Original dataset has no raw audio, only precomputed, badly documented features.

Data



raw audio



Data

Sample rate	Quality level
11,025 Hz	Poor AM radio (low-end multimedia)
22,050 Hz	Near FM radio (high-end multimedia)
32,000 Hz	Better than FM radio (standard broadcast rate)
44,100 Hz	CD
48,000 Hz	Standard DVD
96,000 Hz	High-end DVD

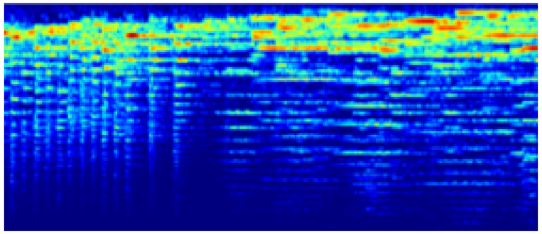
Data

A mel-spectrograms is a kind of time-frequency representation.

It is obtained from an audio signal by computing the Fourier transforms of short, overlapping windows.

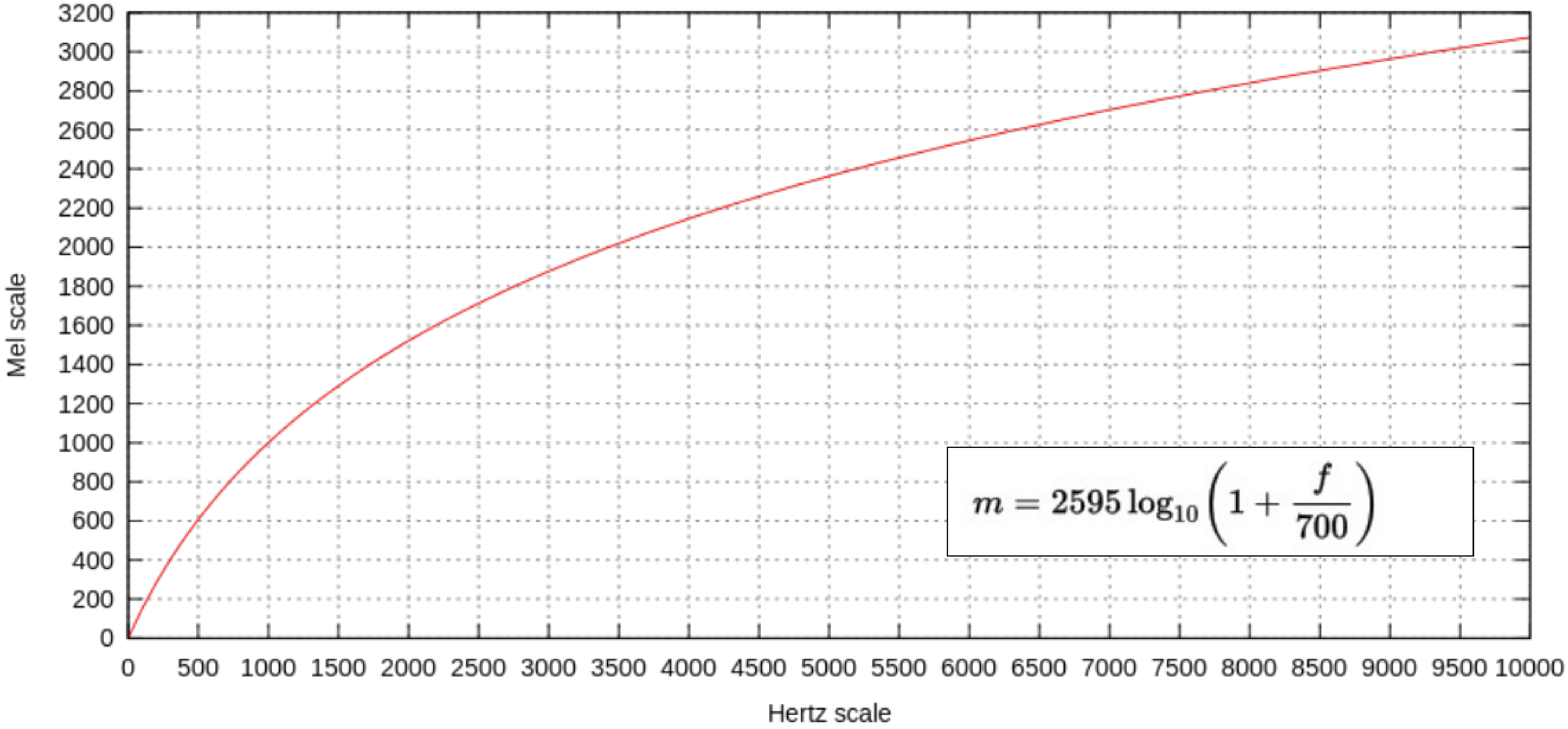
Each of these Fourier transforms constitutes a frame. These successive frames are then concatenated into a matrix to form the spectrogram.

Finally, the frequency axis is changed from a linear scale to a mel scale to reduce the dimensionality, and the magnitudes are scaled logarithmically.



spectrograms

Data



https://en.wikipedia.org/wiki/Mel_scale

Data

MFCC s are commonly derived as follows:

- Take the Fourier transform of (a windowed excerpt of) a signal.
- Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windows.
- Take the logs of the powers at each of the mel frequencies.
- Take the discrete cosine transform of the list of mel log powers, as if it were a signal.
- The MFCCs are the amplitudes of the resulting spectrum.

https://en.wikipedia.org/wiki/Mel-frequency_cepstrum

LibROSA
https://librosa.github.io/librosa/

python_speech_features
https://github.com/jameslyons/python_speech_features

More:

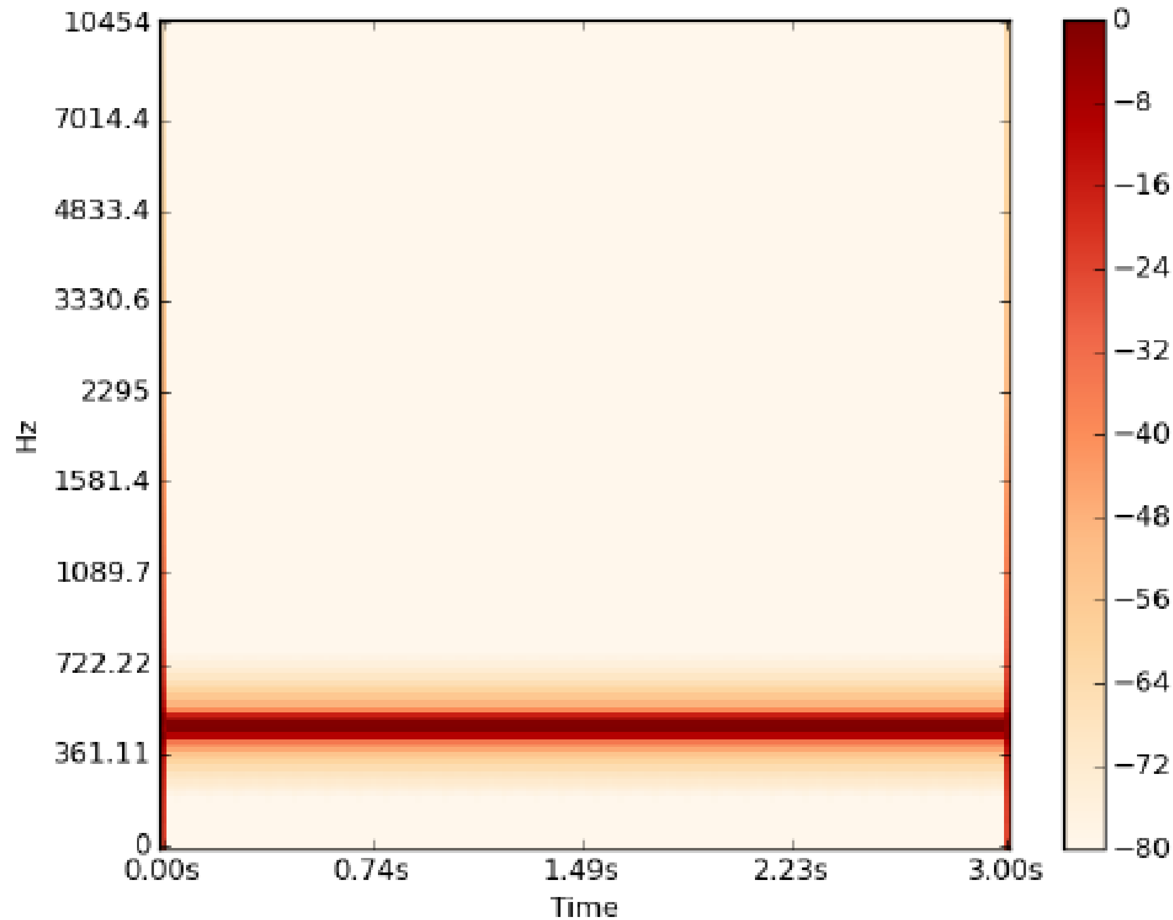
https://github.com/tyiannak/pyAudioAnalysis

https://github.com/naxingyu/opensmile

kapre
https://github.com/keunwoochoi/kapre

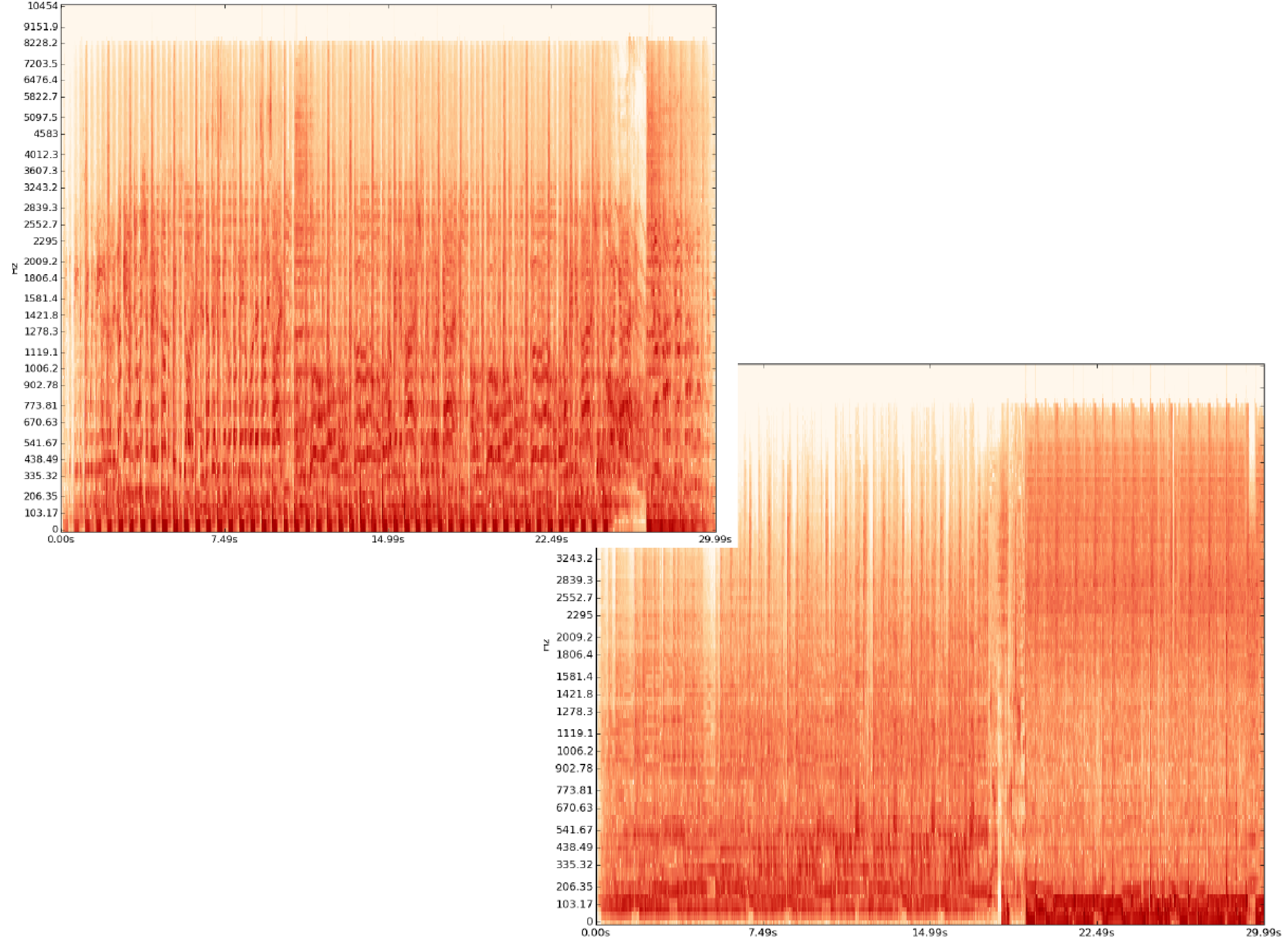
```
model = Sequential()
# A mel-spectrogram layer
model.add(Melspectrogram(n_dft=512, n_hop=256, input_shape=input_shape,
                         border_mode='same', sr=sr, n_mels=128,
                         fmin=0.0, fmax=sr/2, power=1.0,
                         return_decibel_melgram=False, trainable_fb=False,
                         trainable_kernel=False,
                         name='trainable_stft'))
```

