Deep Neural Network Based Polyphonic Sound Event Detection

Emre Cakir, Giambattista Parascandolo

Audio Research Group
Tampere University of Technology, Finland

February 24, 2017

SGN-41007 Guest lecture

TAMPERE UNIVERSITY OF TECHNOLOGY

TAMPERE UNIVERSITY OF TECHNOLOGY

Sound Events

TAMPERE UNIVERSITY OF TECHNOLOGY

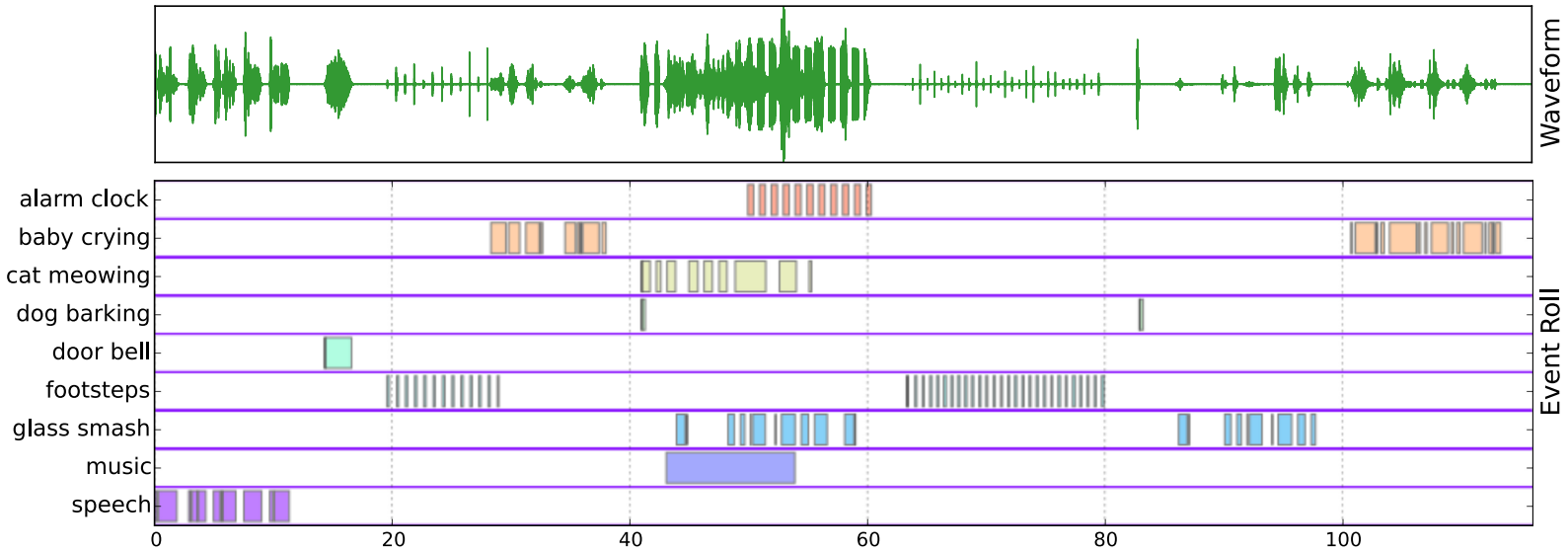
SED Application Areas

- Audio surveillance (gun shots, scream, footsteps)
- Health-care monitoring (coughing, sneezing frequency)
- Context-aware devices (a mobile phone detecting the user is in a meeting and adjusting settings accordingly)
- Smart-home systems
- etc.

TAMPERE UNIVERSITY OF TECHNOLOGY

Polyphonic Sound Event Detection

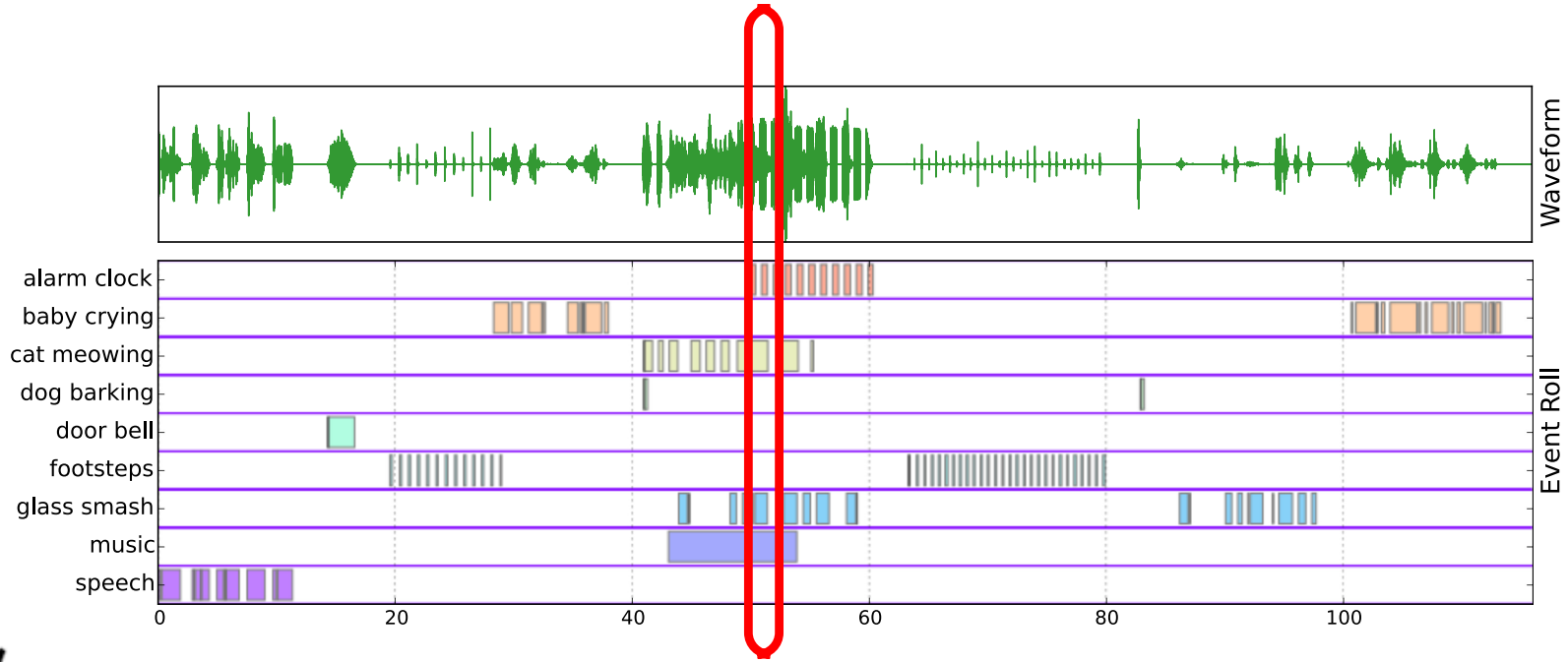
1. Estimate start and end time of each sound event
2. Assign a textual label for each event



TAMPERE UNIVERSITY OF TECHNOLOGY

Polyphonic Sound Event Detection

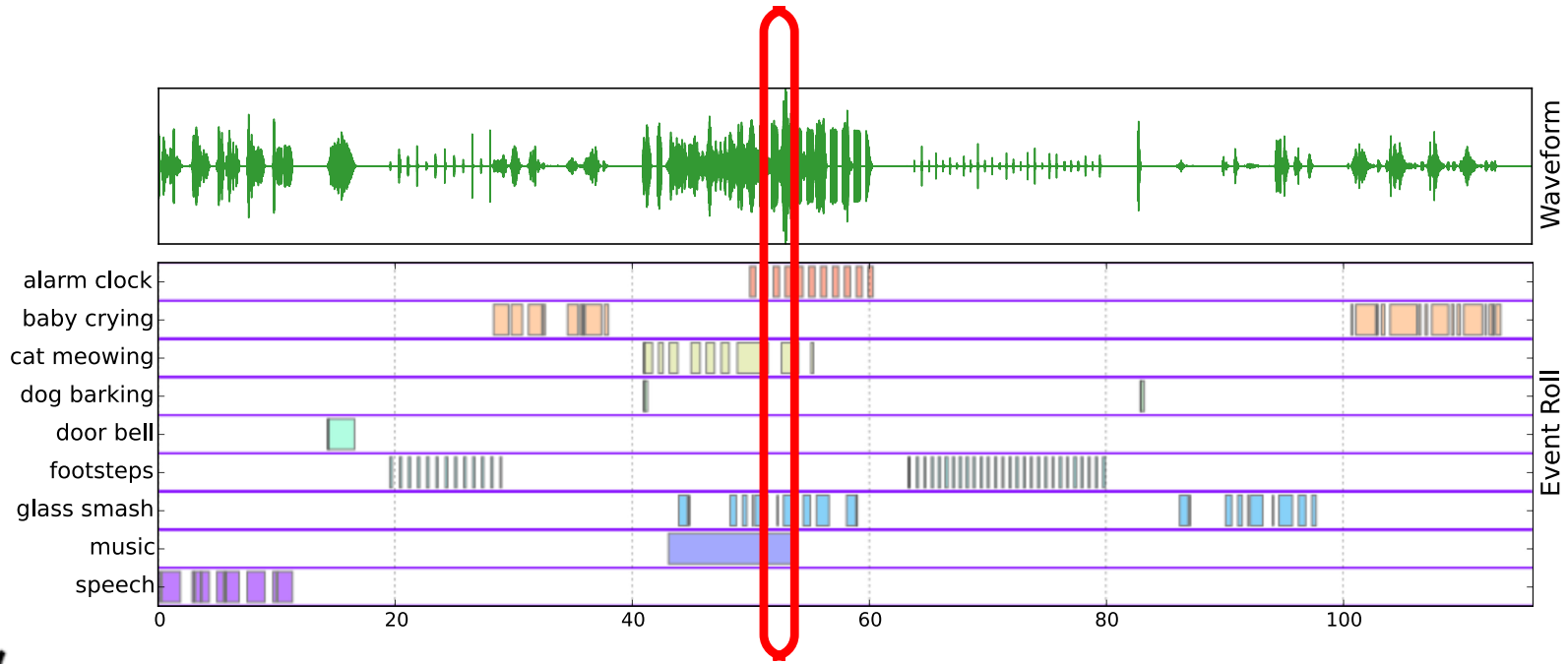
1. Estimate start and end time of each sound event
2. Assign a textual label for each event



