Microsoft Research

Georgia Tech

Speech Enhancement with Convolutional-Recurrent Networks

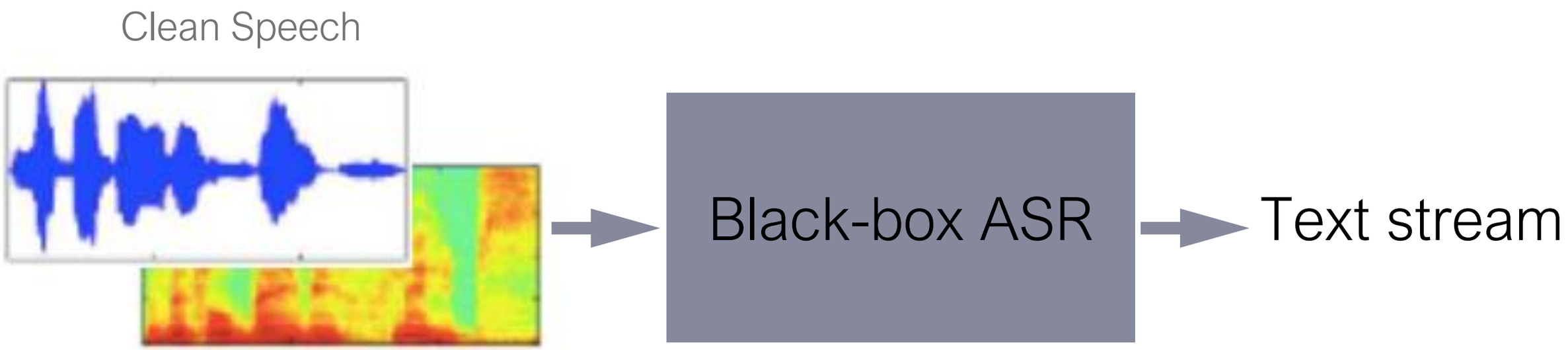
Han Zhao[1], Shuayb Zarar[2], Ivan Tashev[2] and Chin-Hui Lee[3]
Apr. 19th

[1]Machine Learning Department, Carnegie Mellon University
[2]Microsoft Research
[3]School of Electrical Engineering, Georgia Institute of Technology

Carnegie Mellon University

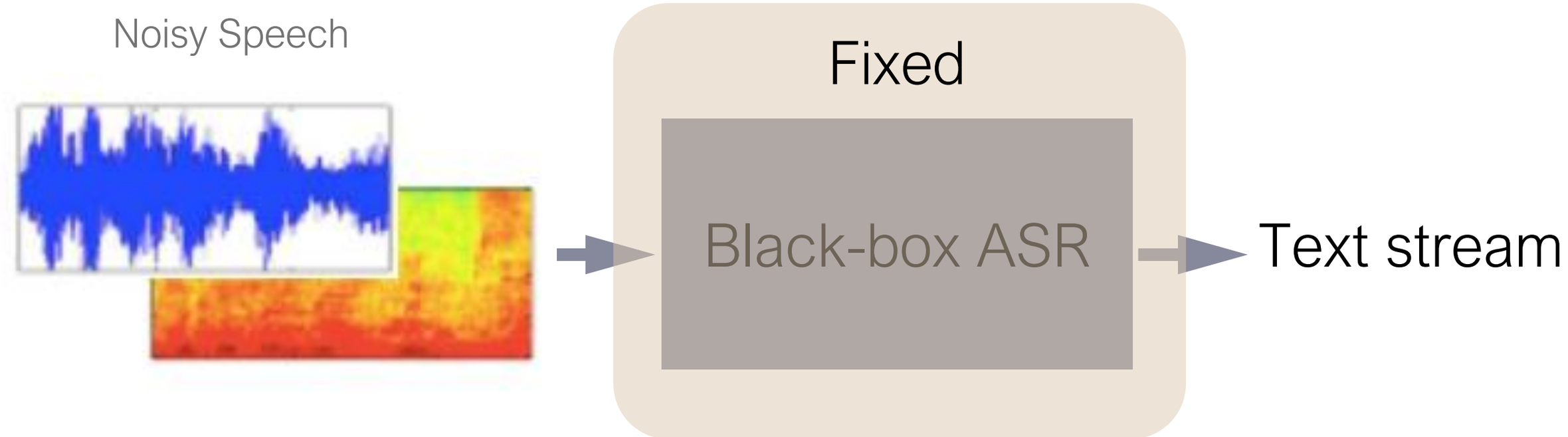
Speech Enhancement — Motivation

ASR system - Training phase



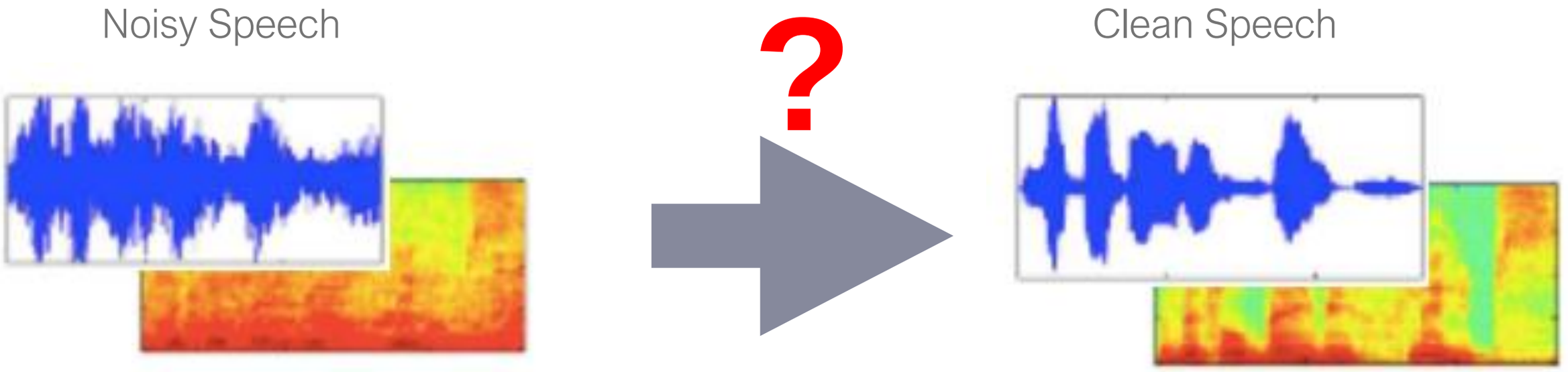
Speech Enhancement — Motivation

ASR system - Inference phase



Speech Enhancement — Motivation

Distribution mismatch



- Similar issues with rendering and perception
- Clean speech is preferred for playback

Speech Enhancement — Motivation

Speech enhancement: from noisy to clean

Outline

- **Background**
- **Data-driven Approach**
- **Convolutional-Recurrent Network for Speech Enhancement**
- **Conclusion**

Background

Problem setup:

Clean signal

$$\hat{x}(t) = x(t) + \varepsilon(t)$$

Noisy signal

(Unknown) noise

Typical assumptions on noise:

- Stationarity: $\varepsilon(t)$ is independent of t
- Noise type: $\varepsilon(t) \sim \mathcal{N}(0, \sigma^2)$

Classic methods: spectral subtraction (Boll 1979), Minimum mean-squared error estimator (Ephraim et al. 1984), Subspace approach (Ephraim et al. 1995)

Background

Classic methods are based on statistical assumptions of noise:

Pros:

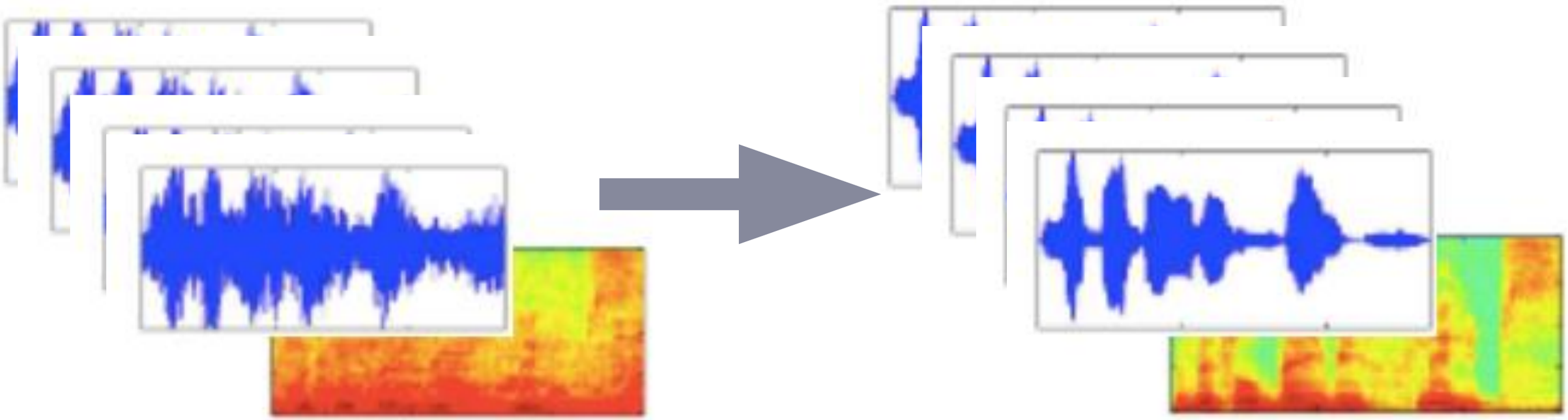
- Simple, and computationally efficient
- Optimality under proper assumption
- Interpretable

Cons:

- Limited to stationary noise
- Restricted to noise with specific characteristics

Data-driven Approach

What if we can collect large datasets of paired signals?



Data-driven Approach

What if we can collect large datasets of paired signals?

Given:

- Paired signals $\{(\hat{x}_i, x_i)\}_{i=1}^{n}, \hat{x}, x \in \mathbb{R}^d$

Goal:

- Build function approximator h such that

$$h(\hat{x}_i) = x_i, \forall i$$

In short: regression based approach, usually $n \sim 10^7, d \sim 256$

Data-driven Approach

$$n \sim 10^7, d \sim 256$$

Parametric regression using Neural Networks:

- Flexible for representation learning
- Scale linearly in n and d
- Natural paradigm for multi-task learning by sharing common representations

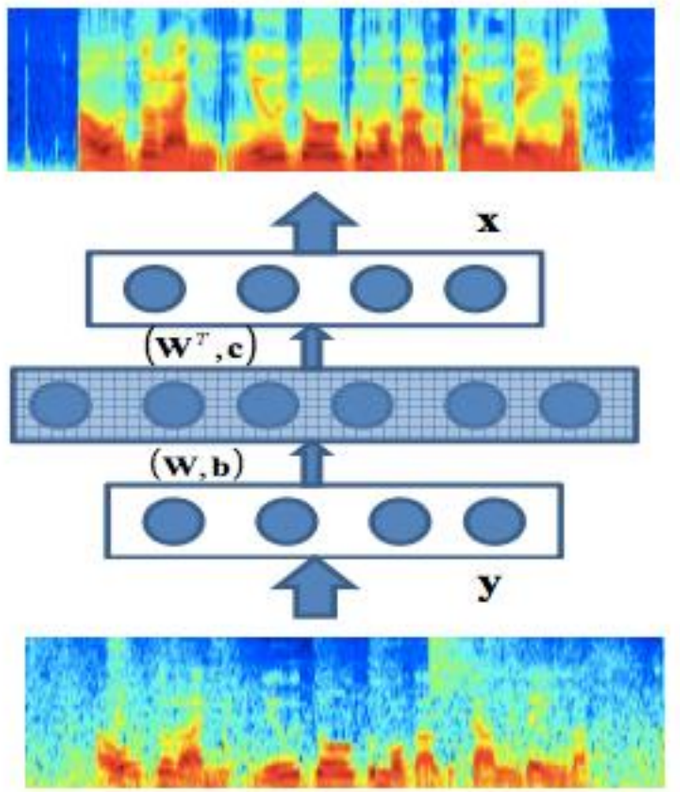


Figure from Lu el al., Interspeech 2013

Data-driven Approach

Related work for speech enhancement

- Recurrent network for noise reduction, Maas et al., ISCA 2012
- Deep denoising auto-encoder, Lu et al., Interspeech 2013
- Weighted denoising auto-encoder, Xia et al., Interspeech 2013
- DNN with symmetric context window, Xu et al., IEEE SPL 2014
- Hybrid of DNN suppression rule, Mirsamadi et al., Interspeech 2016

Data-driven Approach

Speech Enhancement Pipeline:

- Short-term Fourier Transform (STFT) to obtain time-frequency signal $X(\tau, f)$

STFT

$x(t)$ → $X(\tau, f)$

- Build neural networks to approximate filter function h such that

$$h(\hat{X}) \approx X$$

Focus of this talk

- Apply Inverse-STFT (ISTFT) to reconstruct sound wave

ISTFT($h(\hat{X})$)

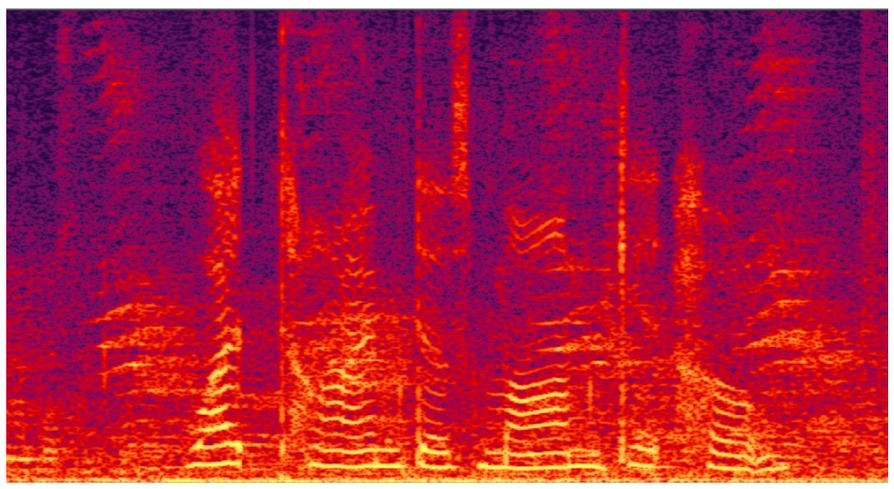
Convolutional-Recurrent Networks for SE

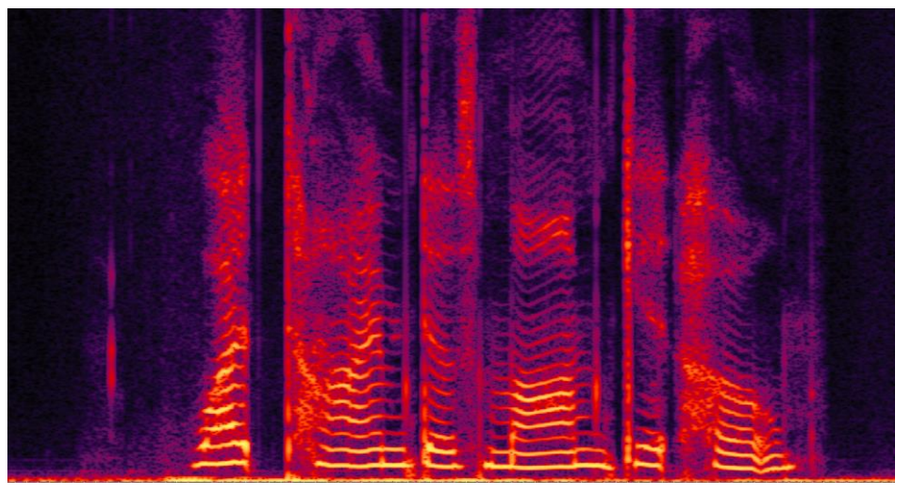
Problem setup:

Given time-frequency signal — spectrogram pair

$$\{(\hat{X}_i(\tau, f), X_i(\tau, f))\}_{i=1}^{n}$$

where $\hat{X}_i, X_i \in \mathbb{R}_+^{t_i \times d}$





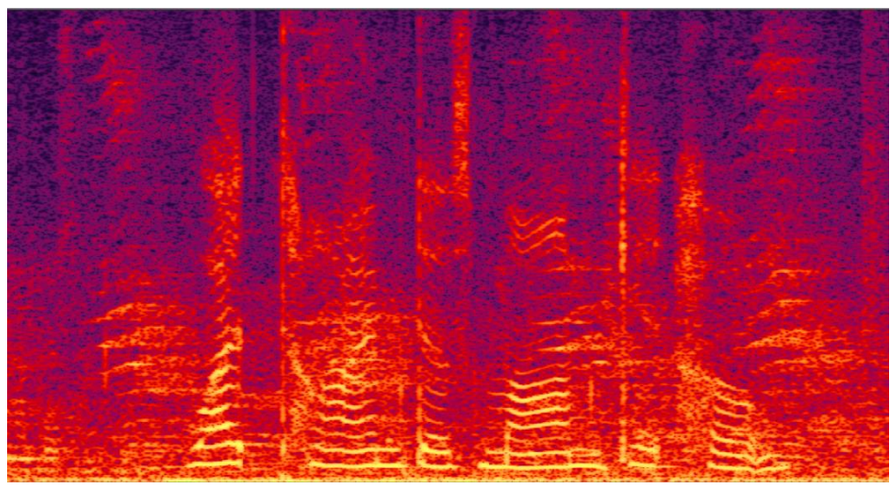
For each utterance, usually $t_i \sim 500$ frames and $d = 256$ frequency bins.

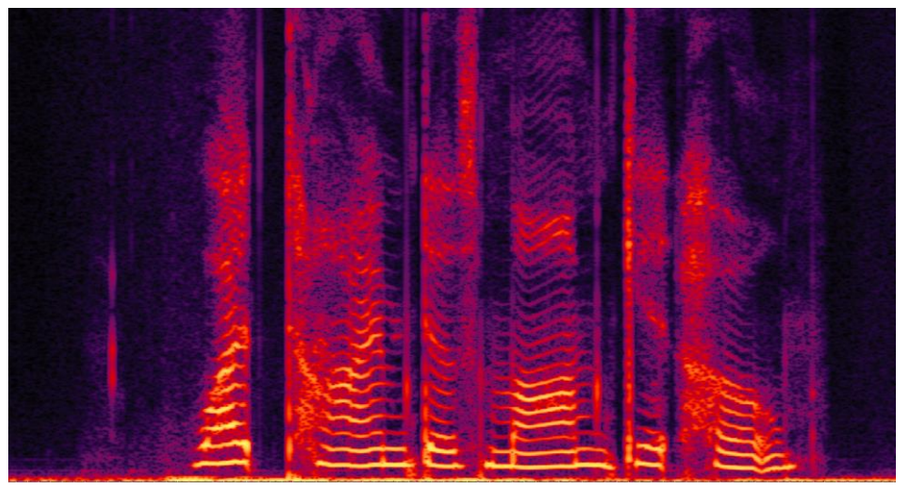
Convolutional-Recurrent Networks for SE

Observations:

Existing DNN-based approaches do not fully exploit the structure of speech signals.

- Frame-based DNN regression approach does not use the temporal locality of spectrogram
- Fully connected DNN regression approach does not exploit the continuity of consecutive frequency bins in spectrogram





Convolutional-Recurrent Networks for SE

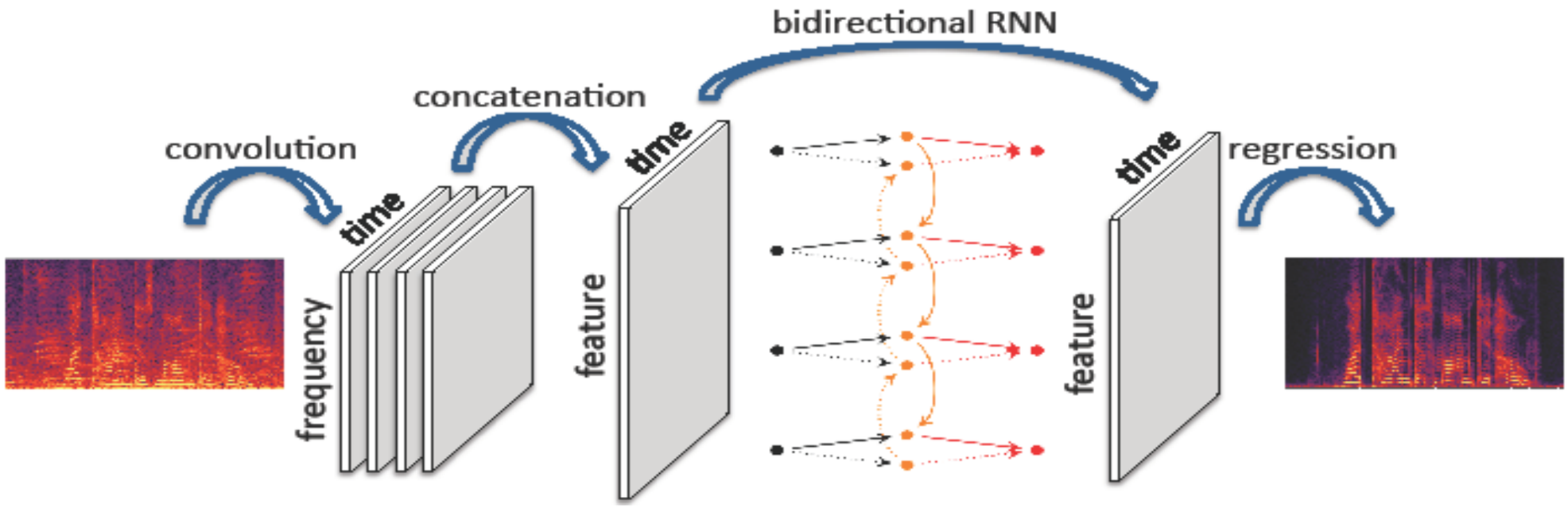
Observations:

Existing DNN-based approaches do not fully exploit the structure of speech signals.

- Frame-based DNN regression approach does not use the temporal locality of spectrogram
  - Use recurrent neural networks
- Fully connected DNN regression approach does not exploit the continuity of consecutive frequency bins in spectrogram
  - Use convolutional neural networks

Convolutional-Recurrent Networks for SE

Proposed: Convolution + bi-LSTM + Linear Regression

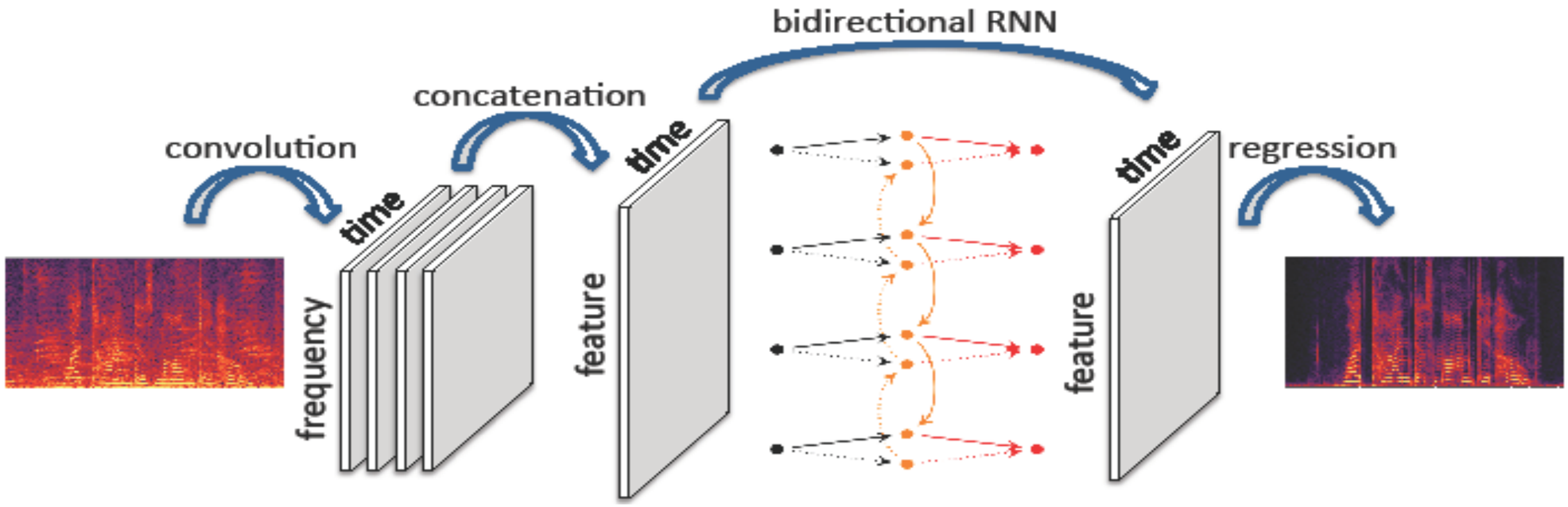


Objective:

$$\min_\theta \quad \sum_{i=1}^{n} ||X_i - h(\hat{X}_i; \theta)||_F^2$$

Convolutional-Recurrent Networks for SE

Proposed: Convolution + bi-LSTM + Linear Regression

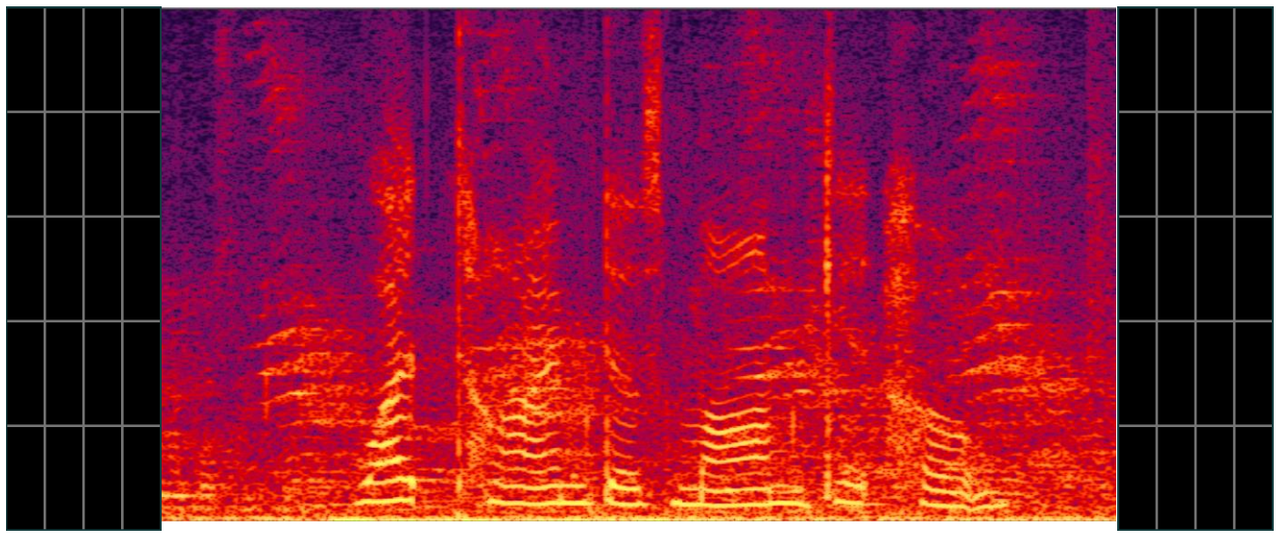


At a high level, why will this model work?

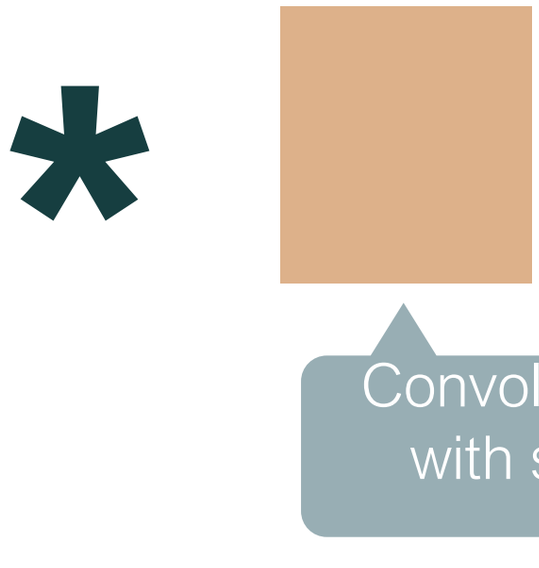
- Continuity of signal in time and frequency domains
- Convolution kernels as linear filters to match local patterns
- bi-LSTM -> symmetric context window with adaptive window size
- End-to-end learning without additional assumptions on noise type

Convolutional-Recurrent Networks for SE

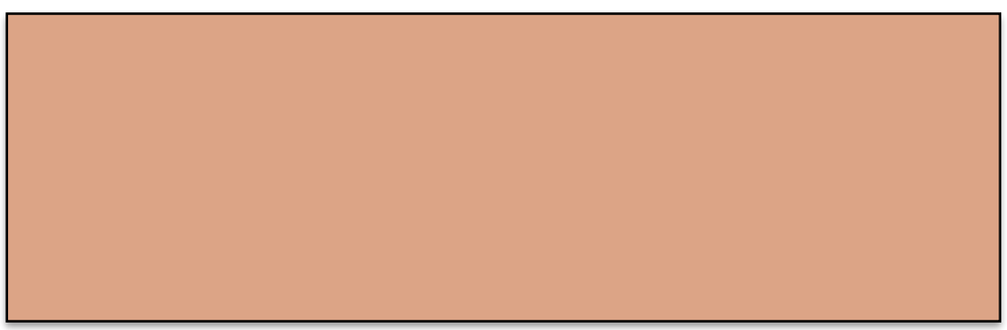
Convolution



Zero-padded spectrogram (t, f)



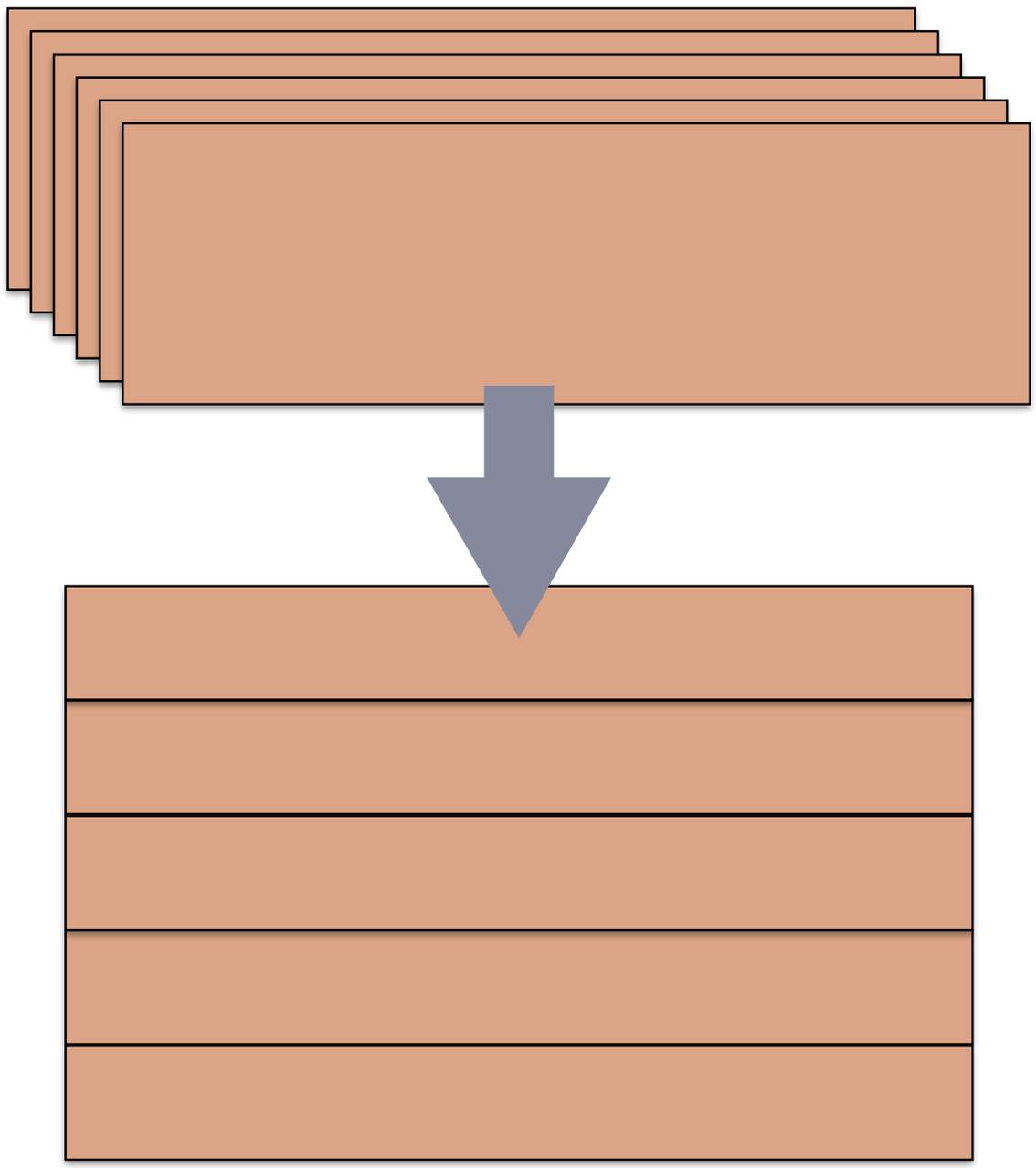
Convolution kernel with size (b, w)



feature map of size (t, f')