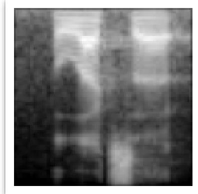
Mel-spectrograms

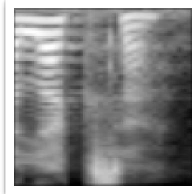


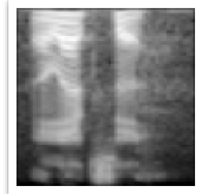
series = np.sin(time)

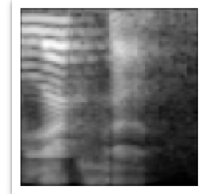
filename = "The Prodigy - Invaders Must Die.mp3"
filename = "Lady GaGa - Poker Face.mp3"

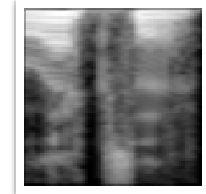

10454
9151.9
8228.2
7203.5
6476.4
5822.7
5097.5
4583
4012.3
3607.3
3243.2
2839.3
2552.7
2295
2009.2
1806.4
1581.4
1421.8
1278.3
1119.1
1006.2
902.78
773.81
670.63
541.67
438.49
335.32
206.35
103.17
0
Hz
0.00s
7.49s
14.99s
22.49s
29.99s
3243.2
2839.3
2552.7
2295
2009.2
1806.4
1581.4
1421.8
1278.3
1119.1
1006.2
902.78
773.81
670.63
541.67
438.49
335.32
206.35
103.17
0
Hz
0.00s
7.49s
14.99s
22.49s
29.99s