TAMPERE UNIVERSITY OF TECHNOLOGY

Polyphonic Sound Event Detection

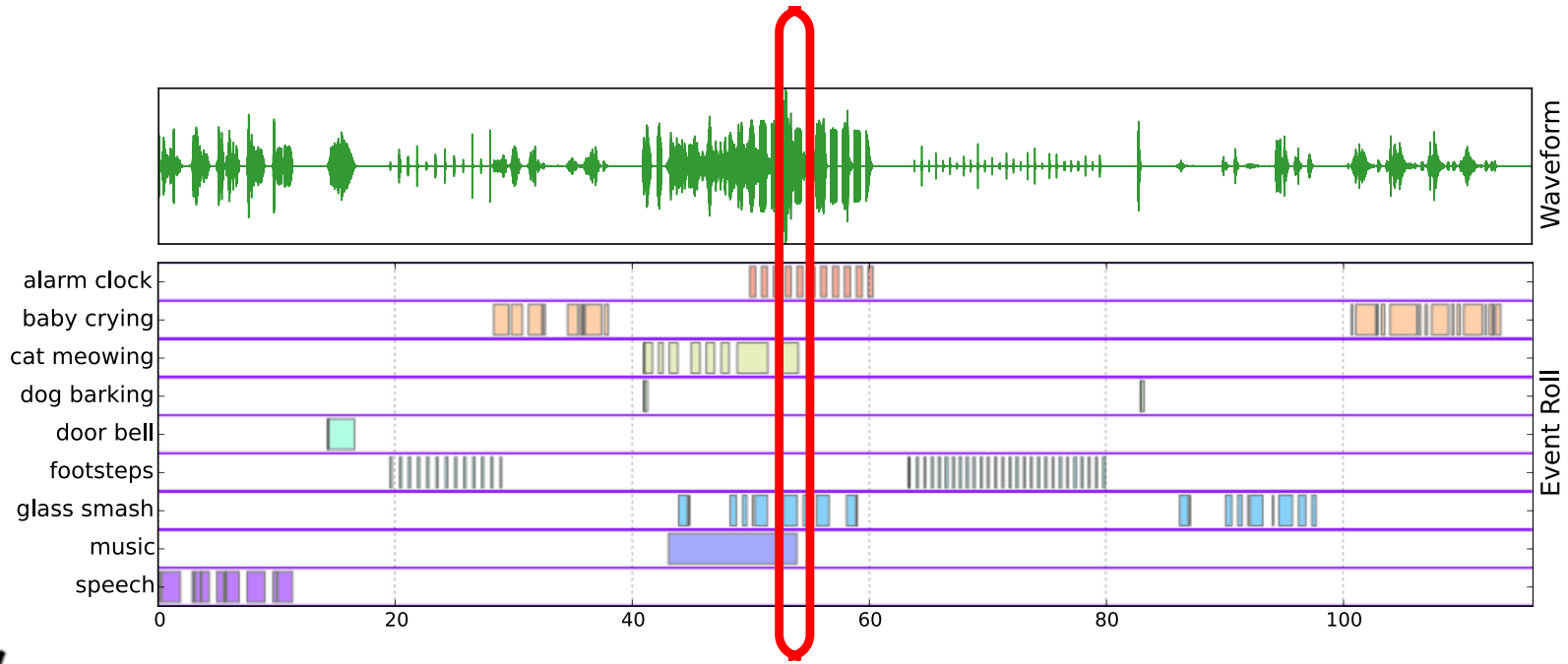
1. Estimate start and end time of each sound event
2. Assign a textual label for each event



TAMPERE UNIVERSITY OF TECHNOLOGY

Polyphonic Sound Event Detection

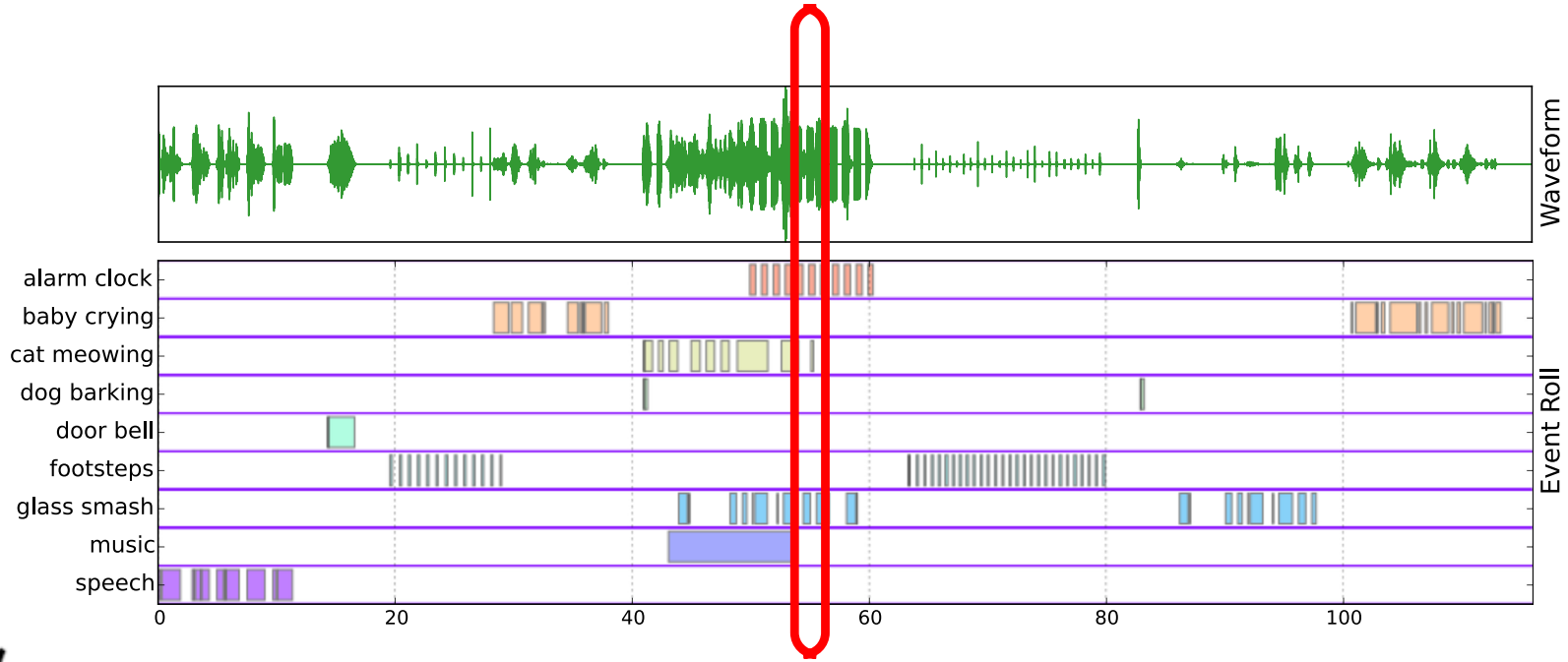
1. Estimate start and end time of each sound event
2. Assign a textual label for each event



TAMPERE UNIVERSITY OF TECHNOLOGY

Polyphonic Sound Event Detection

1. Estimate start and end time of each sound event
2. Assign a textual label for each event



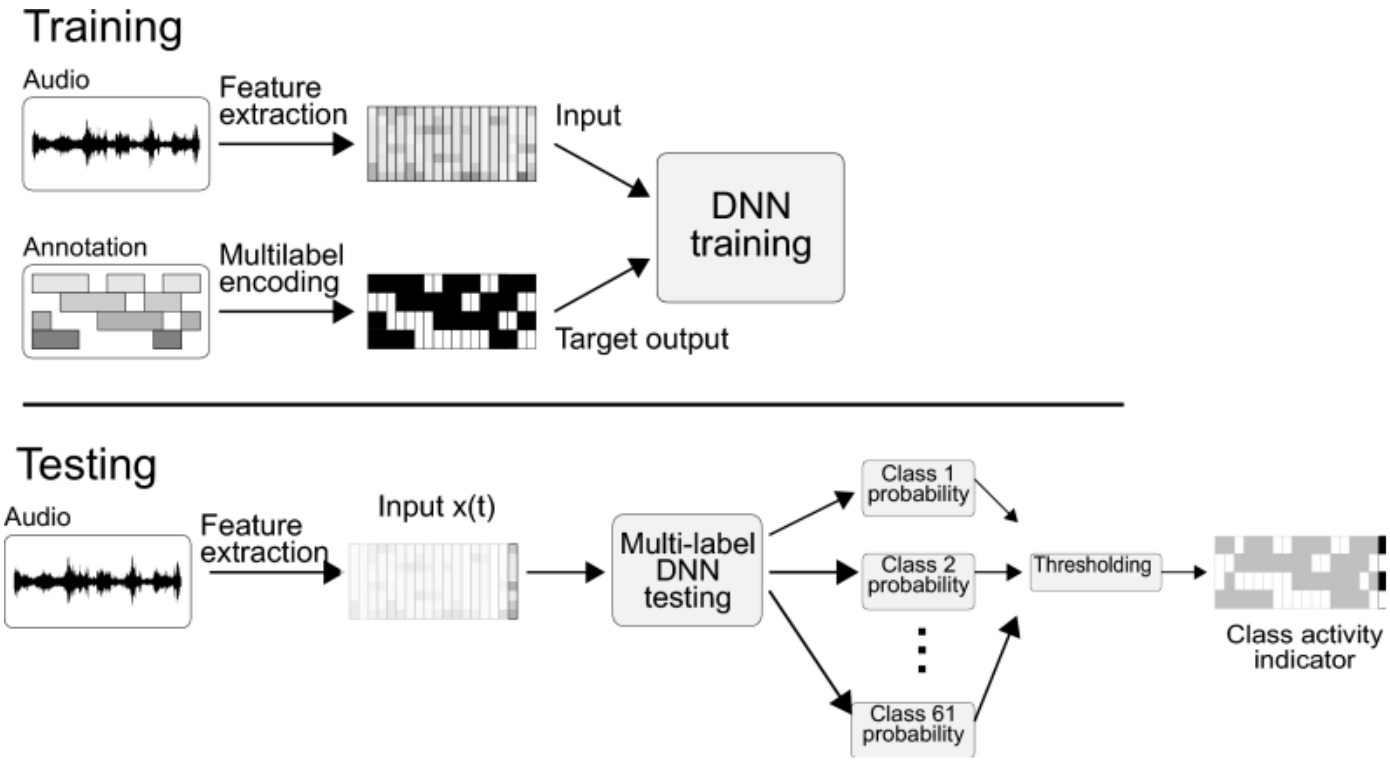
TAMPERE UNIVERSITY OF TECHNOLOGY

Challenges for SED in real-life

- Environmental noise
- Overlapping sound events
- Variation among sources for the same event