Convolutional-Recurrent Networks for SE

Concatenation of feature maps

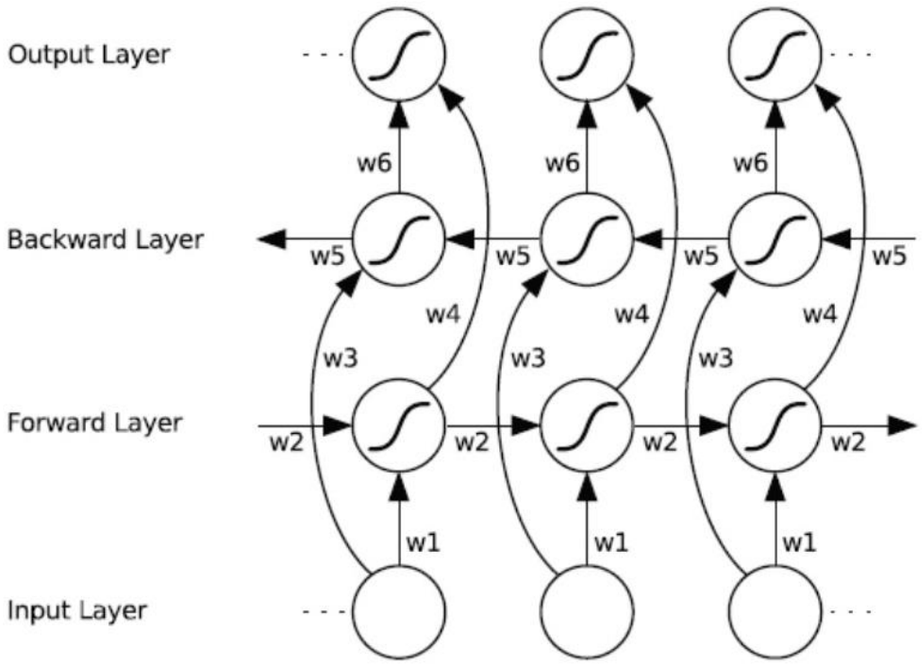


k feature maps, each with size (t, f')

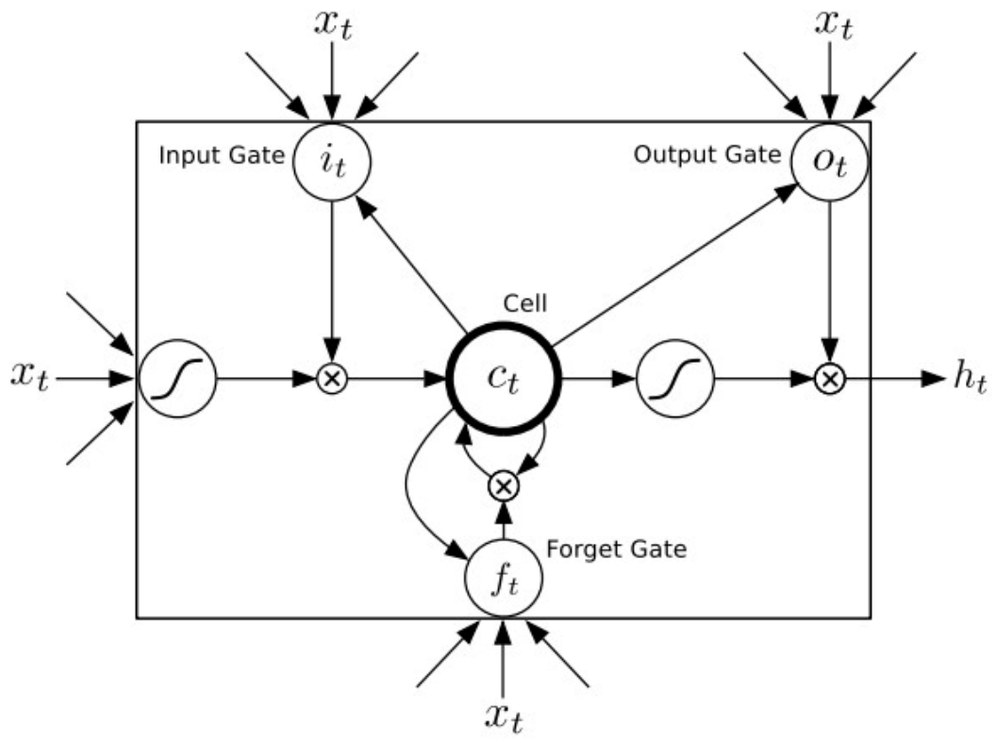
One feature map, with size (t, kf')

Convolutional-Recurrent Networks for SE

bi-directional LSTM



State transition function of LSTM cell:



Convolutional-Recurrent Networks for SE

Linear Regression with $\mathbb{R}_+^d$ Projection

At each time step t:

$$\tilde{X}_t = \max(0, W o_t + b)$$

where o_t is the output state of bi-LSTM at time step t.

Objective function and Optimization

$$\tilde{X}_t = \max(0, W o_t + b)$$

MSE:

$$\min \quad \sum_{i=1}^{n} ||X_i - \tilde{X}_i||_F^2$$

Optimization algorithm: AdaDelta

Experiments

Dataset

Single channel, Microsoft-internal data

- Cortana utterances: male, female and children
- Sampling rate: 16kHz
- Storage format: 24bits precision
- Each utterance: 5~9 seconds
- Noise: subset of MS noise collection, 377 files with 25 types
- 48 room impulse responses from MS RIR collection

	Training	Validation	Test (seen noise)	Test (unseen noise)
# utterances	7,500	1,500	1,500	1,500

Experiments

Evaluation Metric

- Signal-to-Noise Ratio (SNR) dB
- Log-spectral Distance (LSD)
- Mean-squared Error in time domain (MSE)
- Word error rate (WER)
- Perceptual evaluation of speech quality P.862 (PESQ)

Experiments

Comparison with State-of-the-Art Methods

- Classic noise suppressor
- DNN-Symmetric (Xu et al. 2015)
  - Multilayer perceptron, 3 hidden layers (2048x3), 11 context window
- DNN-Causal (Tashev et al. 2016)
  - Multilayer perceptron, 3 hidden layers (2048x3), 7 causal window
- Deep-RNN (Maas et al. 2012)
  - Recurrent autoencoders, 3 hidden layers (500x3), 3 context window

All models are trained using AdaDelta

Experiments

Comparison with State-of-the-Art Methods (seen noise)

	SNR	LSD	MSE	WER	PESQ
Noisy data	15.18	23.07	0.04399	15.40	2.26
Classic NS	18.82	22.24	0.03985	14.77	2.40
DNN-s	44.51	19.89	0.03436	55.38	2.20
DNN-c	40.70	20.09	0.03485	54.92	2.17
RNN	41.08	17.49	0.03533	44.93	2.19
Ours	**49.79**	**15.17**	**0.03399**	**14.64**	**2.86**
Clean data	57.31	1.01	0.0000	2.19	4.48