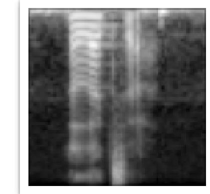

img_18.png

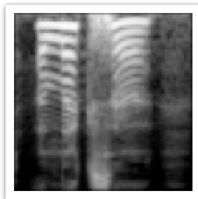

img_19.png

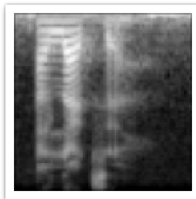

img_20.png

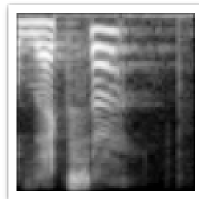

img_21.png

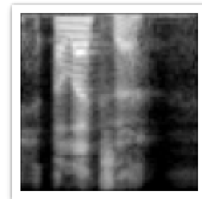

img_22.png

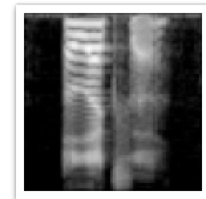

img_23.png

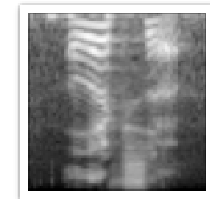

img_24.png

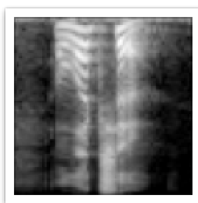

img_25.png

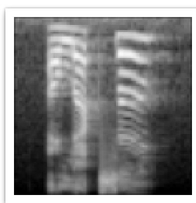

img_26.png

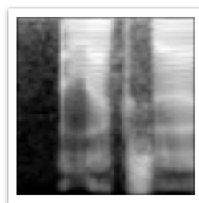

img_27.png

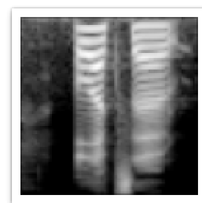

img_28.png

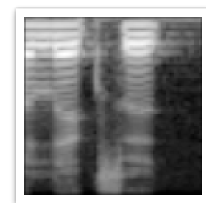

img_29.png

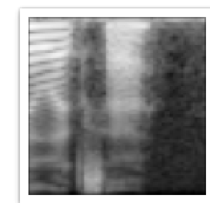

img_30.png

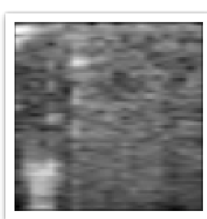

img_31.png

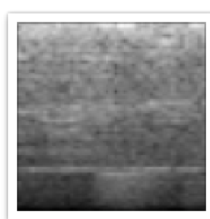

img_32.png

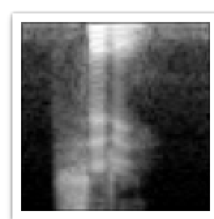

img_33.png

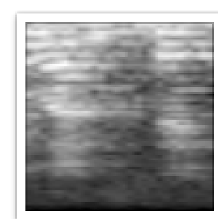

img_34.png


img_35.png

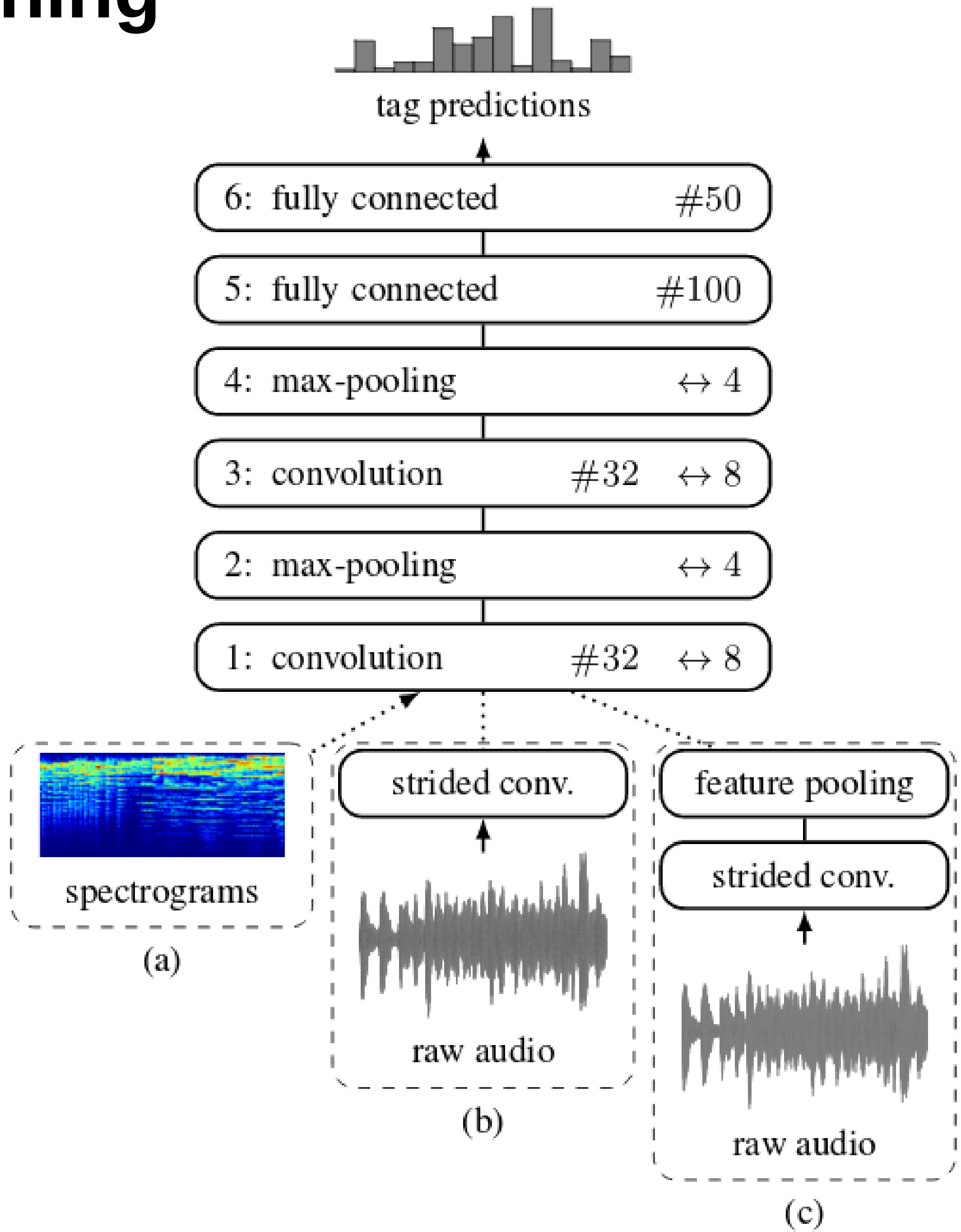

img_data_3_9.png


img_46.png


img_data_2_91.png


img_40.png

Deep Learning



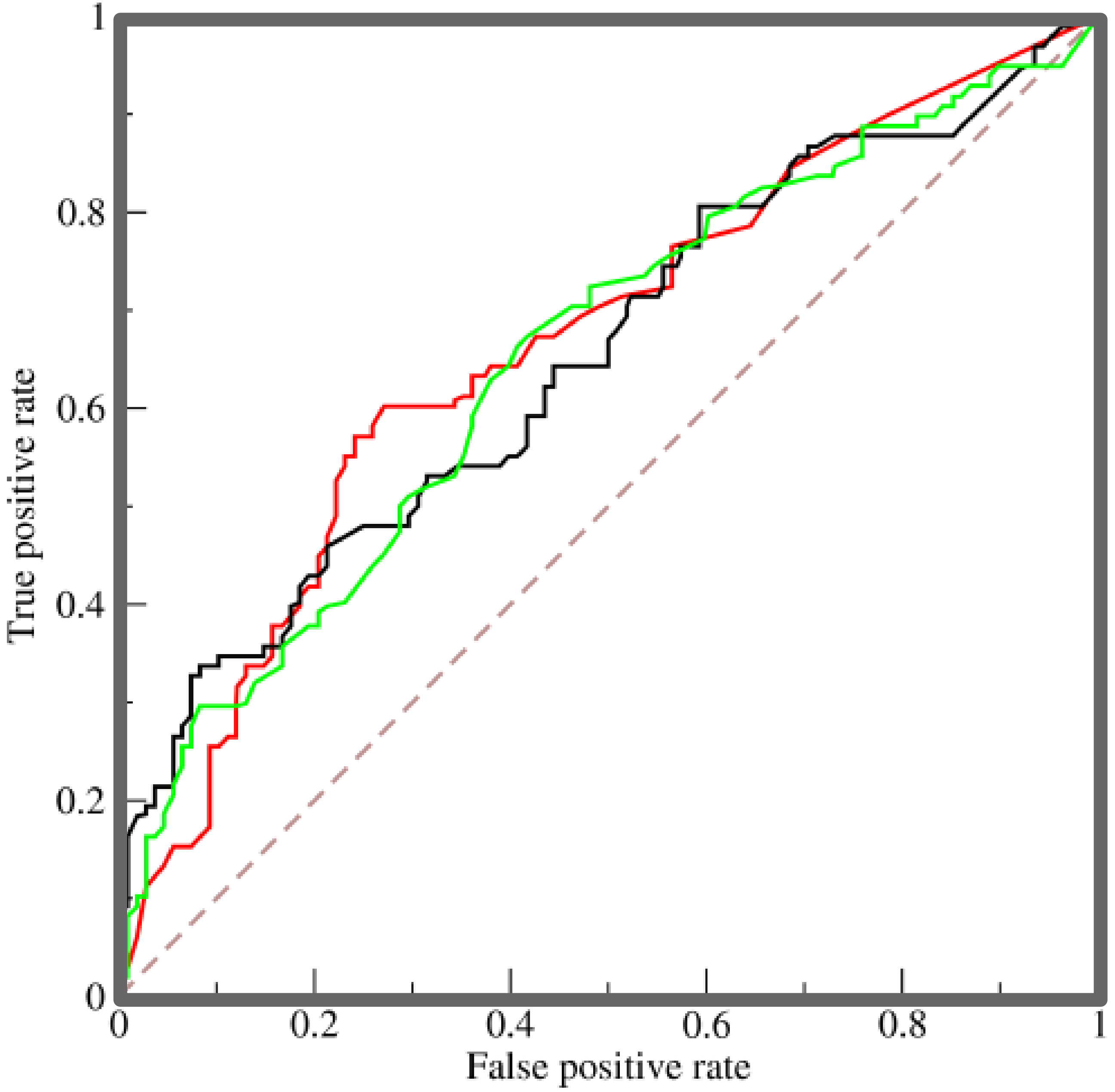
https://redes.unb.br/lasp/files/events/ICASSP2014/papers/p7014-dieleman.pdf

Deep Learning

To evaluate the predictions, was computed the area under the ROC curve (AUC) for each tag and computed the average across all 50 tags.

length	**stride**	**AUC** (spectrograms)	**AUC** (raw audio)
1024	1024	0.8690	0.8366
1024	512	0.8726	0.8365
512	512	0.8793	0.8386
512	256	0.8793	0.8408
256	256	0.8815	0.8487

ROC



https://en.wikipedia.org/wiki/Receiver_operating_characteristic

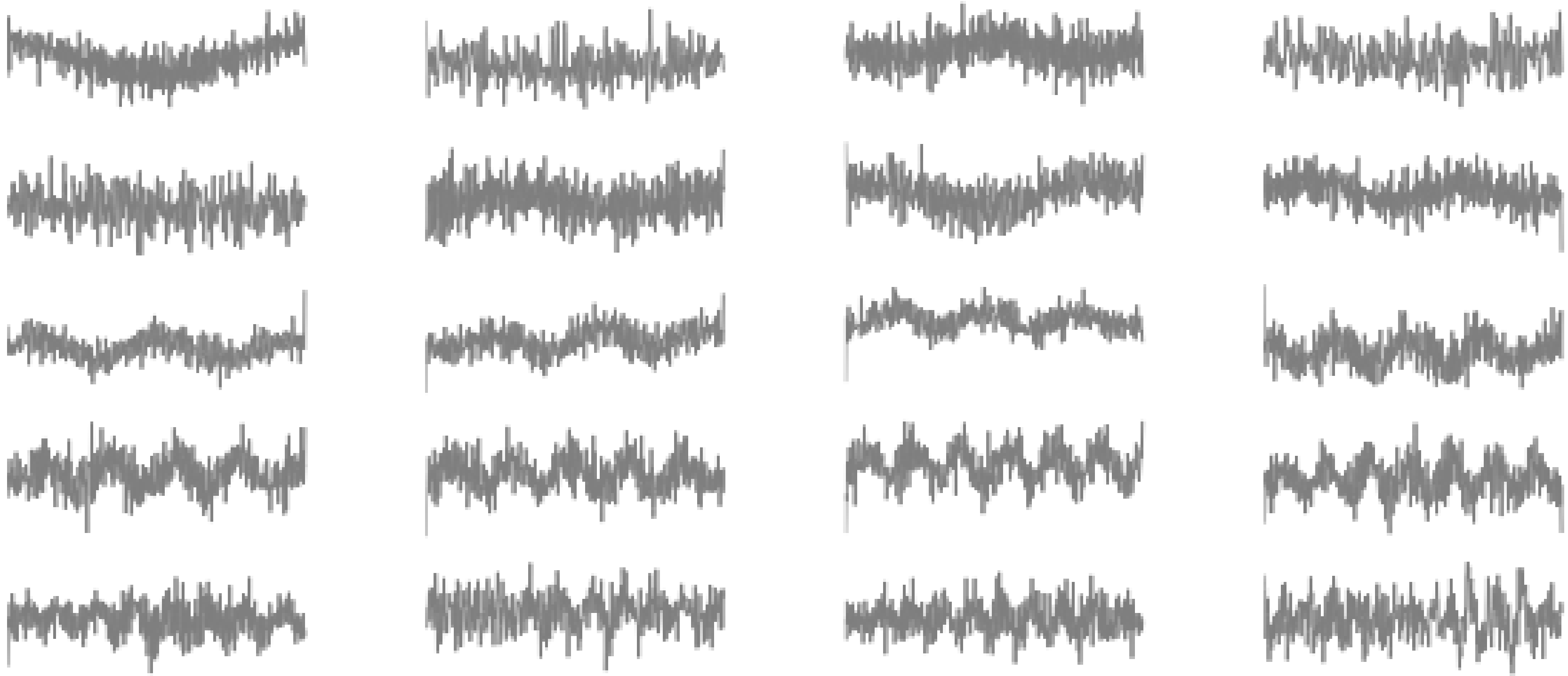
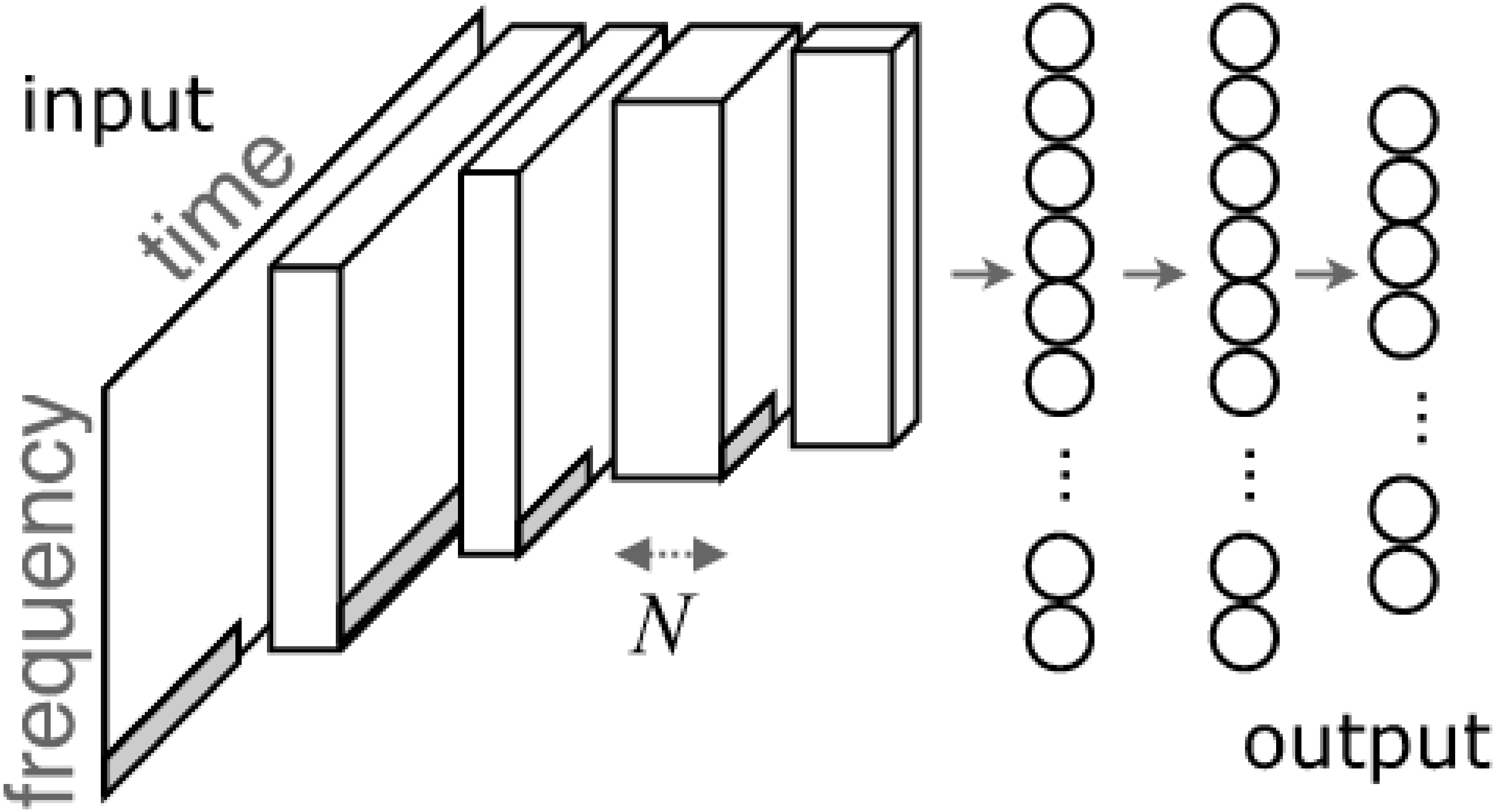


Fig. 3. A subset of filters learned in the lowest layer of a convolutional neural network that processes raw audio signals,