TAMPERE UNIVERSITY OF TECHNOLOGY

System overview



TAMPERE UNIVERSITY OF TECHNOLOGY

Polyphonic Sound Event Detection

1. Sound Representation (feature extraction)
2. Sound Event Classification (DNN learning)

TAMPERE UNIVERSITY OF TECHNOLOGY

Polyphonic Sound Event Detection

1. **Sound Representation**
2. Sound Event Classification

TAMPERE UNIVERSITY OF TECHNOLOGY

Sound Representation

Sound representation most often in frequency domain using short audio frames

- More informative than raw audio
- 2-D input images, similar to image recognition / object detection
- Perception based features

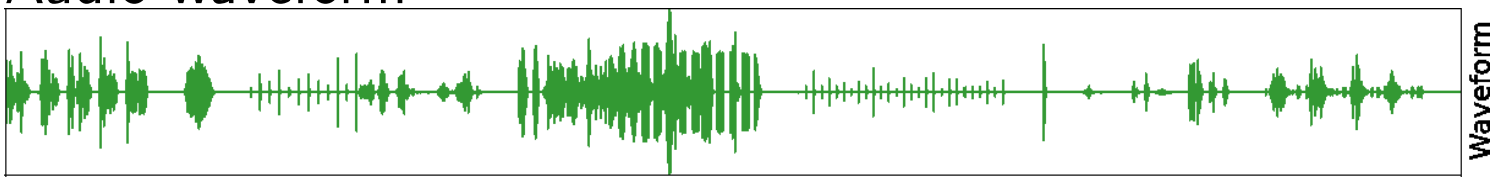
TAMPERE UNIVERSITY OF TECHNOLOGY

Sound representation steps

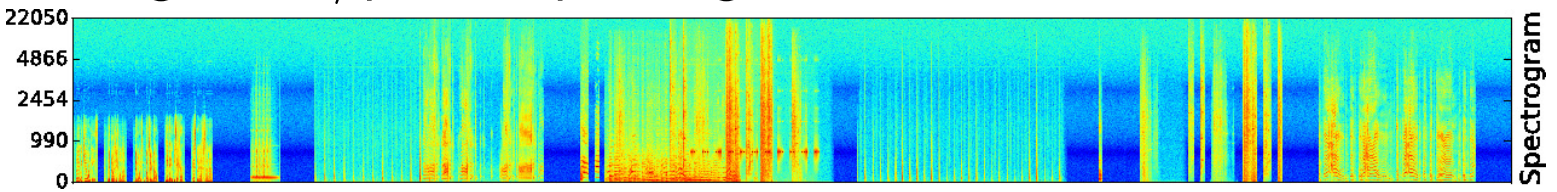
- Frame blocking
- Windowing
- STFT for each short frame
- Perception based post-processing (Mel-band, Bark-band etc.)
- Normalization over each feature