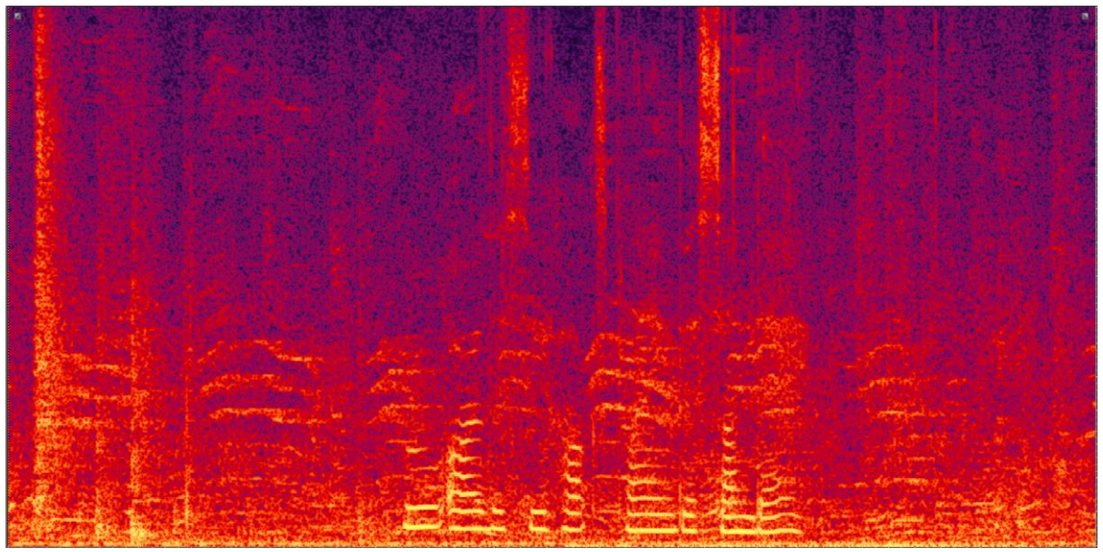
Experiments

Comparison with State-of-the-Art Methods (unseen noise)

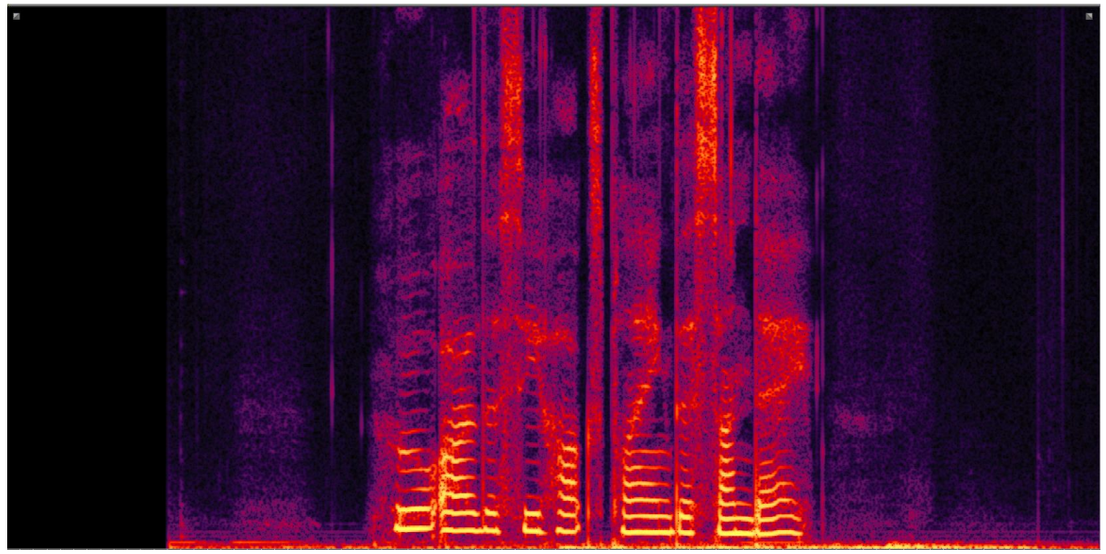
	SNR	LSD	MSE	WER	PESQ
Noisy data	14.78	23.76	0.04786	18.40	2.09
Classic NS	19.73	22.82	0.04201	**15.54**	2.26
DNN-s	40.47	21.07	0.03741	54.77	2.16
DNN-c	38.70	21.38	0.03718	54.13	2.13
RNN	**44.60**	18.81	**0.03665**	52.05	2.06
Ours	39.70	**17.06**	0.04721	16.71	**2.73**
Clean data	58.35	1.15	0.0000	1.83	4.48