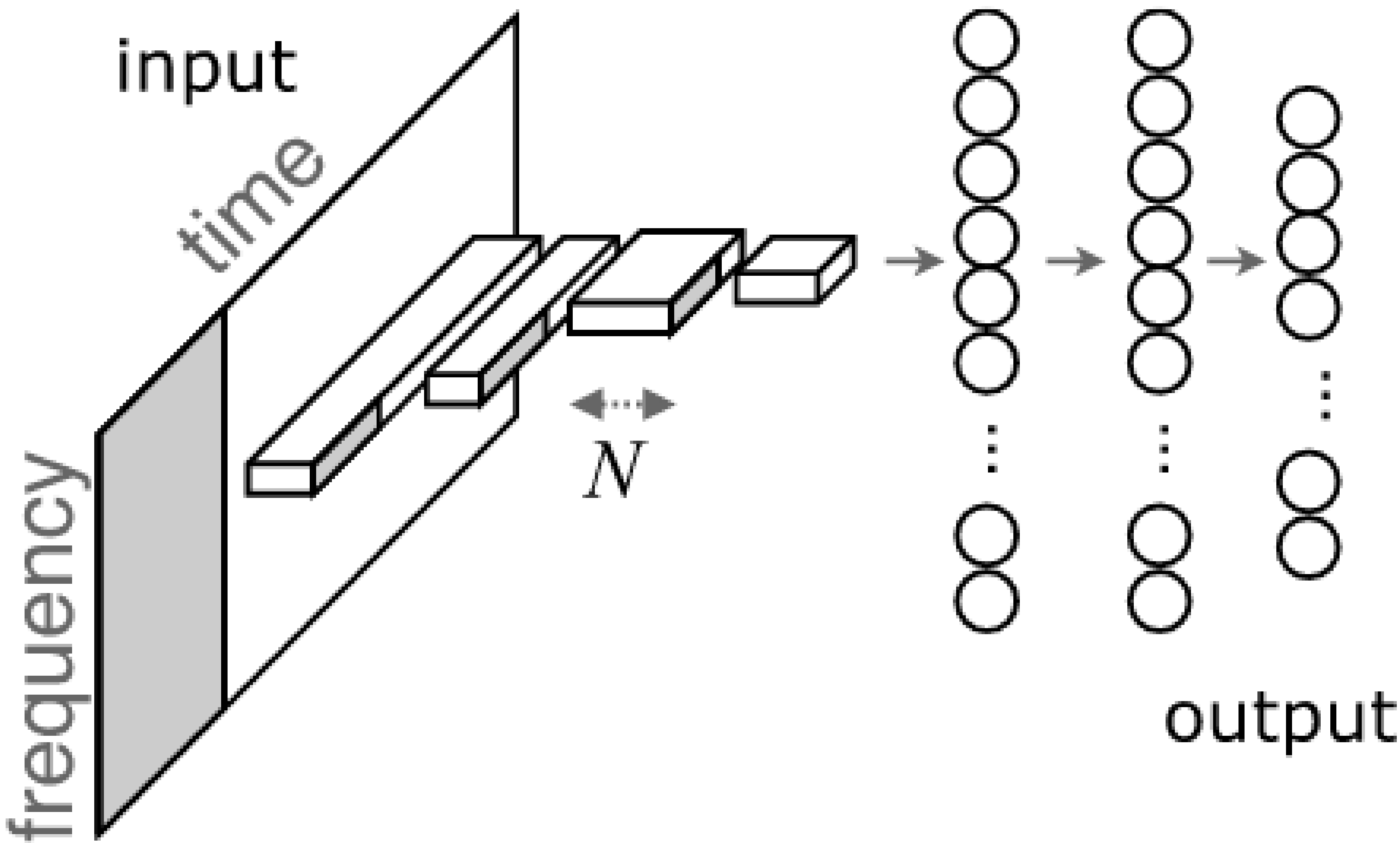
Shown that the networks are able to learn useful features from raw audio: they are able to autonomously discover frequency decompositions.