Sound Representation

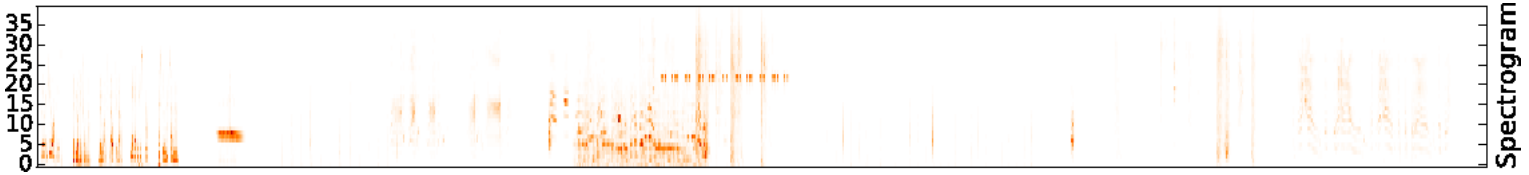
- Audio waveform



- Magnitude/power spectrogram

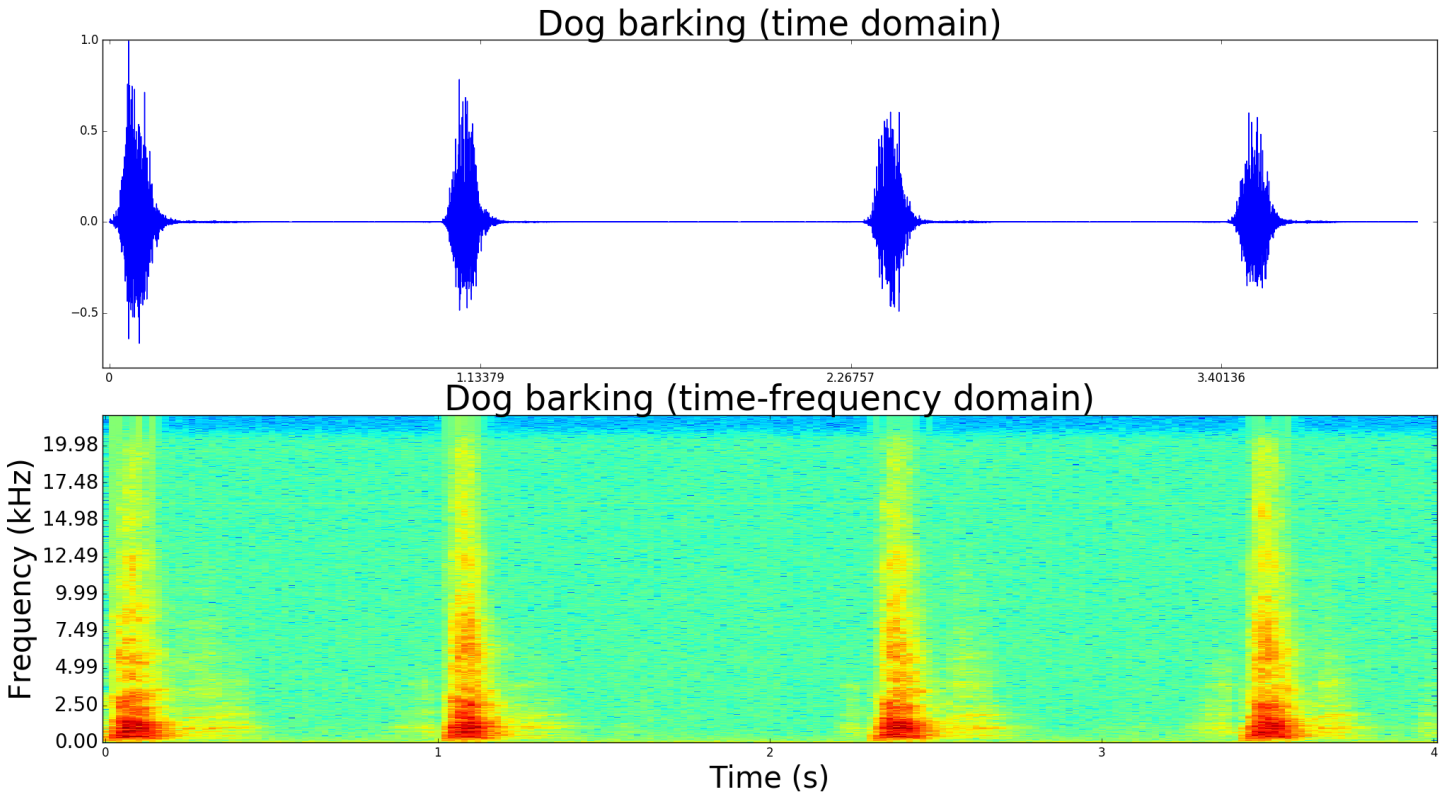


- Mel spectrogram



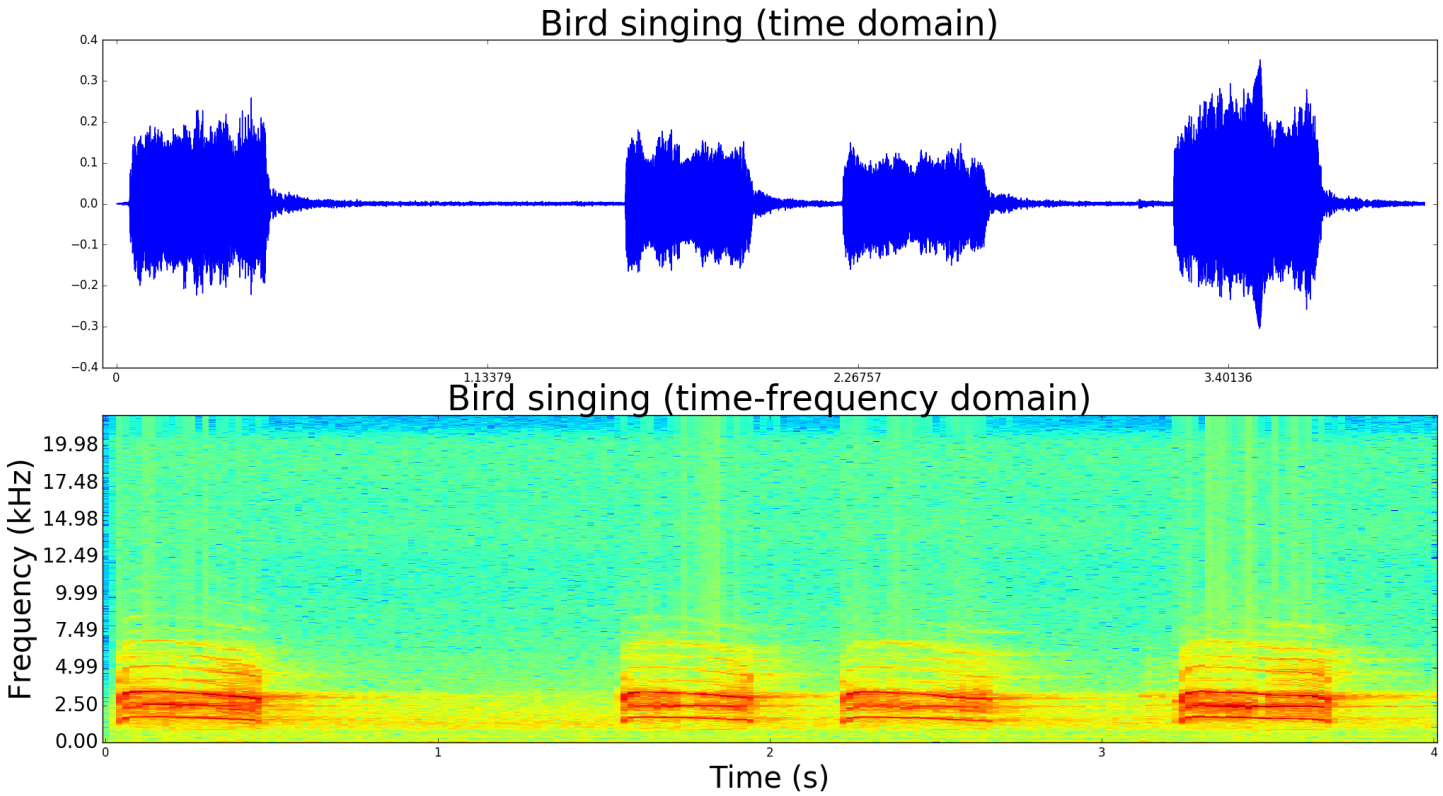
TAMPERE UNIVERSITY OF TECHNOLOGY

Sound Representation



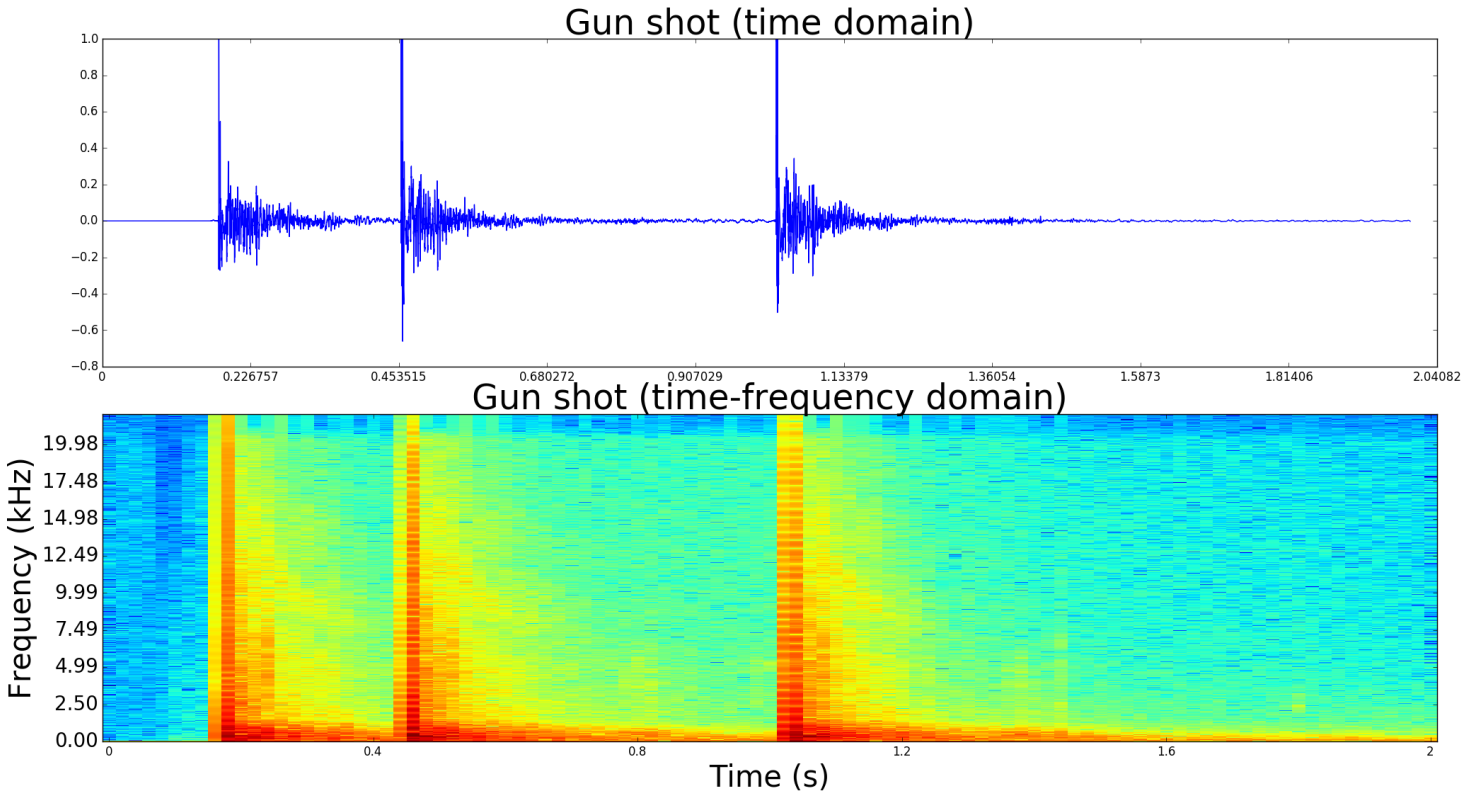
TAMPERE UNIVERSITY OF TECHNOLOGY

Sound Representation



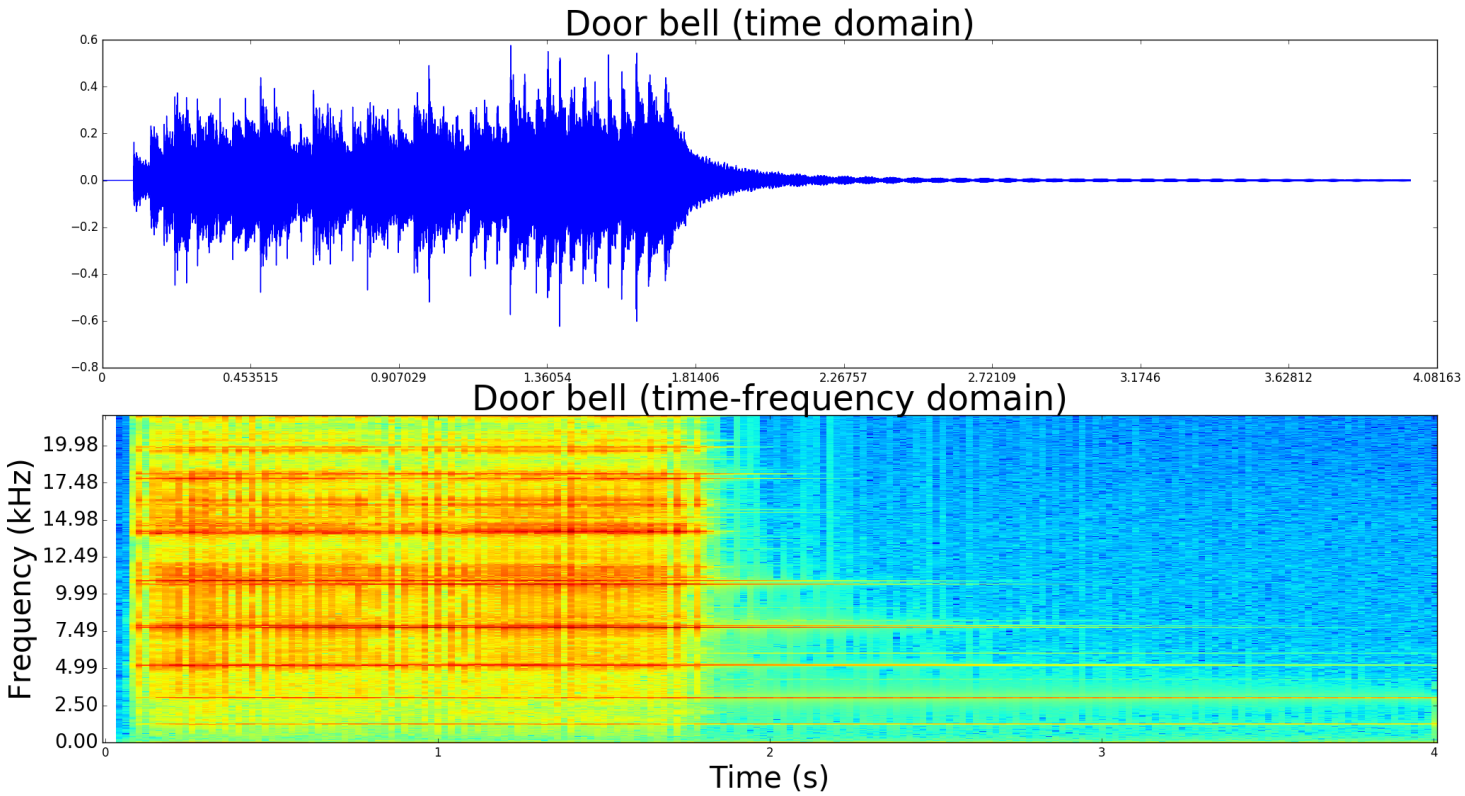
TAMPERE UNIVERSITY OF TECHNOLOGY

Sound Representation



TAMPERE UNIVERSITY OF TECHNOLOGY

Sound Representation



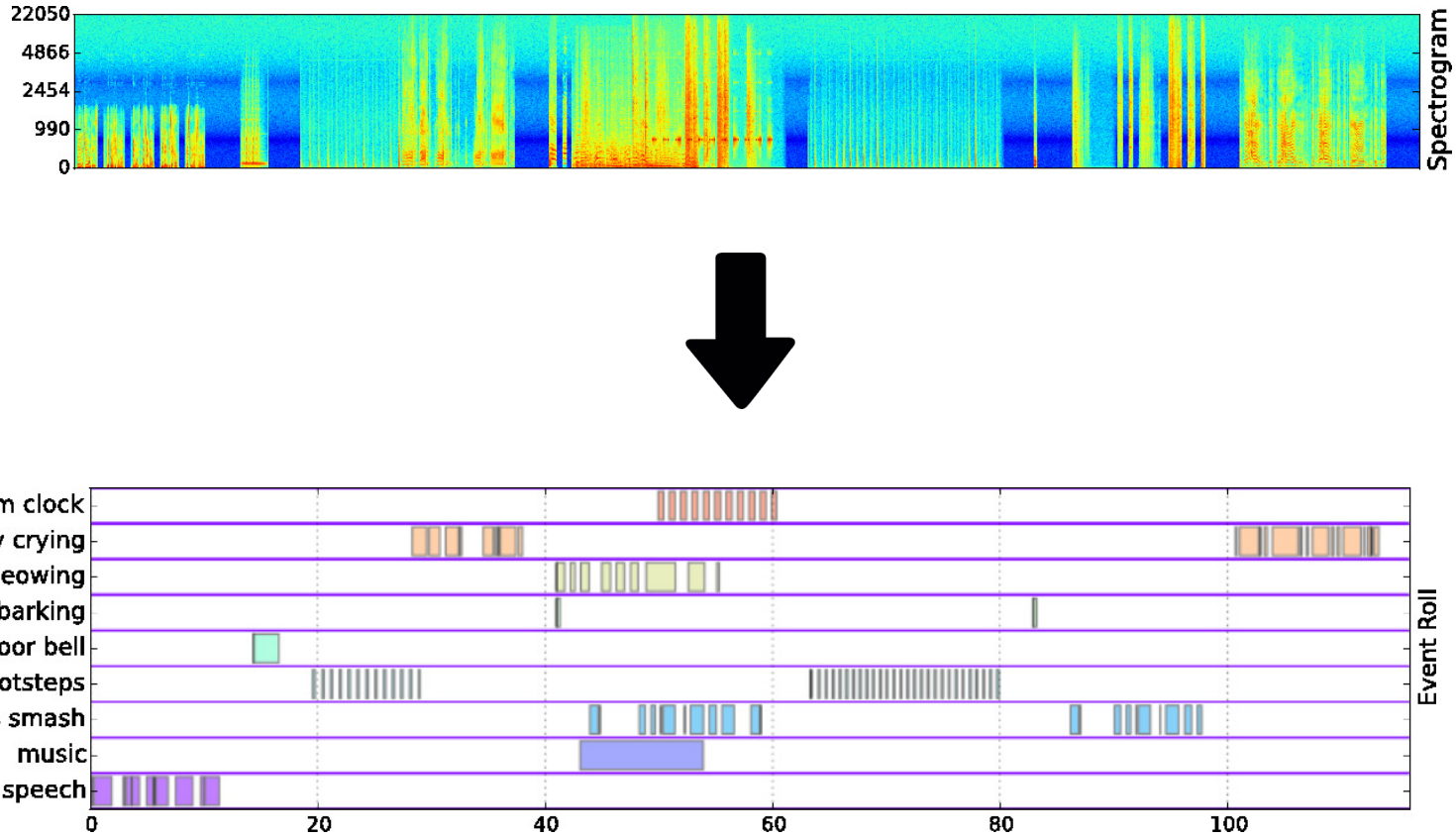
TAMPERE UNIVERSITY OF TECHNOLOGY

Polyphonic Sound Event Detection

1. Sound Representation
2. **Sound Event Classification**

TAMPERE UNIVERSITY OF TECHNOLOGY

Sound Event Classification



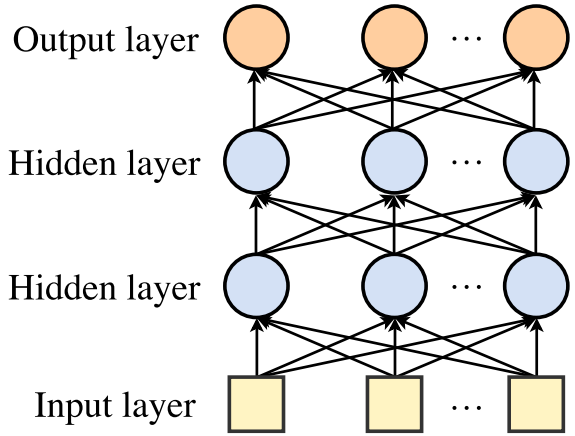
TAMPERE UNIVERSITY OF TECHNOLOGY

Previous work

- Gaussian mixture model (GMM)
- Hidden Markov model (HMM)
- Feedforward neural networks (FNN)
- Convolutional neural networks (CNN)
- Recurrent neural networks (RNN)

TAMPERE UNIVERSITY OF TECHNOLOGY

Previous work (FNN)



- Simple, high expressional power and accuracy compared to established methods (+)
- Context window input to FNN beneficial (+)


TAMPERE UNIVERSITY OF TECHNOLOGY

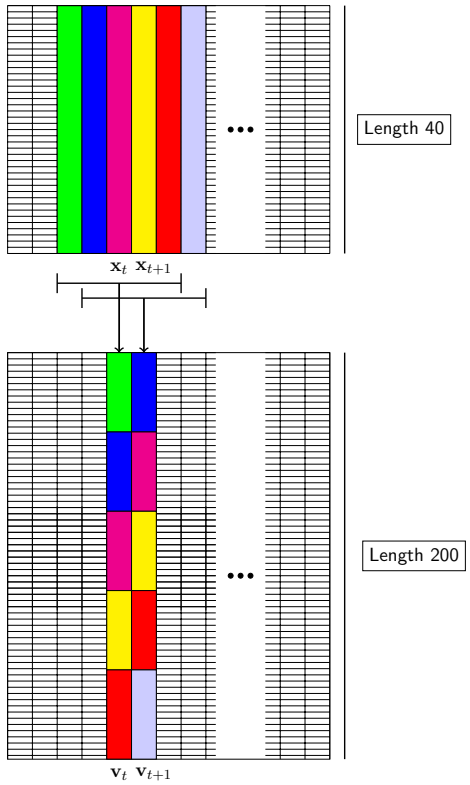