Experiments

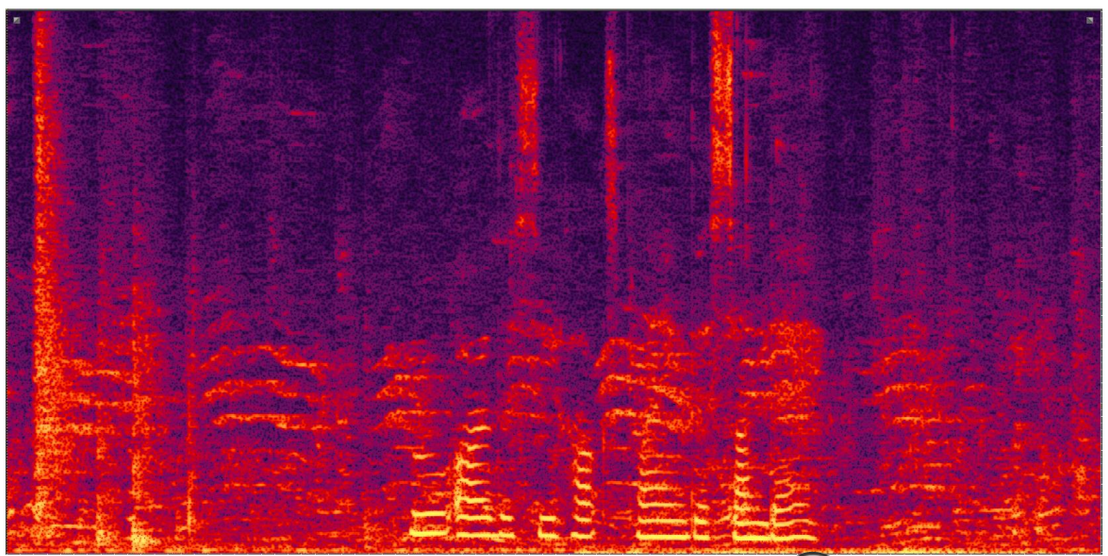
Case Study



Noisy



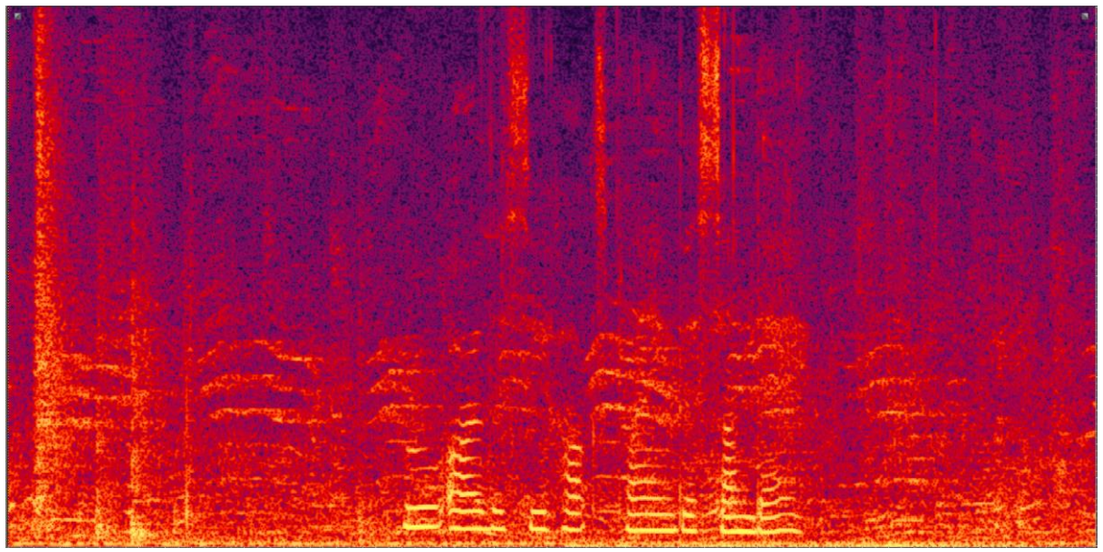
Clean



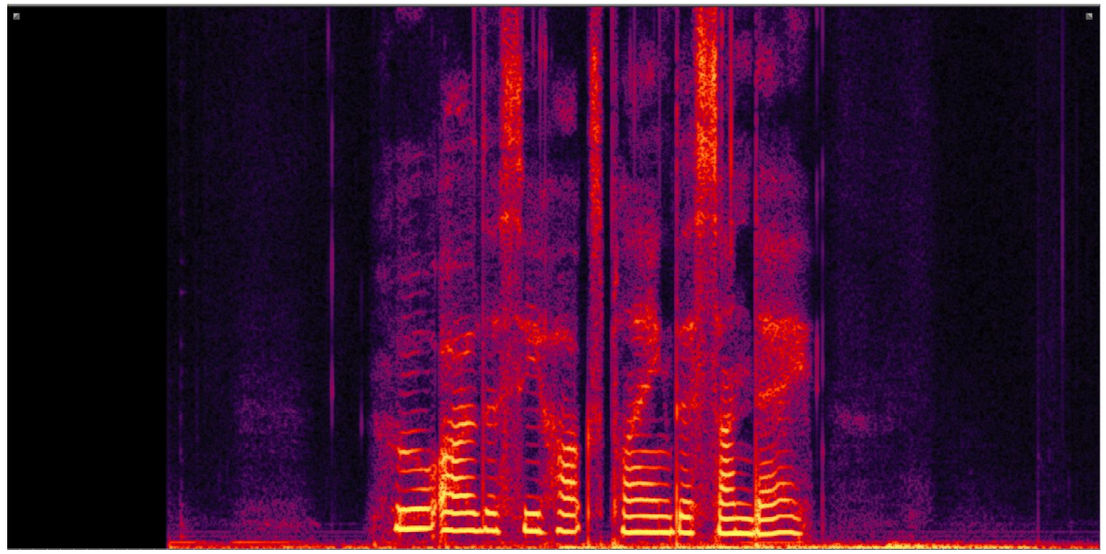
MS-Cortana

Experiments

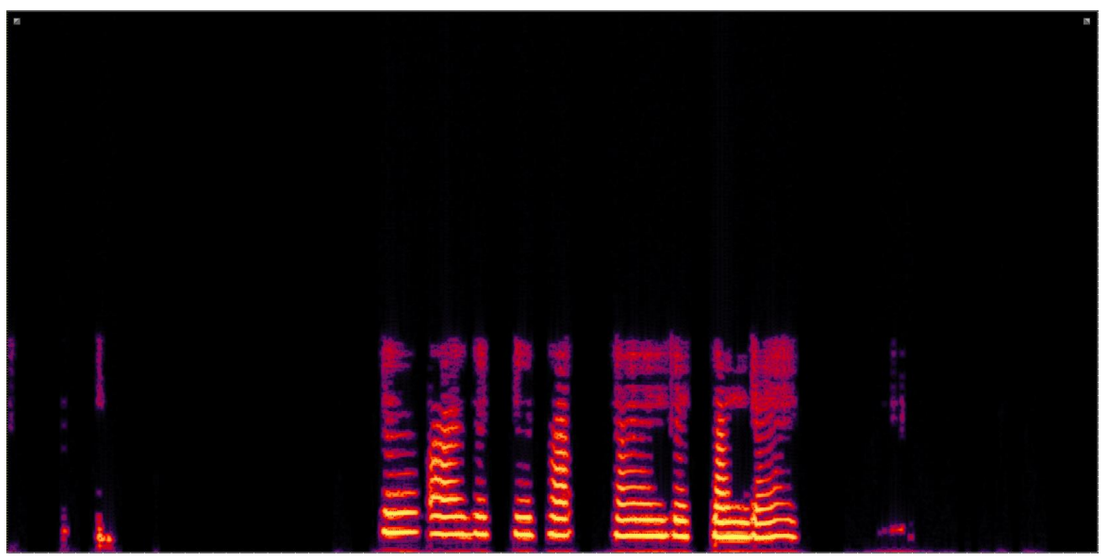
Case Study



Noisy



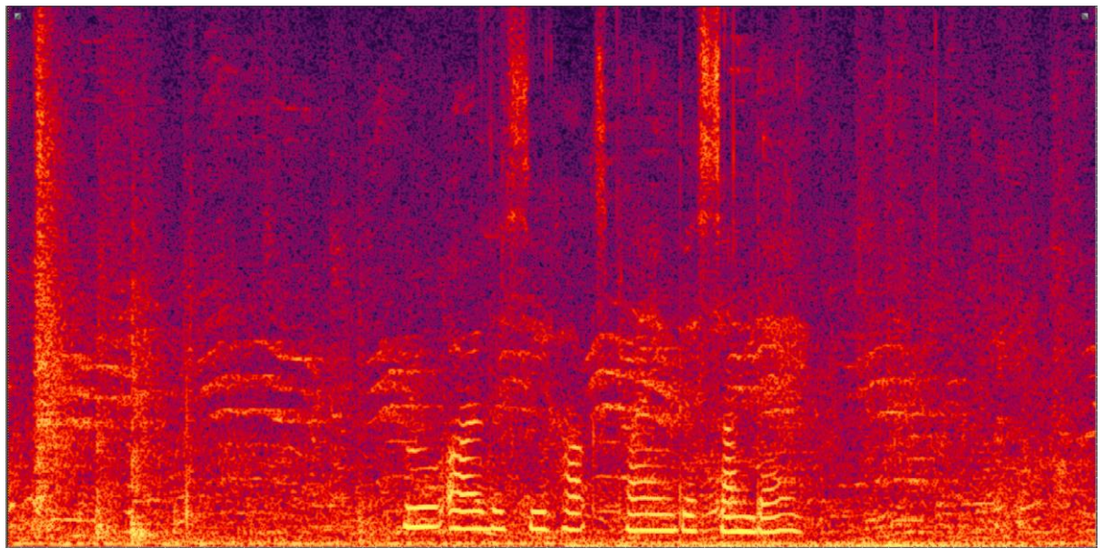
Clean



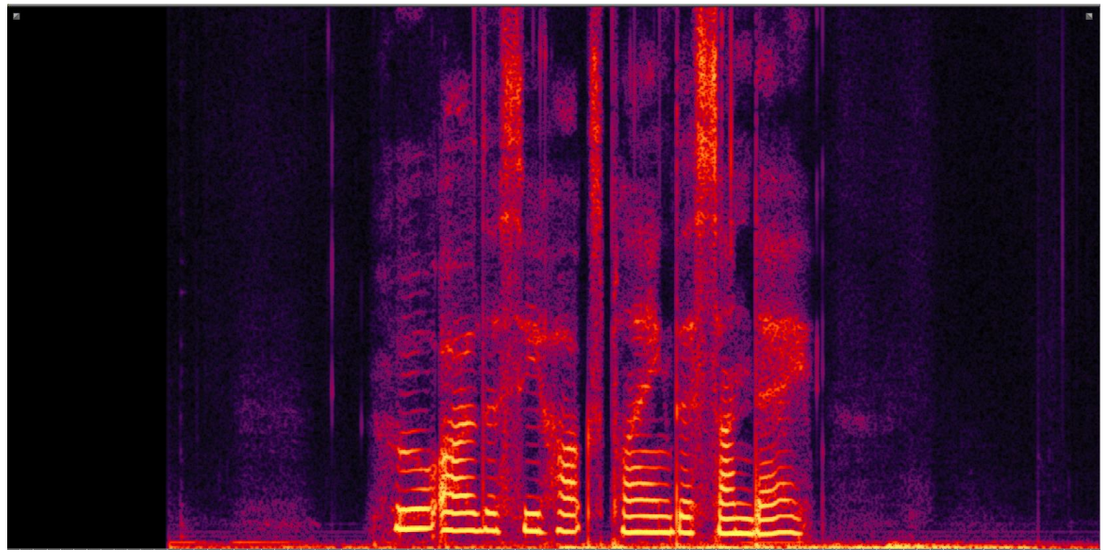
DNN

Experiments

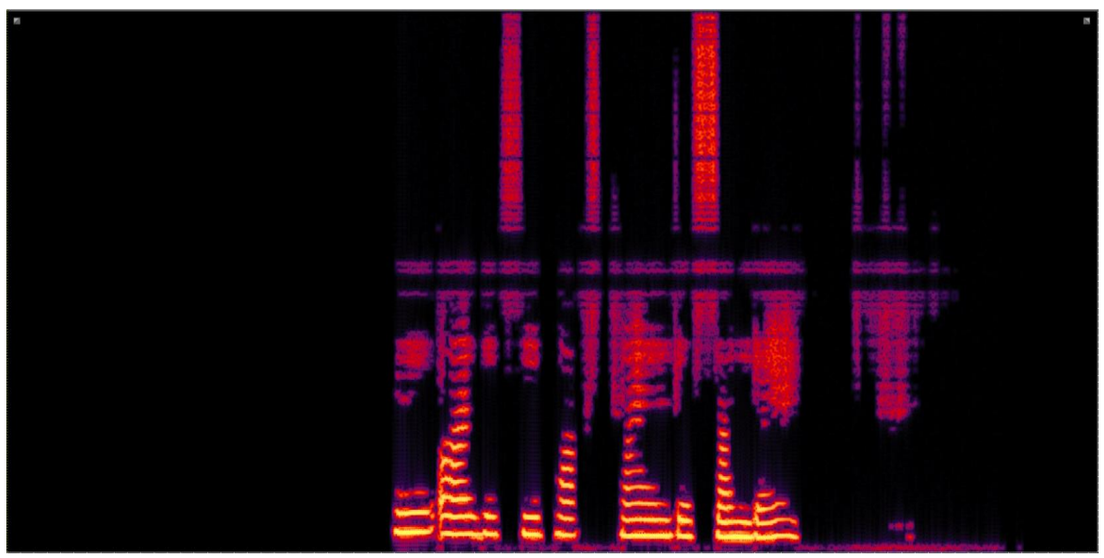
Case Study



Noisy