Deep Learning



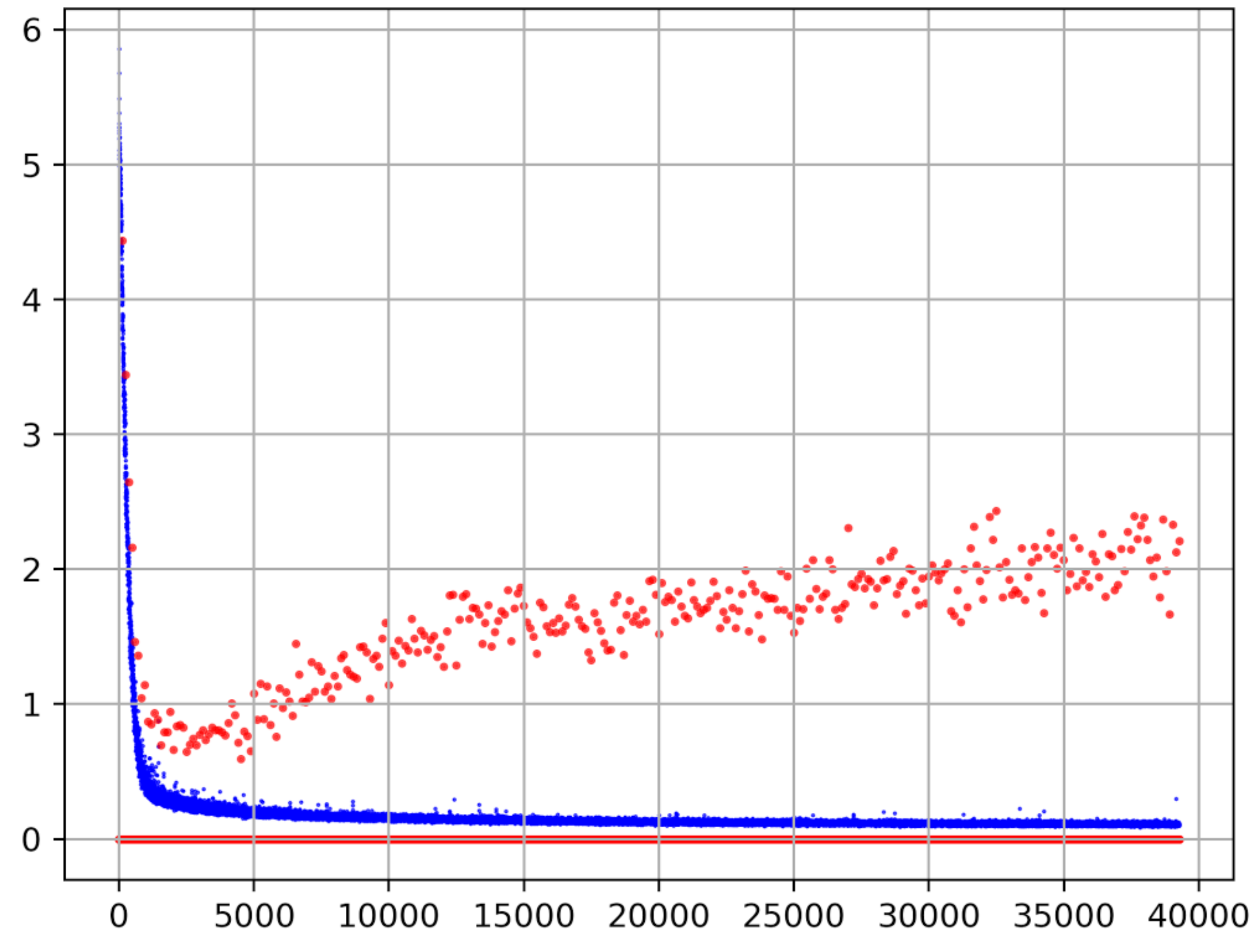
https://arxiv.org/pdf/1609.04243.pdf

Deep Learning

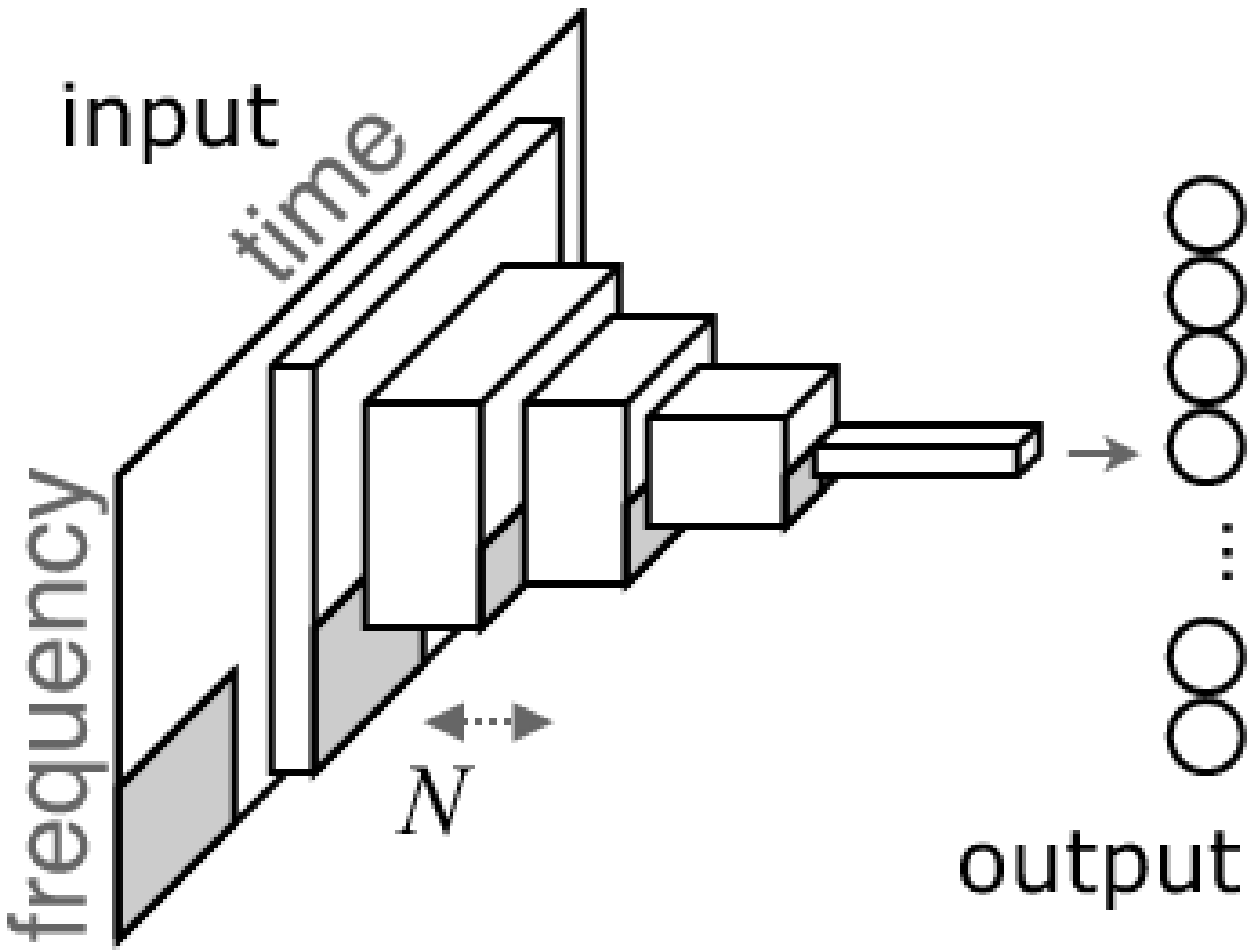


- **Overfitting**

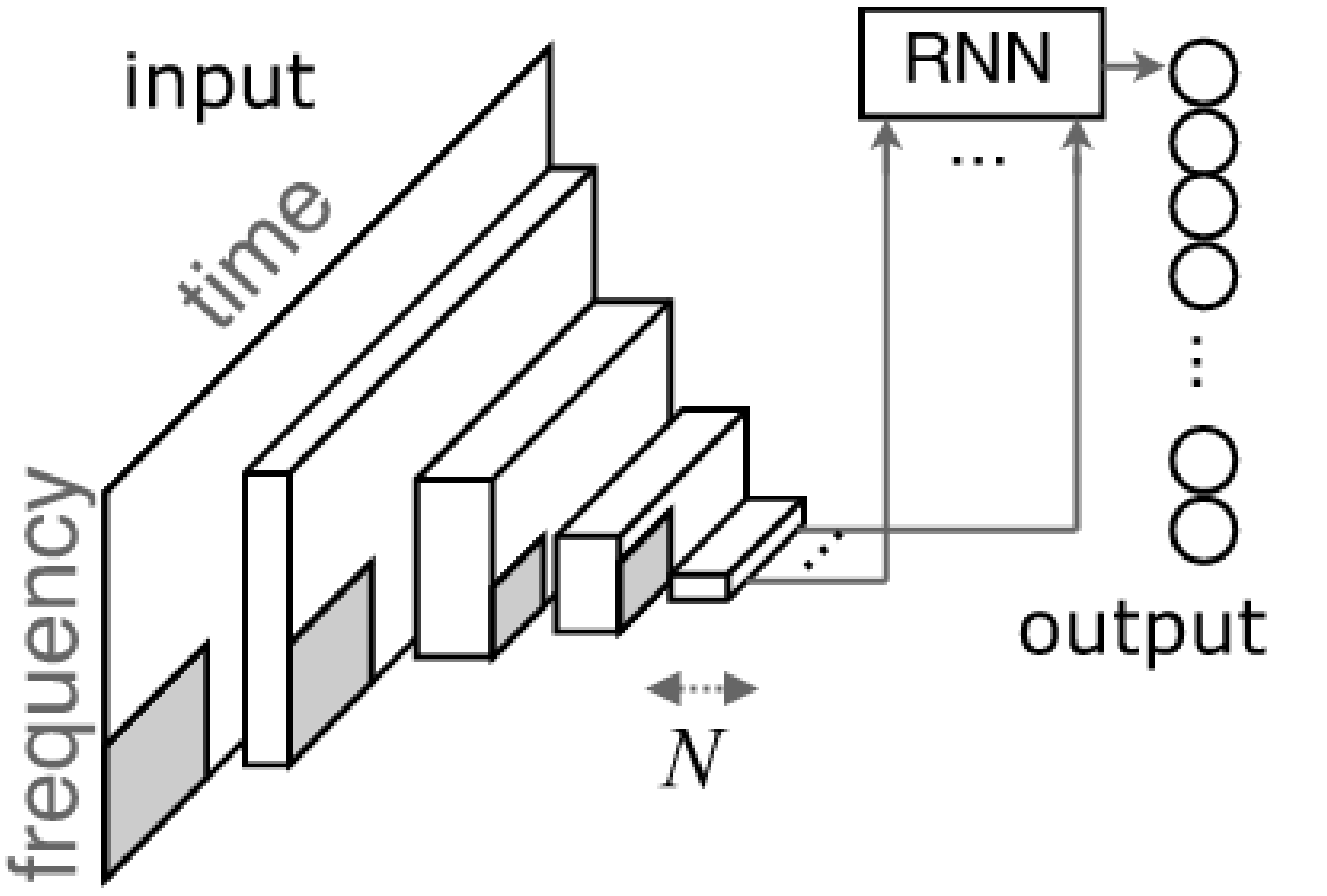
- **Not optimized (slow)**



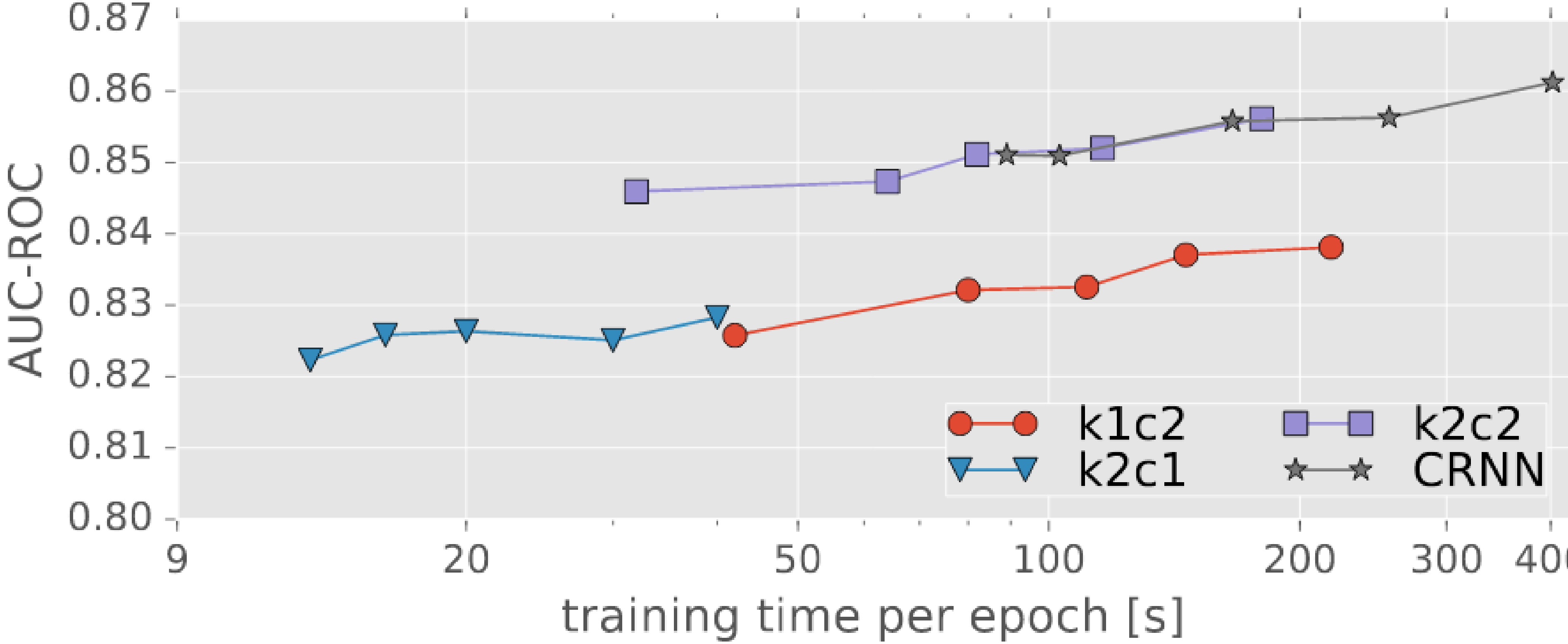
Deep Learning



Deep Learning



Deep Learning



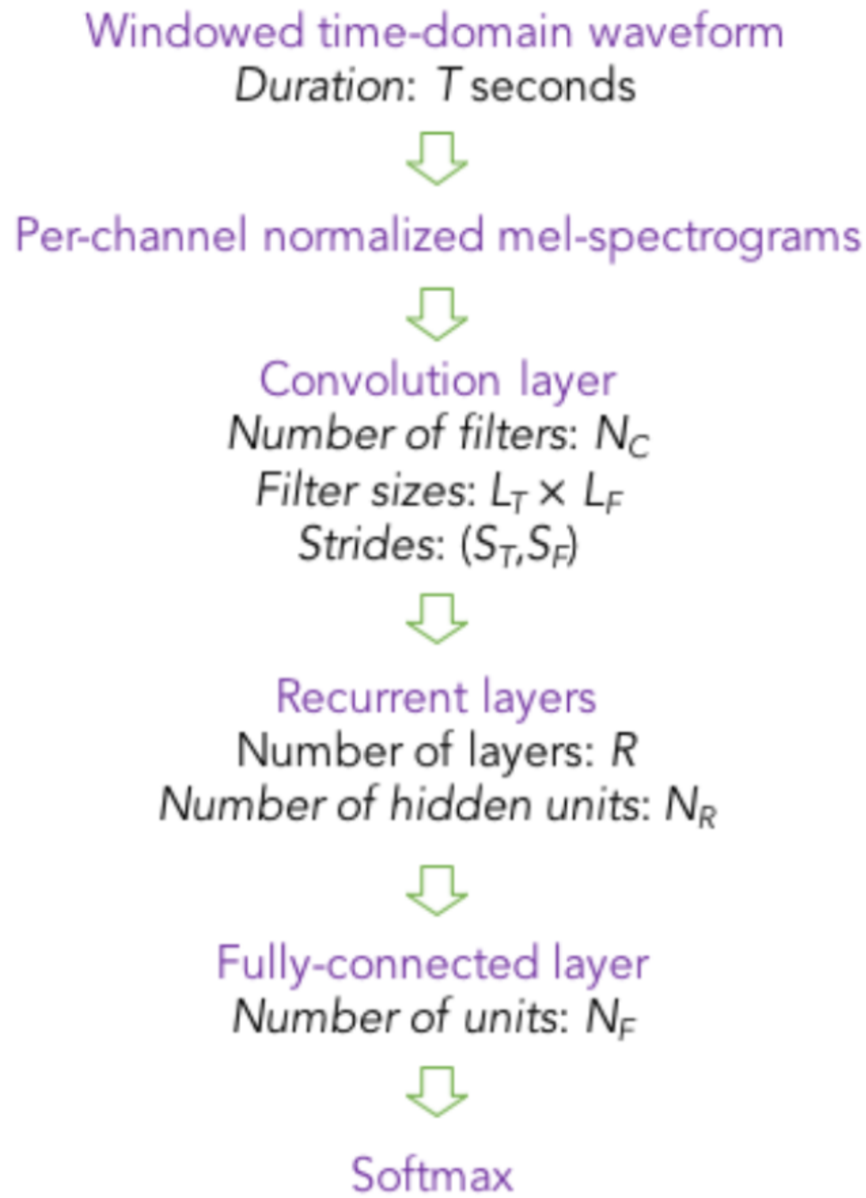


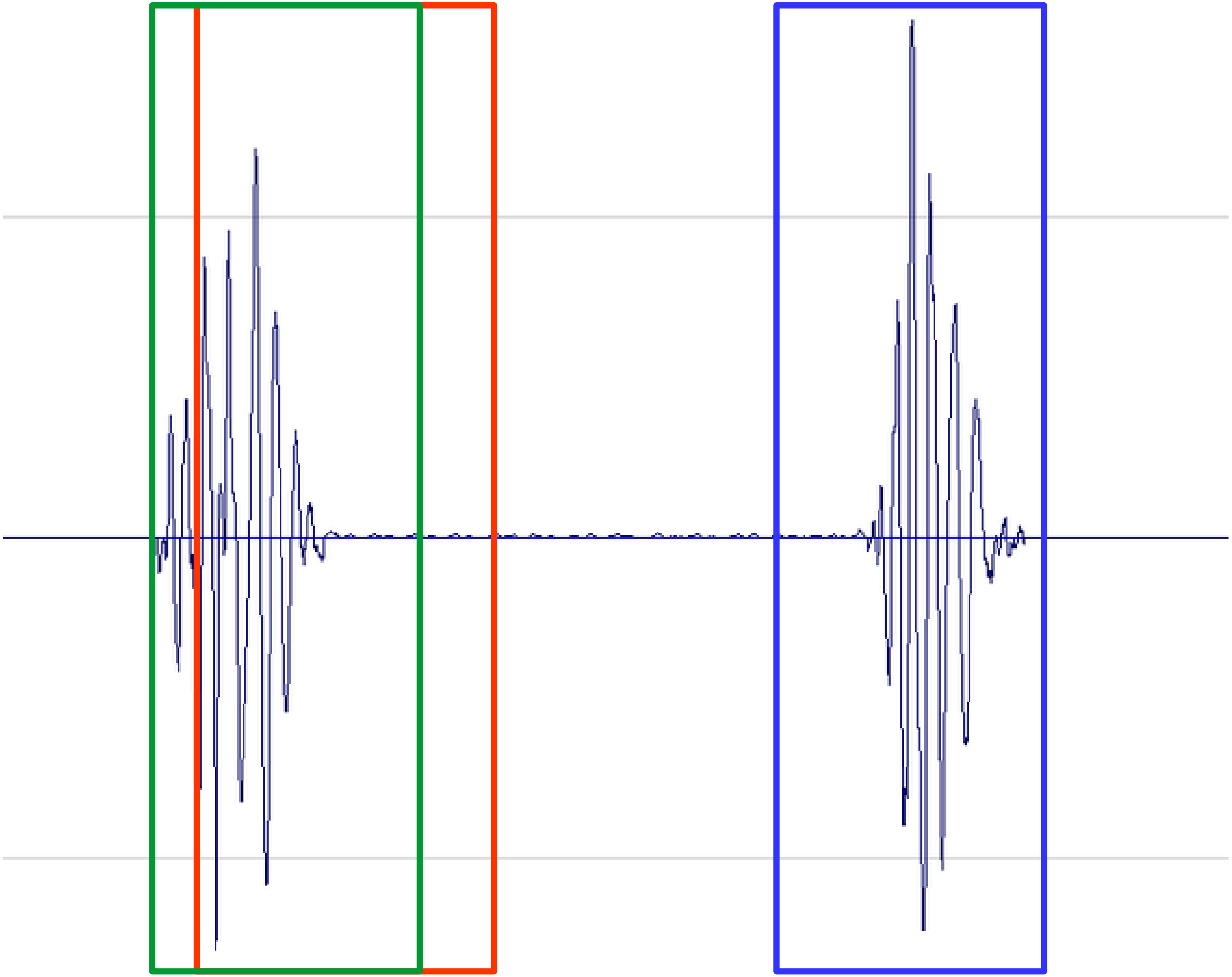
Figure 1: *End-to-end CRNN architecture for KWS.*