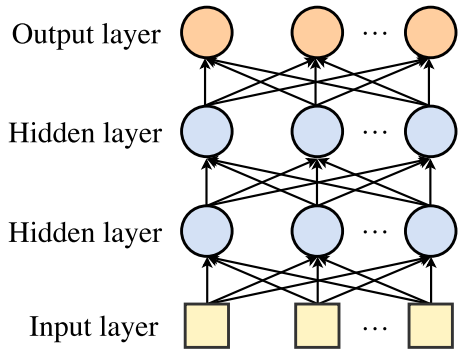


Figure: Concatenation of $\mathbf{x}_t$ with its 2 adjacent frames on both sides to form $\mathbf{v}_t$.

TAMPERE UNIVERSITY OF TECHNOLOGY

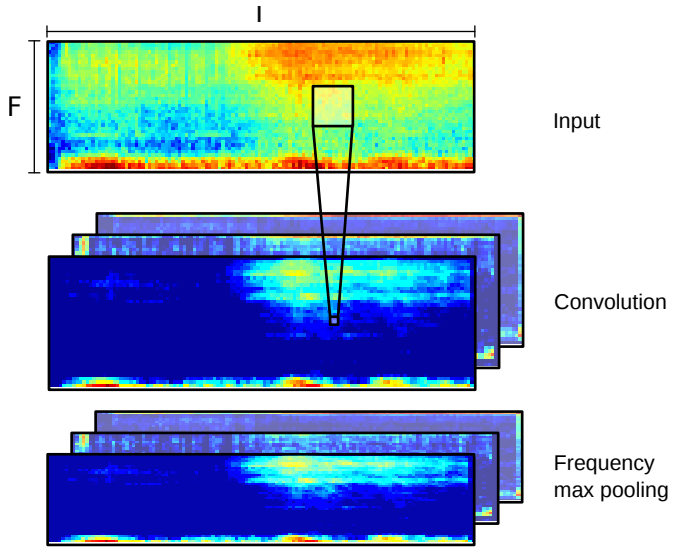
Previous work (FNN)



- Simple, high expressional power and accuracy compared to established methods (+)
- Context window input to FNN beneficial (+)
- Not robust for small variations in frequency content due to fixed connections (-)
- Short temporal context (-)

TAMPERE UNIVERSITY OF TECHNOLOGY

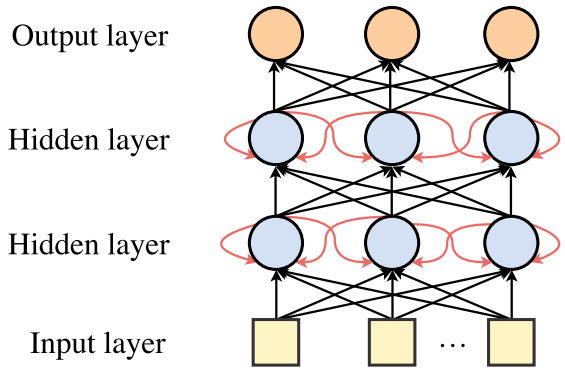
Previous work (CNN)



- Input frequency shift invariance due to shared weights and max pooling (+)
- Short context window (-)


TAMPERE UNIVERSITY OF TECHNOLOGY

Previous work (RNN)



Feedback structure: neuron activations from previous inputs are used while calculating the activation for the current input

- Able to model long-term dependencies between consecutive inputs (+)
- Not robust for variations in frequency content (-)

TAMPERE UNIVERSITY OF TECHNOLOGY

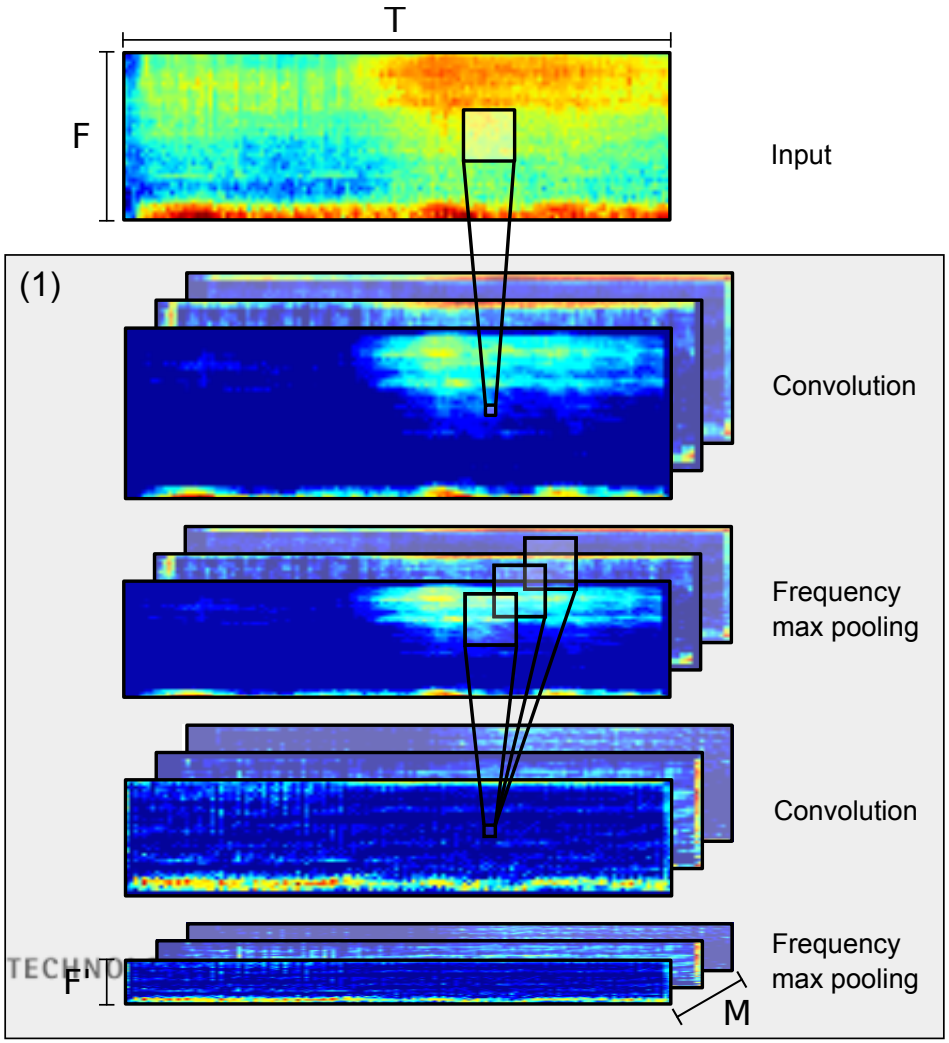
Proposed method

- Combine CNN and RNN in a single deep learning architecture called Convolutional recurrent neural network (CRNN)
- Apply CRNN on polyphonic, scene-independent SED