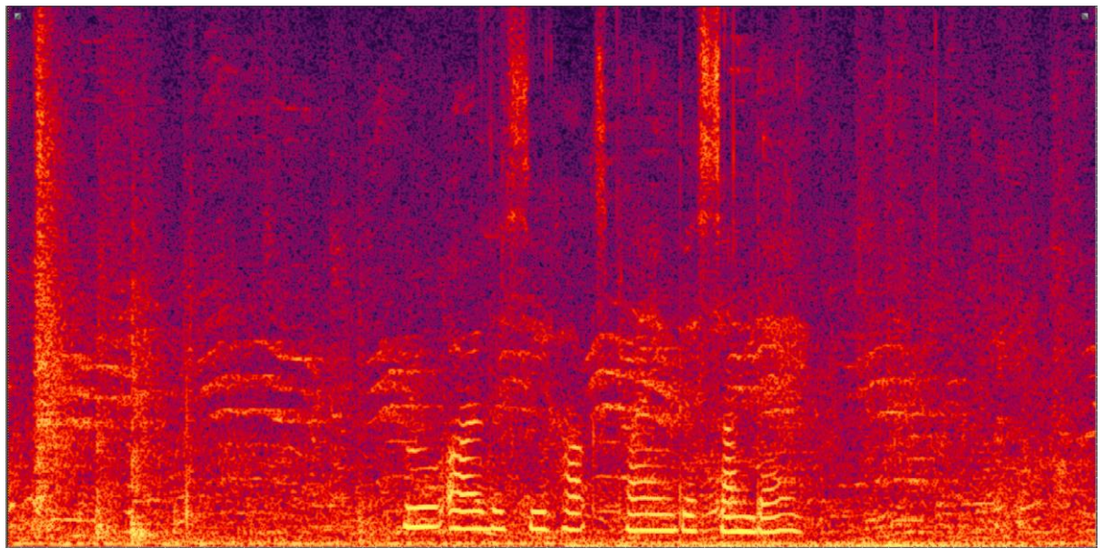
Clean



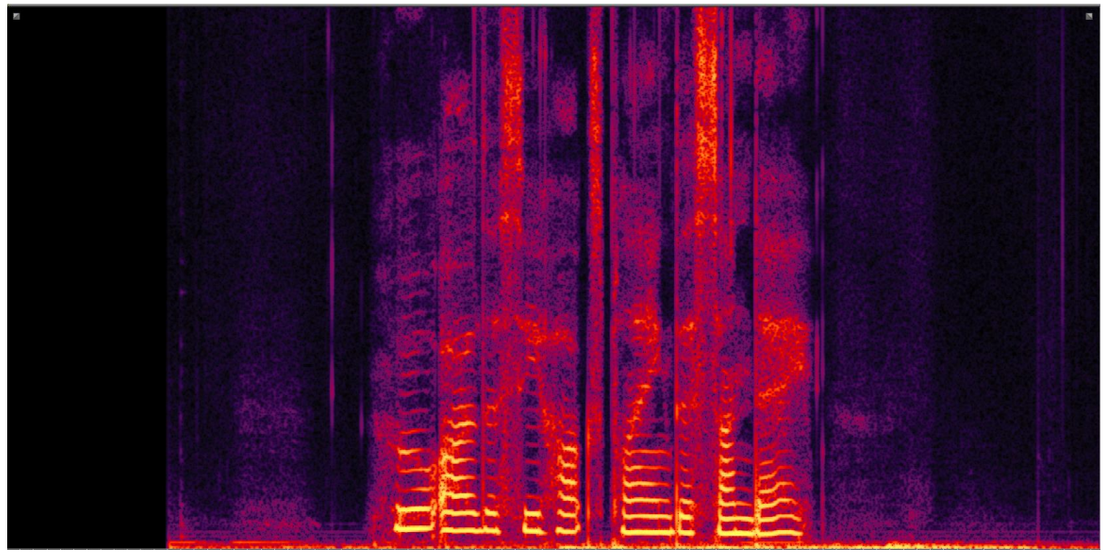
RNN

Experiments

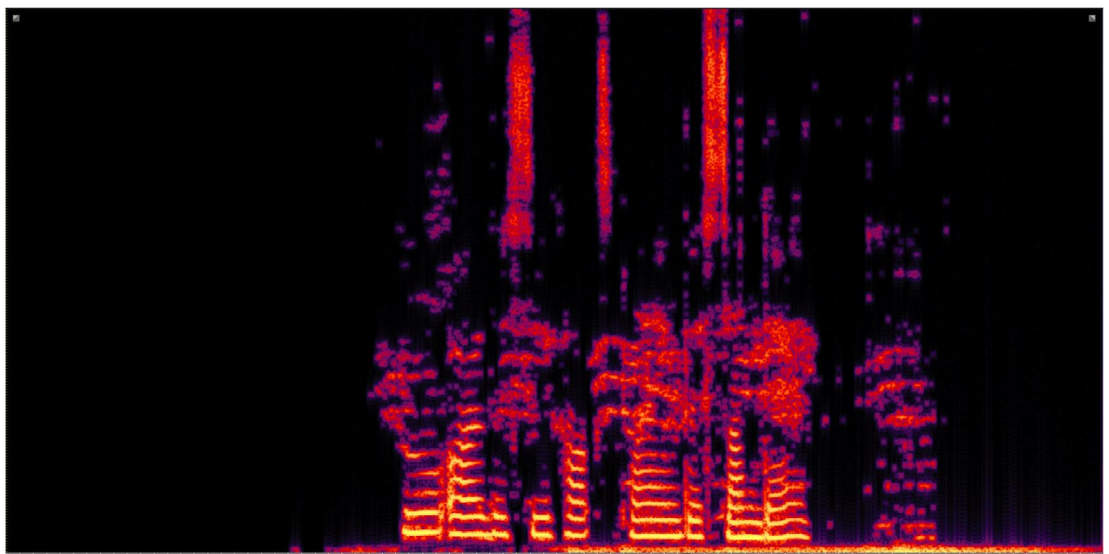
Case Study



Noisy



Clean



Ours

Conclusion

- Convolutions help capture local pattern
- Recurrence helps model sequential structure
- Our model improves SNR by 35 dB and PESQ by 0.6
- With fixed ASR system, improves WER by 1%
- Good generalizations on unseen noise

Conclusion

Thanks