https://arxiv.org/ftp/arxiv/papers/1703/1703.05390.pdf

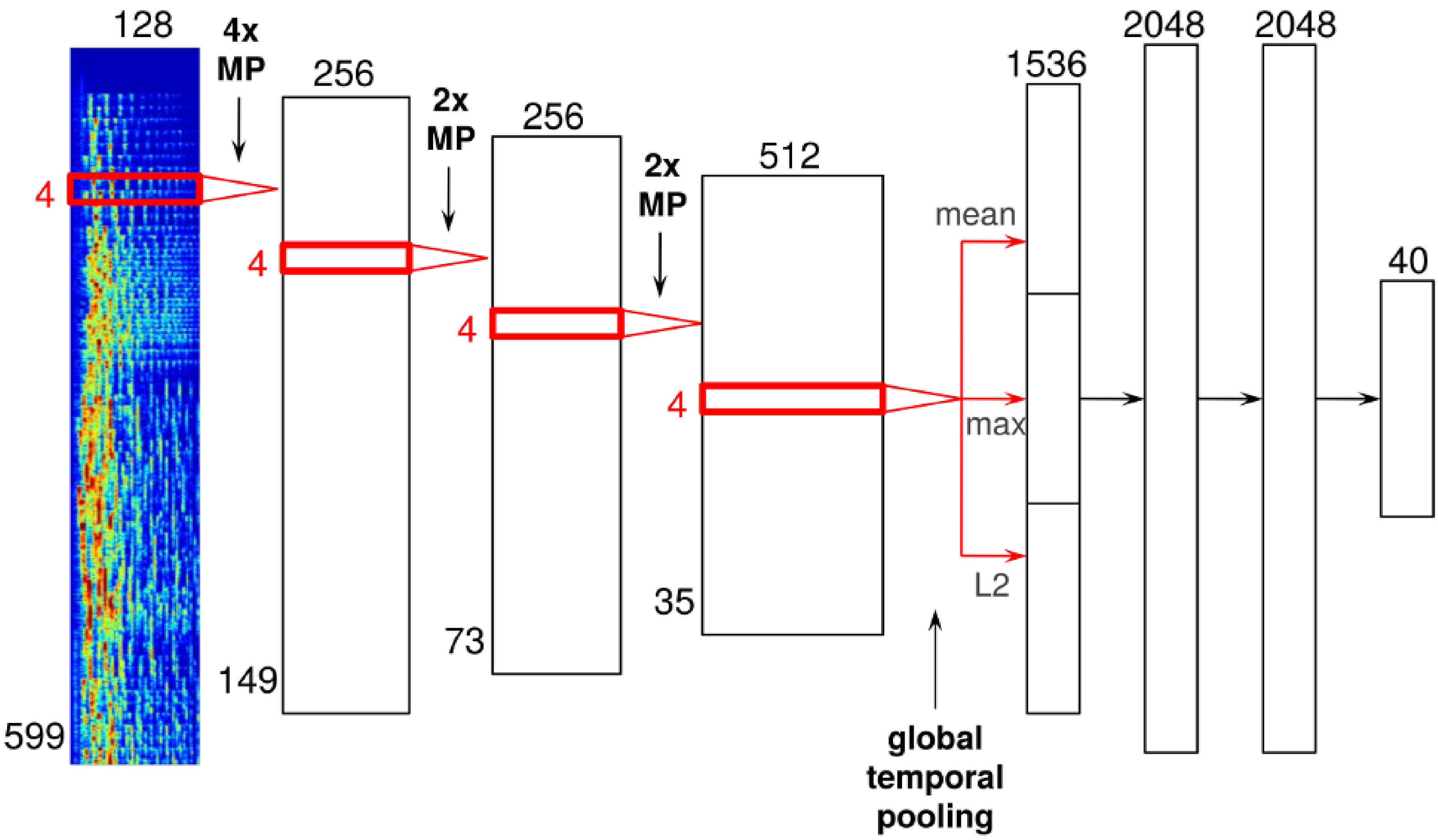
Convolutional			Recurrent			FC	Total number of parameters	FRR (%) for the noise development set with 5 dB SNR	
N_C	(L_T, L_F)	(S_T, S_F)	R	N_R	Recurrent unit	N_F		at 1 FA/hour	at 0.5 FA/hour
32	(20,5)	(8,2)	2	8	GRU	32	45k	5.54	7.44
32	(20,5)	(8,2)	3	8	LSTM	64	68k	6.17	7.68
32	(5,1)	(4,1)	2	8	GRU	64	102k	6.04	7.31
32	(20,5)	(8,2)	2	16	GRU	64	110k	3.48	4.46
32	(20,5)	(20,5)	2	32	GRU	64	110k	5.70	7.99
32	(20,5)	(8,2)	3	16	GRU	64	115k	3.42	4.10
16	(20,5)	(8,2)	2	32	GRU	32	127k	3.53	5.55
32	(20,5)	(12,4)	2	32	GRU	64	143k	5.80	7.72
16	(20,5)	(8,2)	1	32	GRU	64	148k	4.20	6.27
128	(20,5)	(8,2)	3	8	GRU	32	159k	3.83	5.21
64	(10,3)	(8,2)	1	16	GRU	32	166k	3.21	4.31
128	(20,5)	(8,2)	1	32	LSTM	64	197k	3.37	4.56
32	(20,5)	(12,2)	2	32	GRU	64	205k	3.26	4.40
32	(20,5)	(8,2)	1	32	GRU	64	211k	3.00	3.84
32	**(20,5)**	**(8,2)**	**2**	**32**	**GRU**	**64**	**229k**	**2.85**	**3.79**
32	(40,10)	(8,2)	2	32	GRU	64	239k	3.57	5.03
32	(20,5)	(8,2)	3	32	GRU	64	248k	3.00	3.42
32	(20,5)	(8,2)	2	32	LSTM	64	279k	3.06	4.41
32	(20,5)	(8,1)	2	32	GRU	64	352k	2.23	3.31
64	(20,5)	(8,2)	2	32	GRU	64	355k	2.43	3.99
64	(20,5)	(8,2)	2	32	LSTM	32	407k	3.11	4.04
64	(10,3)	(4,1)	2	32	GRU	64	674k	3.37	4.35
128	(20,5)	(8,2)	2	32	GRU	128	686k	2.64	3.78
32	(20,5)	(8,2)	2	128	GRU	128	1513k	2.23	2.95
256	(20,5)	(8,2)	4	64	GRU	128	2551k	2.18	3.42
128	(20,5)	(4,1)	4	64	GRU	128	2850k	2.64	3.21

For all networks,the input is assumed to be of size 96×1366 (mel-frequency band × time frame) and single channel.