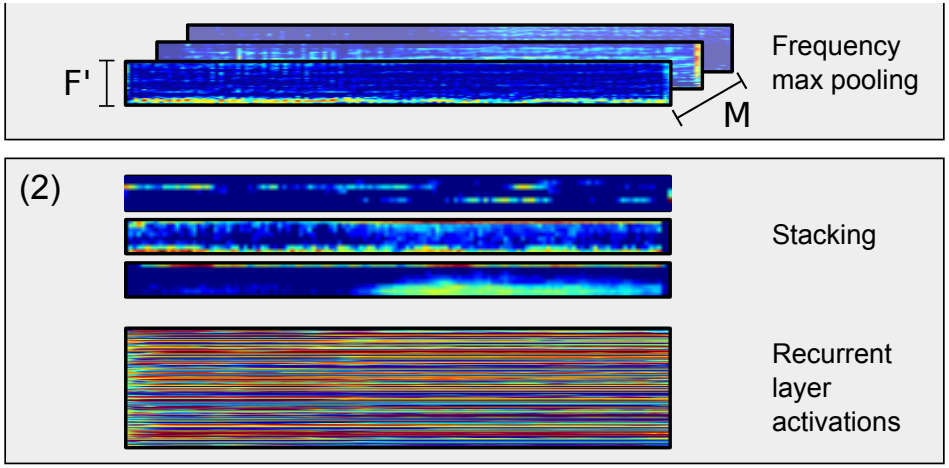
ArXiv: https://arxiv.org/abs/1702.06286

TAMPERE UNIVERSITY OF TECHNOLOGY

Proposed method - Convolutional Recurrent Neural Network

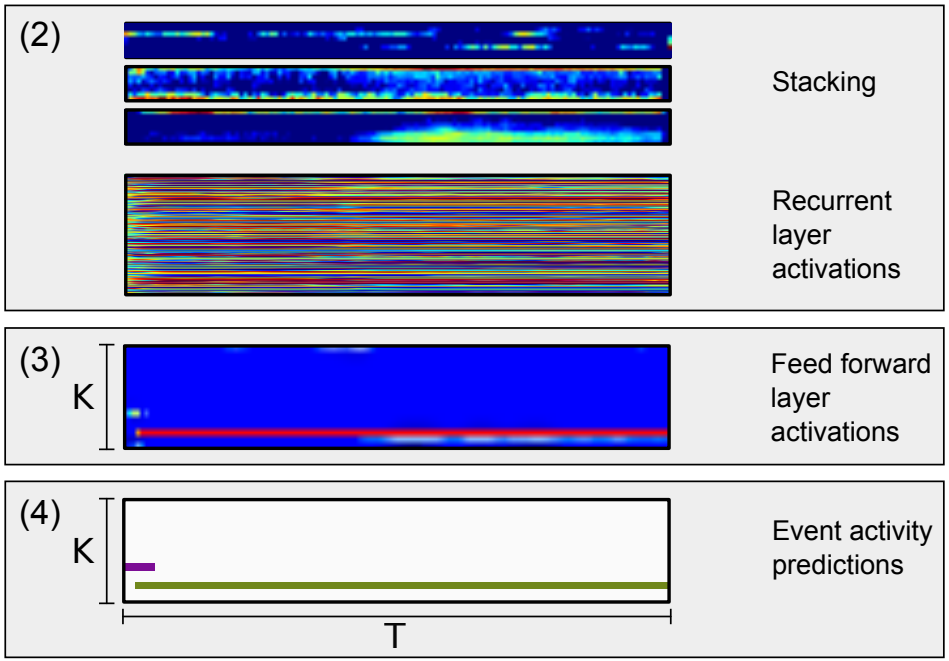


Proposed method - Convolutional Recurrent Neural Network



TAMPERE UNIVERSITY OF TECHNOLOGY

Proposed method - Convolutional Recurrent Neural Network



TAMPERE UNIVERSITY OF TECHNOLOGY

CNN works as a feature extractor, RNN does temporal integration and language modeling.

Google and Baidu successfully used similar architectures for speech recognition.
CNNs and RNNs as subsets of CRNNs

- A CNN is a CRNN with 0 recurrent layers
- An RNN is a CRNN with 0 convolutional layers

TAMPERE UNIVERSITY OF TECHNOLOGY

4 datasets.
Preprocessing: 40 log mel bands, 40ms frames, 50% overlap
Grid-search for network hyper-parameters

- Hidden units for conv and recurr layers $\{96, 256\}$

4 datasets.
Preprocessing: 40 log mel bands, 40ms frames, 50% overlap
Grid-search for network hyper-parameters

- Hidden units for conv and recurr layers $\{96, 256\}$
- Hidden layers 1 to 3 for recurr, 1 to 4 for conv

TAMPERE UNIVERSITY OF TECHNOLOGY

4 datasets.
Preprocessing: 40 log mel bands, 40ms frames, 50% overlap
Grid-search for network hyper-parameters

- Hidden units for conv and recurr layers $\{96, 256\}$
- Hidden layers 1 to 3 for recurr, 1 to 4 for conv
- Several frequency maxpooling arrangements

TAMPERE UNIVERSITY OF TECHNOLOGY

4 datasets.
Preprocessing: 40 log mel bands, 40ms frames, 50% overlap
Grid-search for network hyper-parameters

- Hidden units for conv and recurr layers $\{96, 256\}$
- Hidden layers 1 to 3 for recurr, 1 to 4 for conv
- Several frequency maxpooling arrangements
- Select the best architecture on validation to compute test results

TAMPERE UNIVERSITY OF TECHNOLOGY

- ReLU as hidden activations, sigmoid for the output layer. Cut-off for binary thresholding at 0.5.

- ReLU as hidden activations, sigmoid for the output layer. Cut-off for binary thresholding at 0.5.
- 5×5 kernels in conv layers

- ReLU as hidden activations, sigmoid for the output layer. Cut-off for binary thresholding at 0.5.
- 5×5 kernels in conv layers
- Gated Recurrent Units (GRU) in all recurrent layers. Simpler alternative to LSTM.

- ReLU as hidden activations, sigmoid for the output layer. Cut-off for binary thresholding at 0.5.
- 5×5 kernels in conv layers
- Gated Recurrent Units (GRU) in all recurrent layers. Simpler alternative to LSTM.
- Dropout in all layers 0.25

- ReLU as hidden activations, sigmoid for the output layer. Cut-off for binary thresholding at 0.5.
- 5×5 kernels in conv layers
- Gated Recurrent Units (GRU) in all recurrent layers. Simpler alternative to LSTM.
- Dropout in all layers 0.25
- Batchnormalization in all non-recurrent layers

- ReLU as hidden activations, sigmoid for the output layer. Cut-off for binary thresholding at 0.5.
- 5×5 kernels in conv layers
- Gated Recurrent Units (GRU) in all recurrent layers. Simpler alternative to LSTM.
- Dropout in all layers 0.25
- Batchnormalization in all non-recurrent layers
- Adam as optimizer, cross-entropy as loss

- ReLU as hidden activations, sigmoid for the output layer. Cut-off for binary thresholding at 0.5.
- 5×5 kernels in conv layers
- Gated Recurrent Units (GRU) in all recurrent layers. Simpler alternative to LSTM.
- Dropout in all layers 0.25
- Batchnormalization in all non-recurrent layers
- Adam as optimizer, cross-entropy as loss
- Keras for Python, Theano backend, trained on Tesla GPUs

Results

- Baseline: GMM and FNN
- CNN, RNN and CRNN with hyperparameters from grid search
- Mean and std for ten experiments with different random weight initialization

TAMPERE UNIVERSITY OF TECHNOLOGY

Results - TUT-SED Synthetic 2016

Table: F1 score and error rate results for single frame segments ($F1_{\text{frm}}$ and ER_{frm}) and one second segments ($F1_{\text{1sec}}$ and ER_{1sec}). Bold face indicates the best performing method for the given metric. Standard deviation is presented after $\pm$ sign.

	TUT-SED Synthetic 2016			
Method	$F1_{\text{frm}}$	ER_{frm}	$F1_{\text{1sec}}$	ER_{1sec}
GMM [37]	40.5	0.78	45.3	0.72
FNN [15]	49.2±0.8	0.68±0.02	50.2±1.4	1.1±0.1
CNN	59.8±0.9	0.56±0.01	59.9±1.2	0.78±0.08
RNN	52.8±1.5	0.6±0.02	57.1±0.9	0.64±0.01
CRNN	**66.4±0.6**	**0.48±0.01**	**68.7±0.7**	**0.47±0.01**

TAMPERE UNIVERSITY OF TECHNOLOGY

Effect of convolutional filter shape

Table: $F1_{\text{frm}}$ for accuracy vs. convolution filter shape for TUT-SED Synthetic 2016 dataset. $(*, *)$ represents filter lengths in frequency and time axis, respectively.

Filter shape	(3,3)	(5,5)	(11,11)	(1,5)	(5,1)	(3,11)	(11,3)
$F1_{\text{frm}}$	67.2	**68.3**	62.6	28.5	60.6	67.4	61.2

Thumb rule for CNNs: use 3 x 3 kernels, strided convolutions instead of pooling, global average pooling at the end, many CONV - (L)RELU - BATCHNORM blocks or residual connections.