Deep Learning



Deep Learning

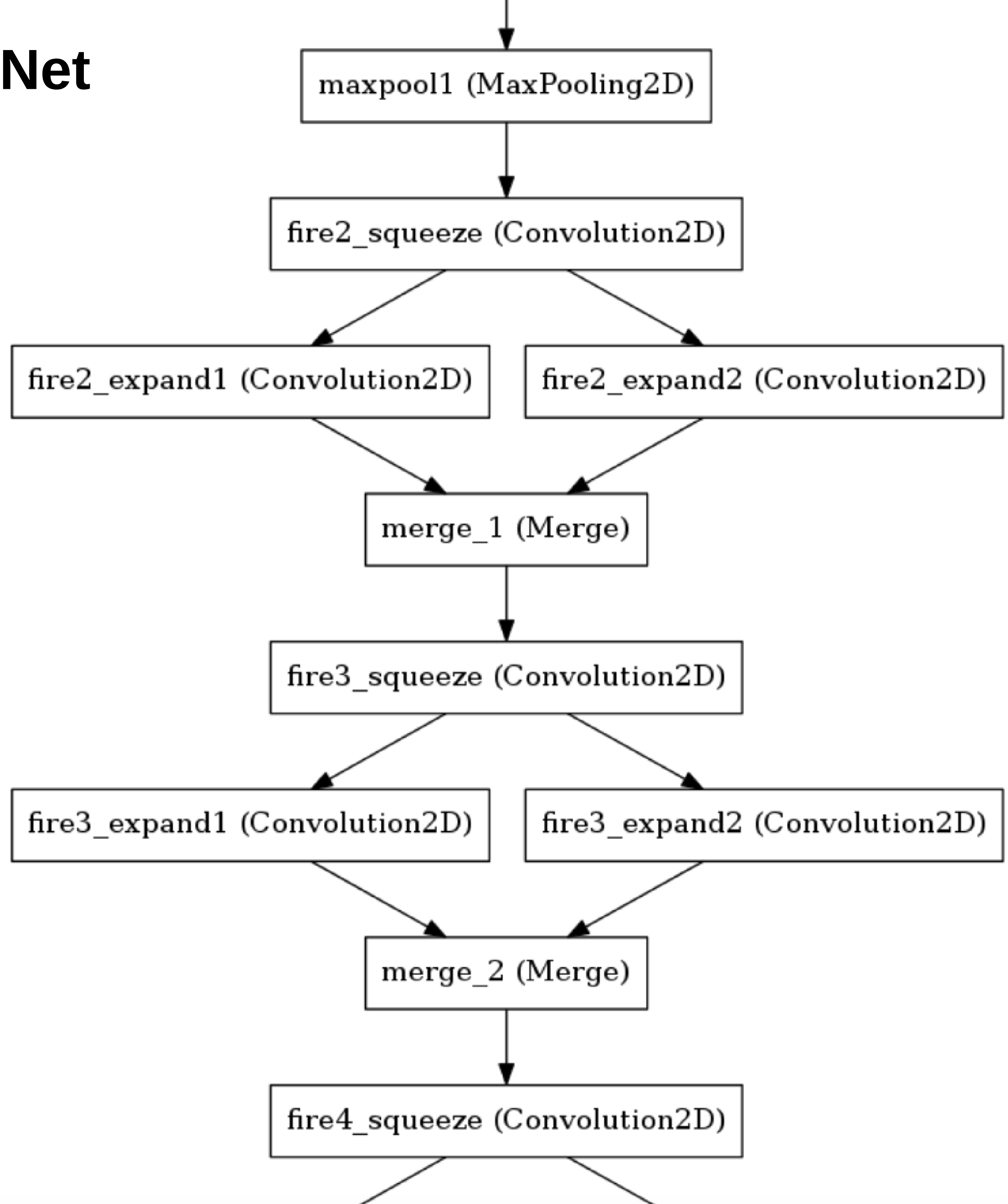


http://benanne.github.io/2014/08/05/spotify-cnns.html

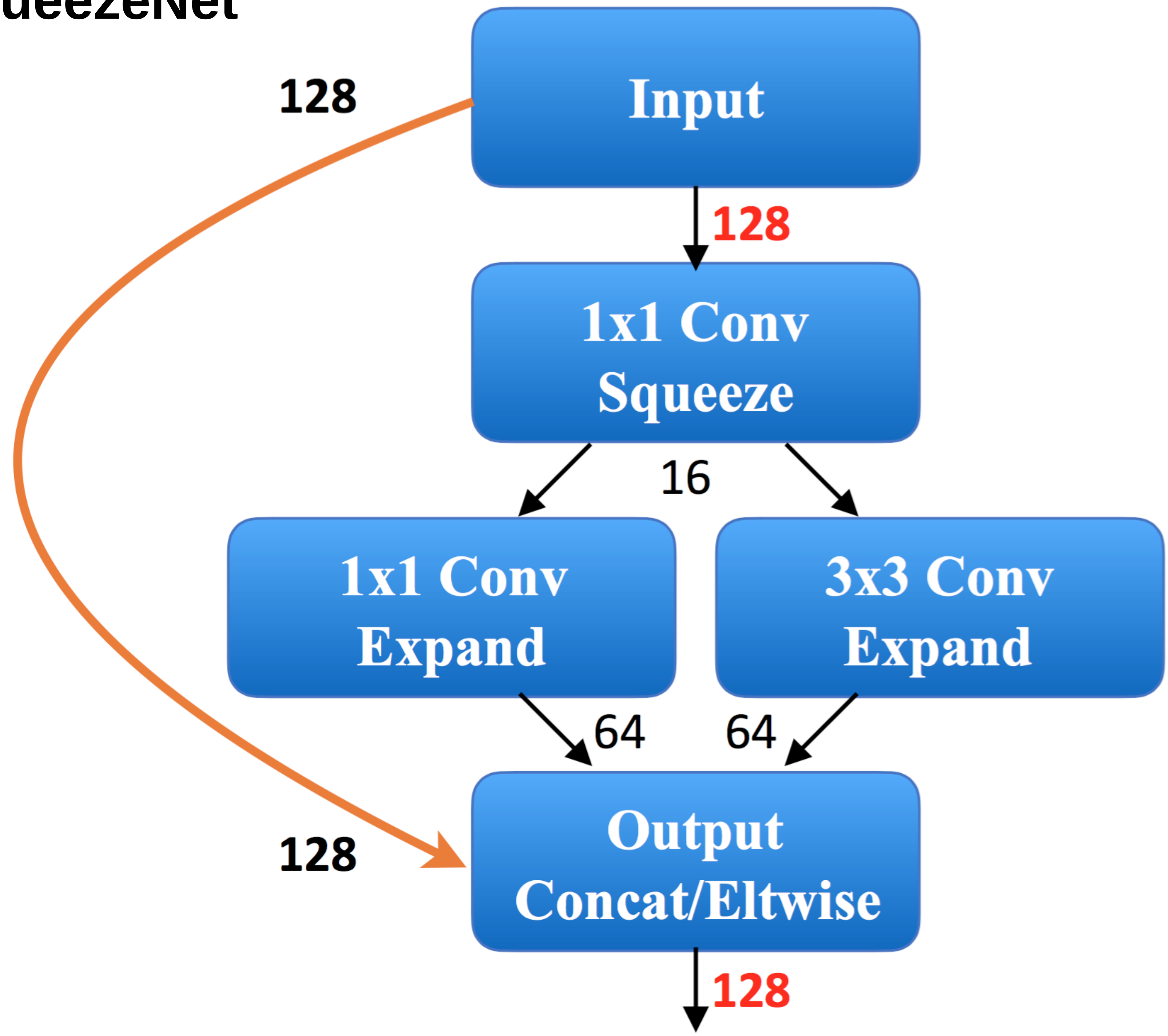
VGG

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

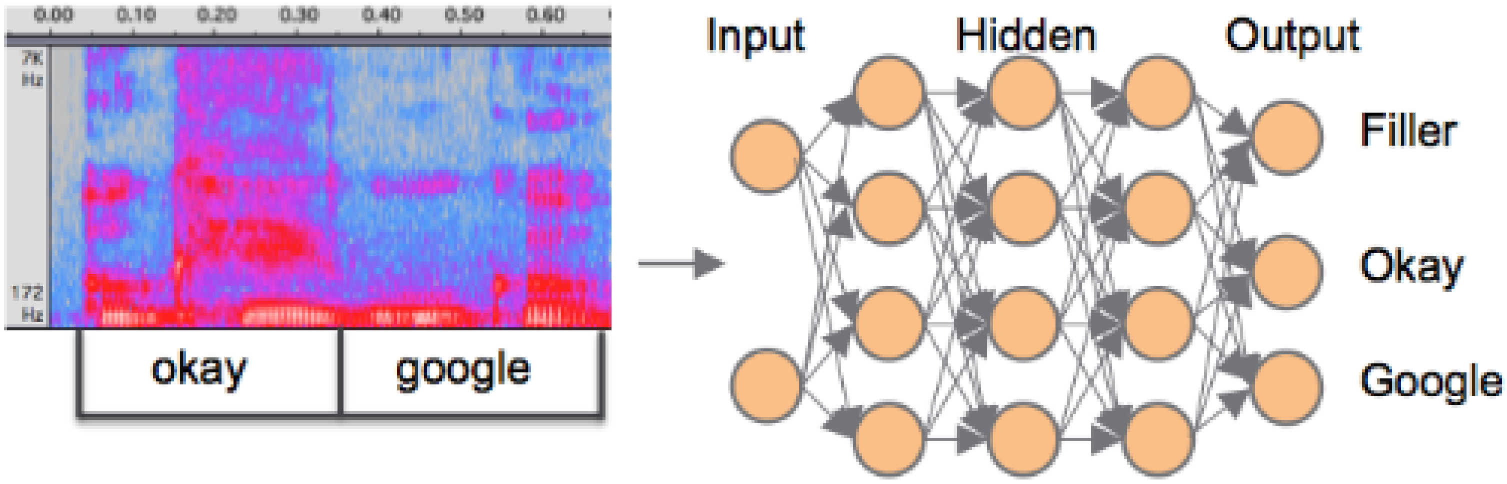
SqueezeNet


maxpool1 (MaxPooling2D)
fire2_squeeze (Convolution2D)
fire2_expand1 (Convolution2D)
fire2_expand2 (Convolution2D)
merge_1 (Merge)
fire3_squeeze (Convolution2D)
fire3_expand1 (Convolution2D)
fire3_expand2 (Convolution2D)
merge_2 (Merge)
fire4_squeeze (Convolution2D)

SqueezeNet



128
Input
128
1x1 Conv
Squeeze
16
1x1 Conv
Expand
3x3 Conv
Expand
64
64
128
Output
Concat/Eltwise
128

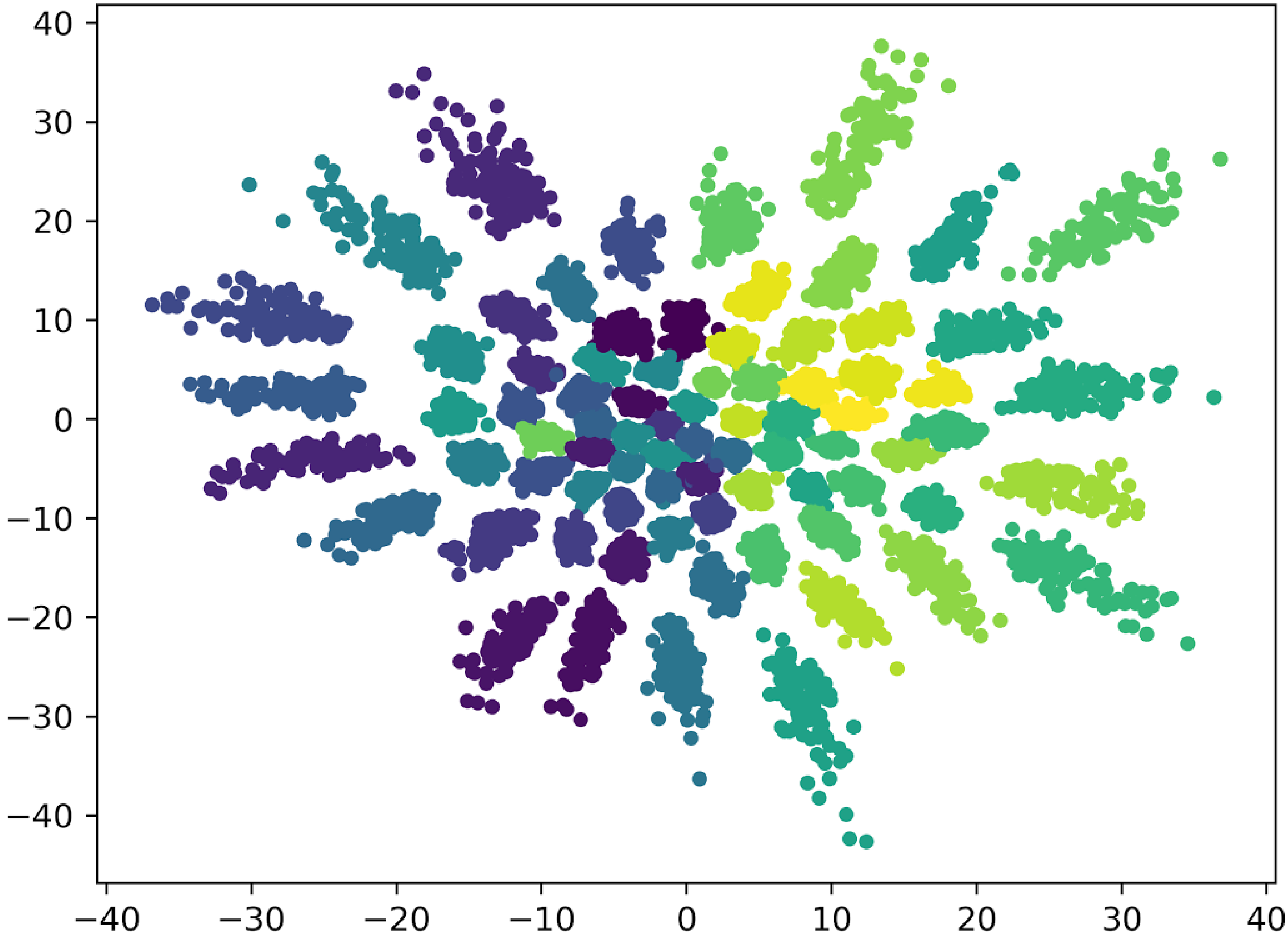


(i) Feature Extraction (ii) Deep Neural Network

Keyword recognition results achieve 45% relative improvement with respect to a competitive Hidden Markov Model-based system.