TAMPERE UNIVERSITY OF TECHNOLOGY

Number of parameters vs. accuracy

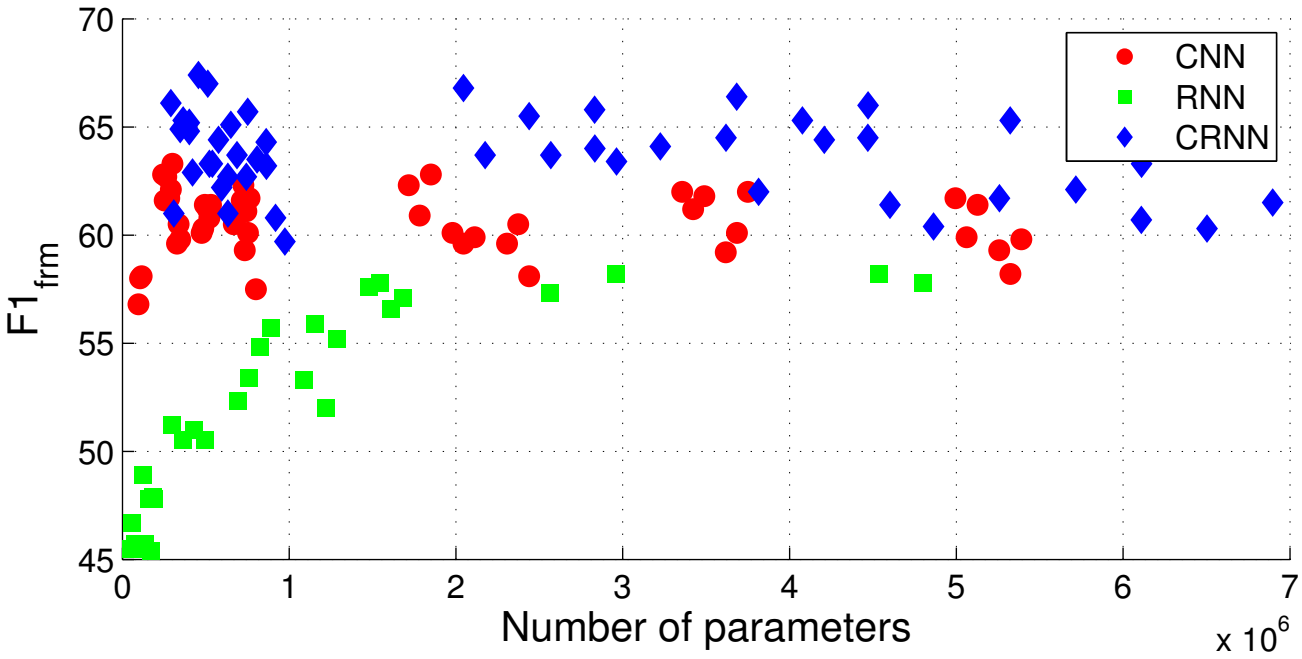


Figure: Number of parameters vs. accuracy for CNN, RNN and CRNN.

TAMPERE UNIVERSITY OF TECHNOLOGY

Frequency shift invariance

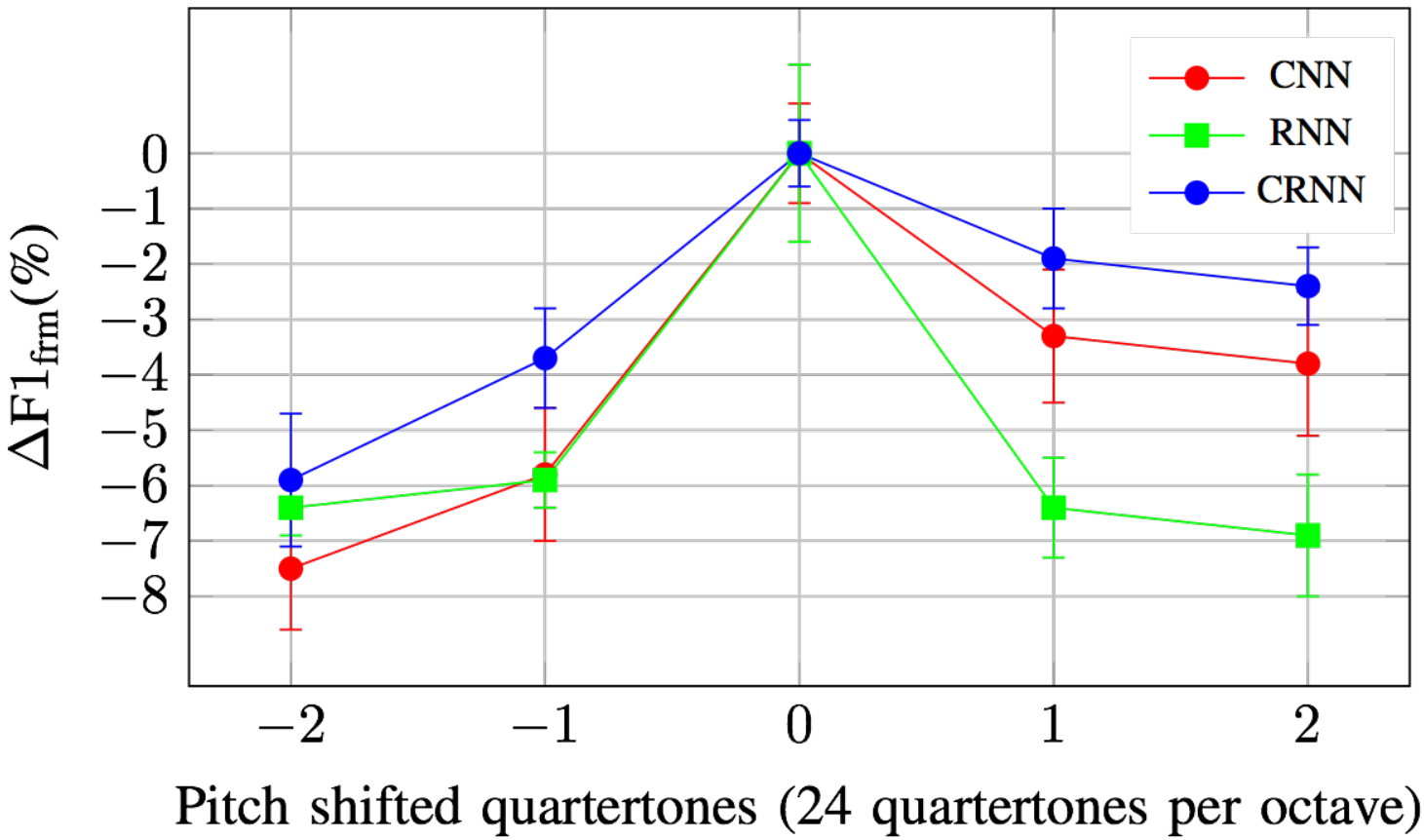
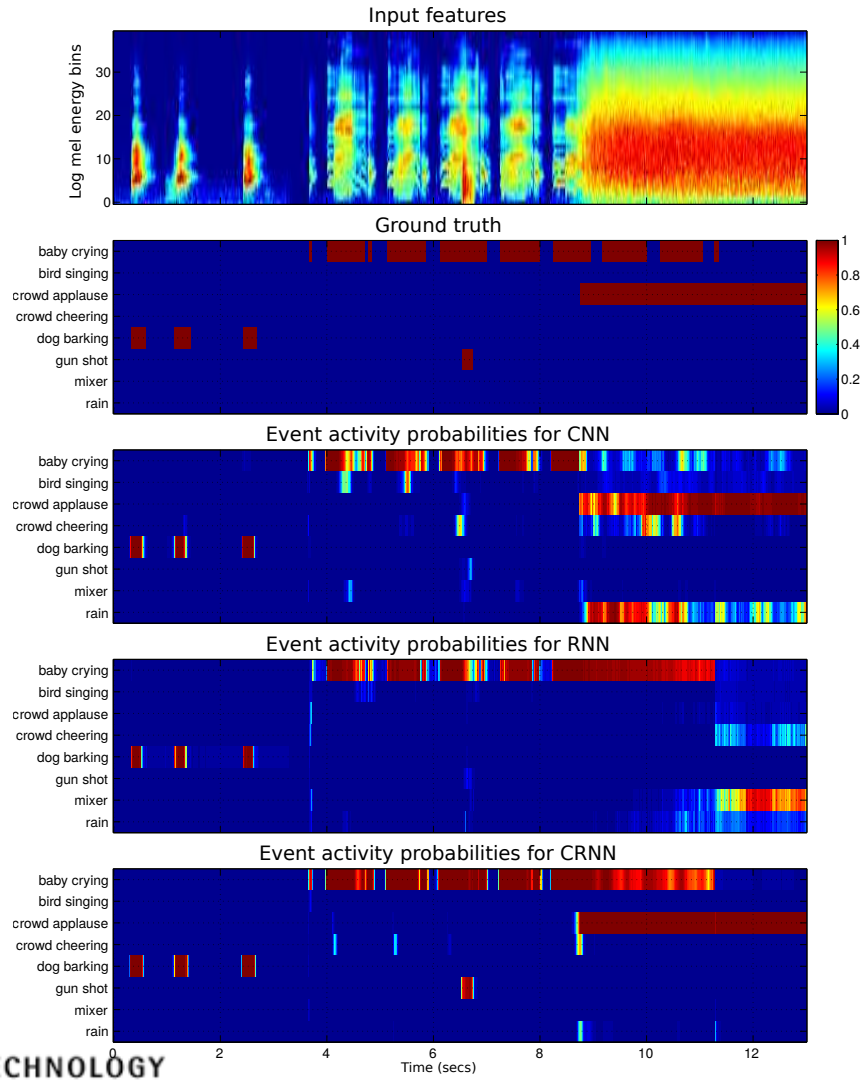


Figure: Accuracy vs. pitch-shifting over ±2 quartertones for CNN, RNN and CRNN.

Closer look on network outputs



TAMPERE UNIVERSITY OF TECHNOLOGY

Figure: Input features, ground truth and event activity probabilities for CNN, RNN

Results - TUT-SED 2009 (CASA)

Method	TUT-SED 2009 - CASA $F1_{\text{frm}}$
GMM [37]	33.0
FNN [15]	60.9±0.4
CNN	64.8±0.2
RNN	62.4±1.0
CRNN	**69.7±0.4**

Table: For a cleaner view here we only show the results using the main metric. The results for the other metrics are consistent with these, and can be found in the paper.

TAMPERE UNIVERSITY OF TECHNOLOGY

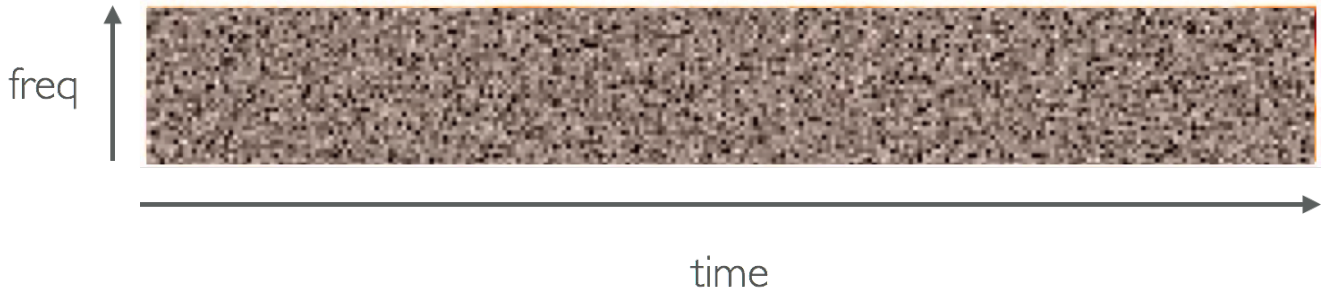
Results - TUT-SED 2009 (CASA)

On the 2 smaller datasets there was little improvement :/

TAMPERE UNIVERSITY OF TECHNOLOGY

What did our neurons learn?

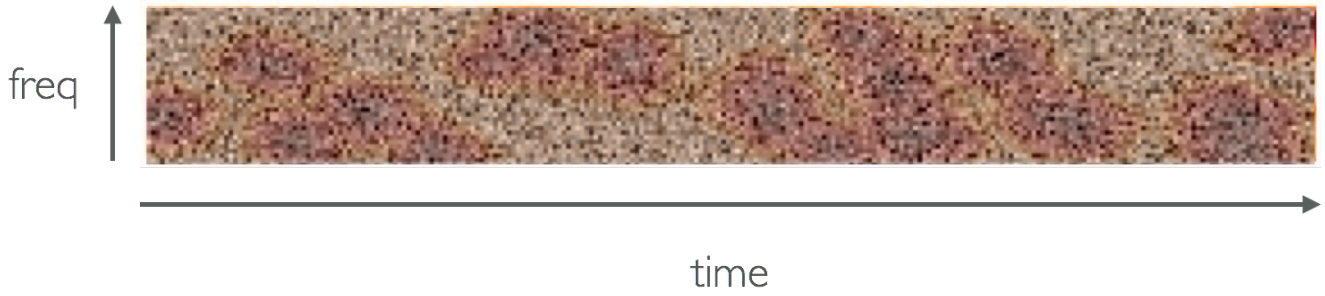
Let's feed a random noise input, compute gradients to maximize the activation of a specific neuron and use them to modify the input and make it more likable to the neuron! (a cool new technique presented a few years ago)



TAMPERE UNIVERSITY OF TECHNOLOGY

What did our neurons learn?

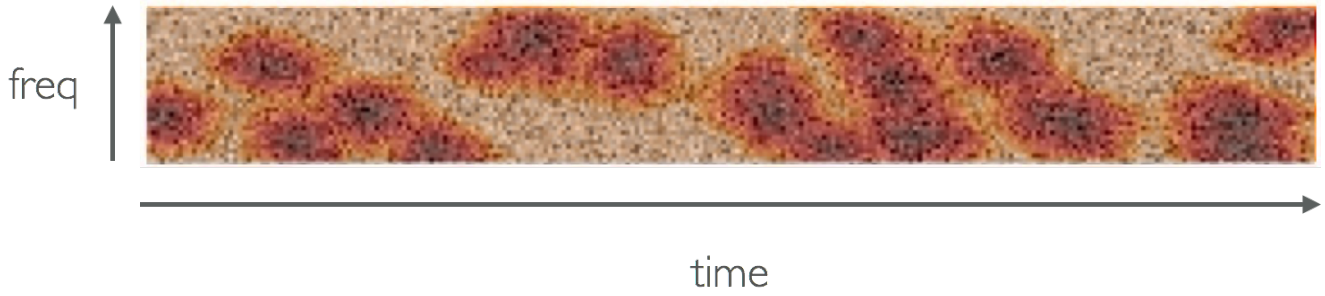
Let's feed a random noise input, compute gradients to maximize the activation of a specific neuron and use them to modify the input and make it more likable to the neuron! (a cool new technique presented a few years ago)



TAMPERE UNIVERSITY OF TECHNOLOGY

What did our neurons learn?

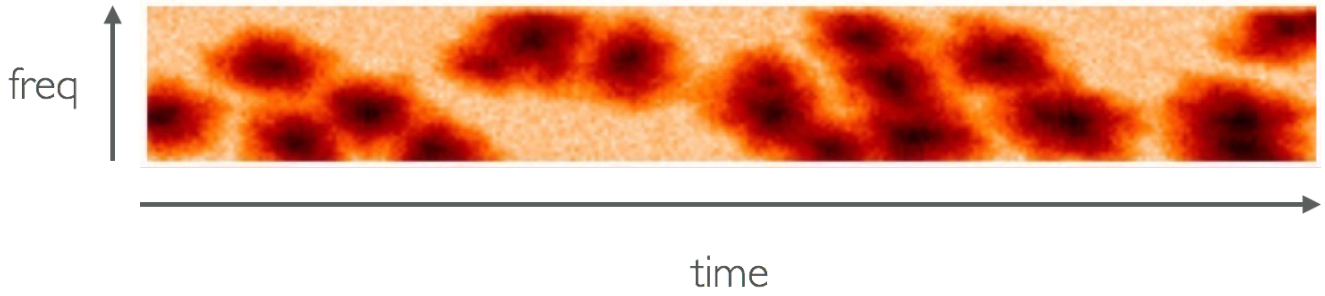
Let's feed a random noise input, compute gradients to maximize the activation of a specific neuron and use them to modify the input and make it more likable to the neuron! (a cool new technique presented a few years ago)



TAMPERE UNIVERSITY OF TECHNOLOGY

What did our neurons learn?

Let's feed a random noise input, compute gradients to maximize the activation of a specific neuron and use them to modify the input and make it more likable to the neuron! (a cool new technique presented a few years ago)



TAMPERE UNIVERSITY OF TECHNOLOGY

Visualization of convolutional layers

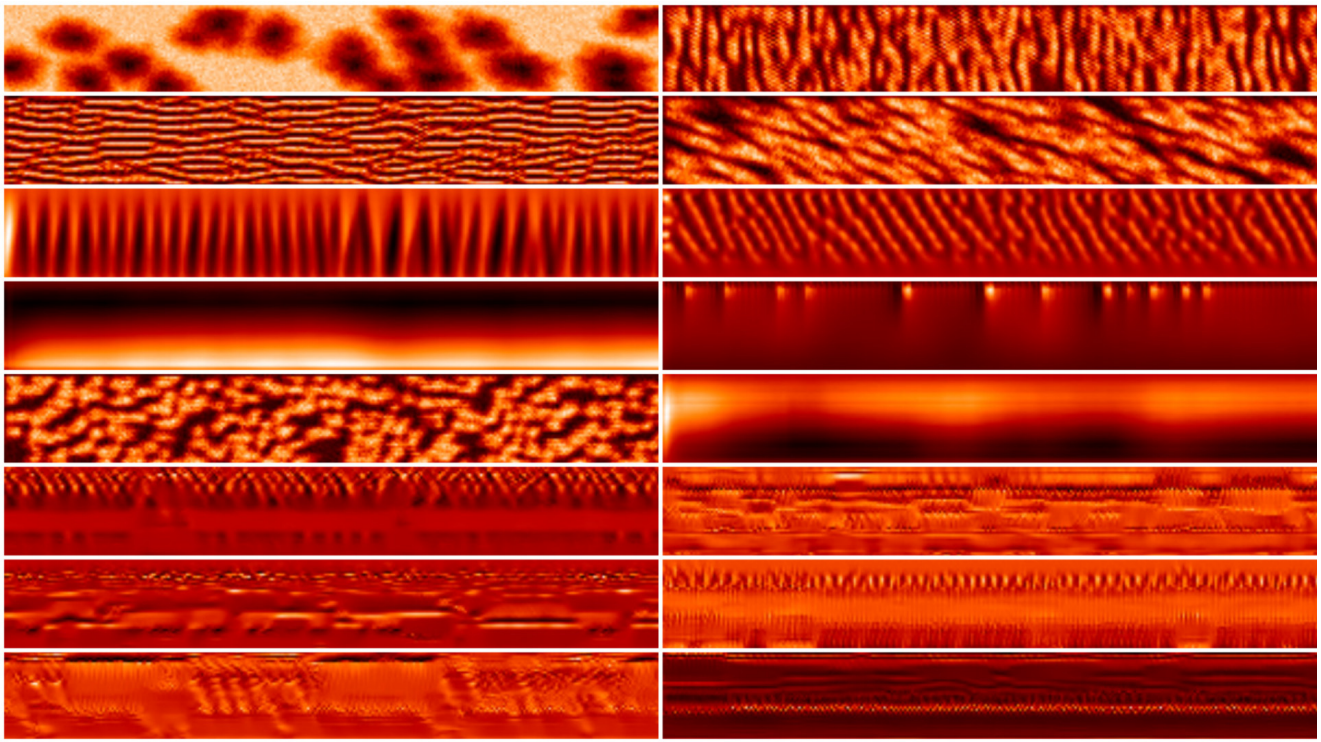


Figure: Two columns of crops from input patterns that would strongly activate certain neurons from different layers of the CRNN. On the horizontal axis is time, on the vertical axis mel bands. On both columns the rows 1 and 2 are from neurons in the first convolutional layer, columns 3 to 5 from the second, and columns from 6 to 8 from the third.

Conclusions

- CRNNs considerably improve the results on SED compared to CNN and RNN alone

TAMPERE UNIVERSITY OF TECHNOLOGY

Conclusions

- CRNNs considerably improve the results on SED compared to CNN and RNN alone
- Small datasets make training very difficult

TAMPERE UNIVERSITY OF TECHNOLOGY

Possible ways to improve?

- ?
- ?
- ?
- ?

TAMPERE UNIVERSITY OF TECHNOLOGY

Possible ways to improve

- Data augmentation

TAMPERE UNIVERSITY OF TECHNOLOGY

Possible ways to improve

- Data augmentation
- Transfer learning (e.g. train the network first on a large dataset, then fine-tune on the smaller one)

TAMPERE UNIVERSITY OF TECHNOLOGY

Possible ways to improve

- Data augmentation
- Transfer learning (e.g. train the network first on a large dataset, then fine-tune on the smaller one)
- Residual connections (Search for ResNets)

TAMPERE UNIVERSITY OF TECHNOLOGY

Possible ways to improve

- Data augmentation
- Transfer learning (e.g. train the network first on a large dataset, then fine-tune on the smaller one)
- Residual connections (Search for ResNets)
- More data

TAMPERE UNIVERSITY OF TECHNOLOGY