https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/42537.pdf

User identification



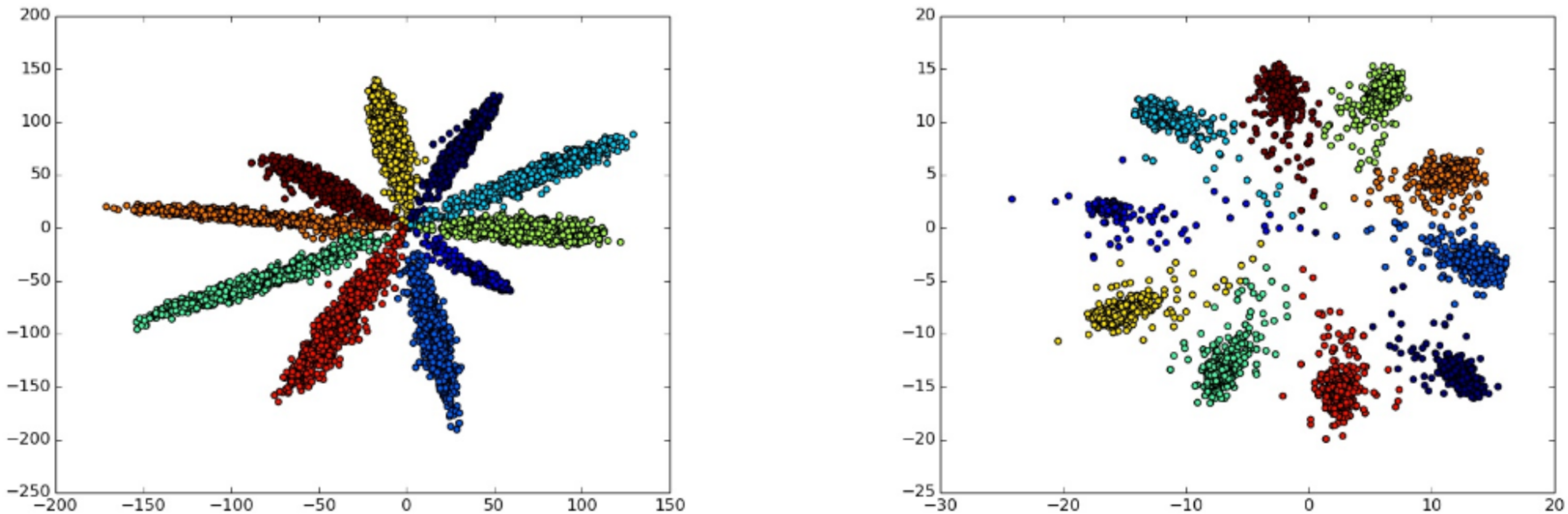
User identification

FaceNet

https://arxiv.org/pdf/1503.03832.pdf

Center loss

https://arxiv.org/pdf/1707.07391.pdf

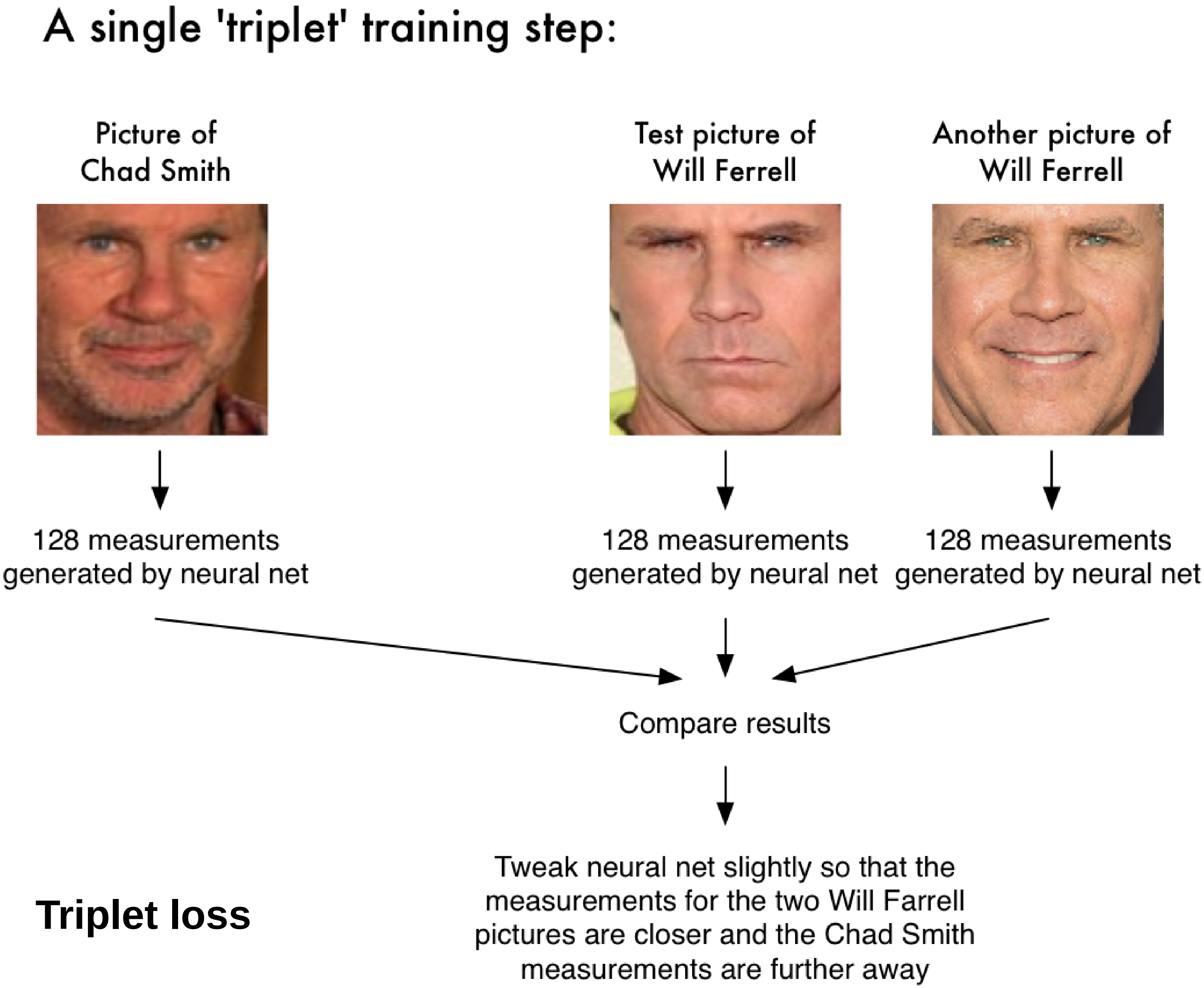


(a) softmax loss (b) center loss

Center loss

$$L_c = \frac{1}{2} \sum_{i=1}^{m} \|x_i - c_{y_i}\|_2^2 \tag{1}$$

Where L_c denotes the center loss. m denotes the number of training samples in a min-batch. $x_i \in R_d$ denotes the ith training sample. y_i denotes the label of x_i. $c_{y_i} \in R_d$ denotes the y_ith class center of deep features. d is the feature dimension.



Triplet loss

https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78

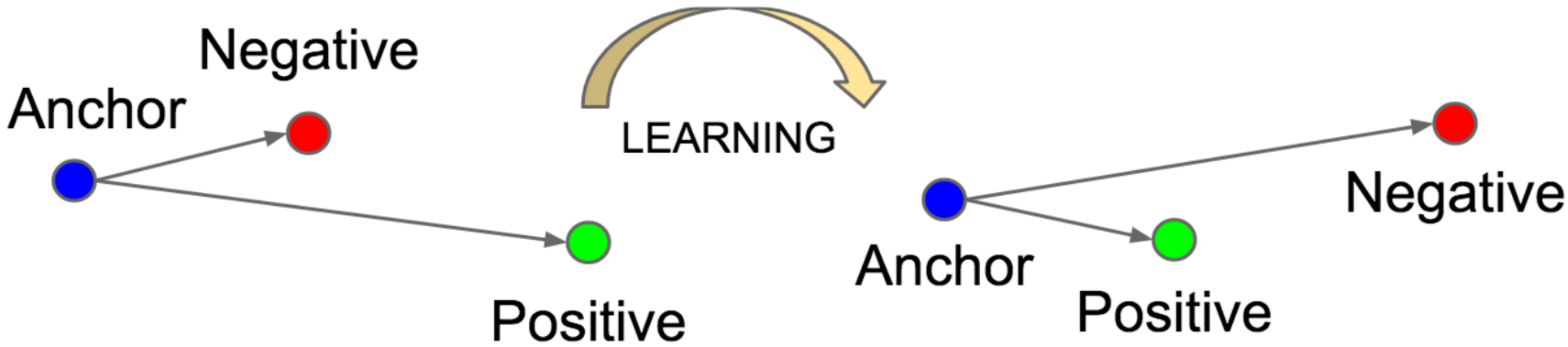
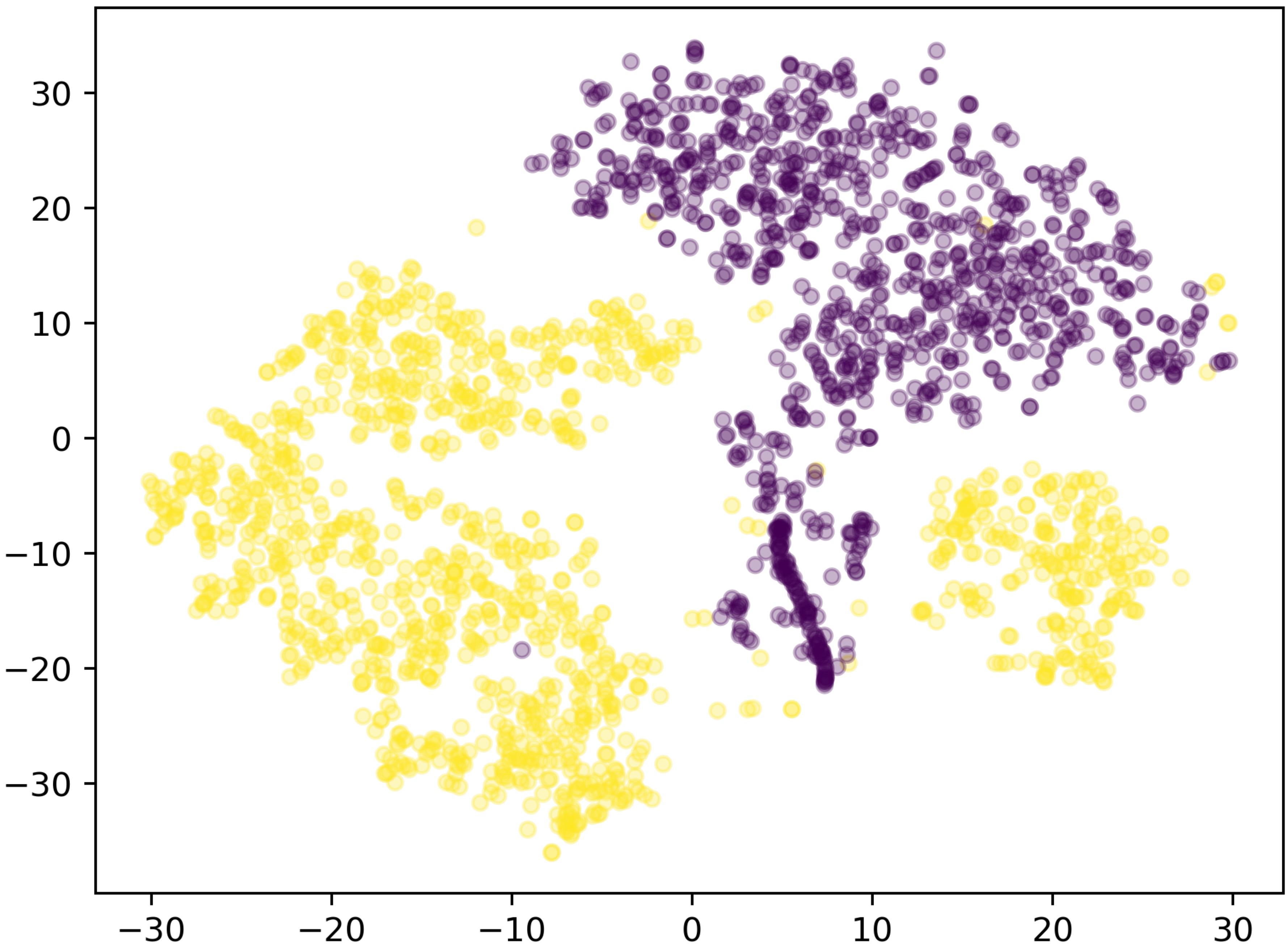


Figure 3. The **Triplet Loss** minimizes the distance between an *anchor* and a *positive*, both of which have the same identity, and maximizes the distance between the *anchor* and a *negative* of a different identity.

The loss that is being minimized is then $L =$

$$\sum_{i}^{N} \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$


30
20
10
0
−10
−20
−30
−30
−20
−10
0
10
20
30

- https://redes.unb.br/lasp/files/events/ICASSP2014/papers/p7014-dieleman.pdf
- http://benanne.github.io/2014/08/05/spotify-cnns.html
- https://arxiv.org/pdf/1609.04243.pdf
- https://arxiv.org/pdf/1606.00298.pdf
- https://github.com/keunwoochoi/music-auto_tagging-keras
- http://aqibsaeed.github.io/2016-09-03-urban-sound-classification-part-1/
- https://github.com/aqibsaeed/Urban-Sound-Classification
- https://arxiv.org/ftp/arxiv/papers/1703/1703.05390.pdf

Facebook
https://www.facebook.com/neverdraw

LinkedIn
https://www.linkedin.com/in/awesomengineer

Github
https://github.com/spaceuniverse