Denoising with Generative Models

Loïs Bilat
MA3 IN - Fall 2019 - Semester Project - 12 ECTS
Supervisor : Brian Sifriger

Problem + Why is it important ?

- Record high quality audio => perfect conditions necessary
- Can we do this in software ? => implement in e.g. smartphones

If successful, can be applied to other types of signals (radar, lidar, radio, ...)

Precise problem statement

- Input : low quality audio (noise, low resolution, reverberation, ...)
- Output : better audio (less noise, higher resolution, ..)

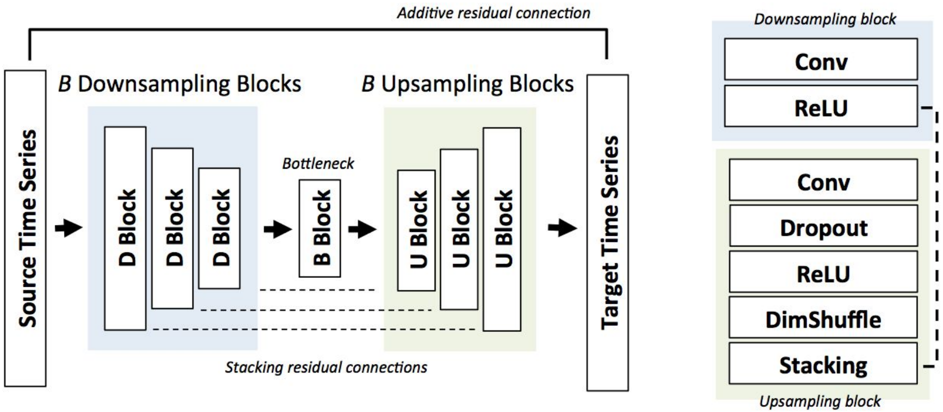
Sounds better => success !

Previous work

Kuleshov, Enam, Ermon : **Audio Super Resolution with Neural Networks**, 2017

- Audio super-resolution applied on speech
- One of the only paper that also tried with music (piano)
- Convolutional neural network that works directly on the audio signal
- Bottleneck architecture

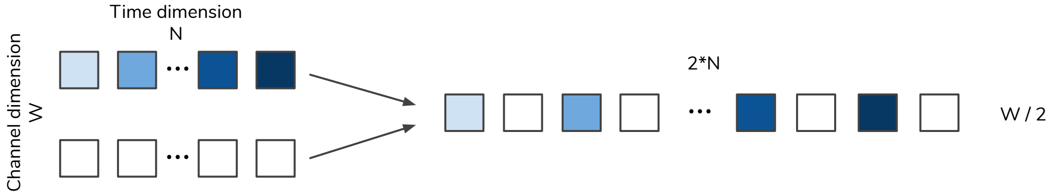
One previous work



Bottleneck architecture

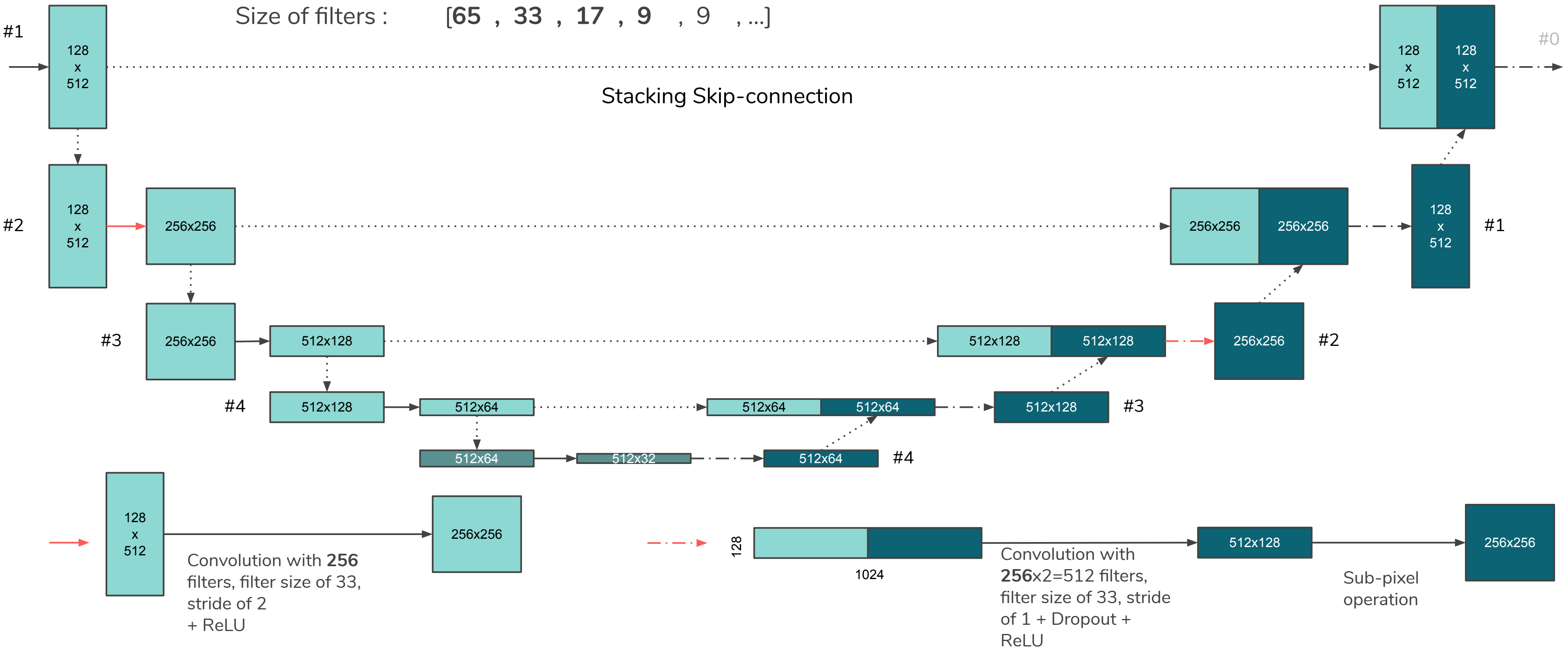
- N Downsampling blocks
- N Upsampling blocks
- Skip connections (stacking and additive)

DimShuffle : Sub-pixel operation



Stacking skip-connection

Number of filters : [**128, 256, 512, 512**, 512, ...]
Size of filters : [**65 , 33 , 17 , 9** , 9 , ...]

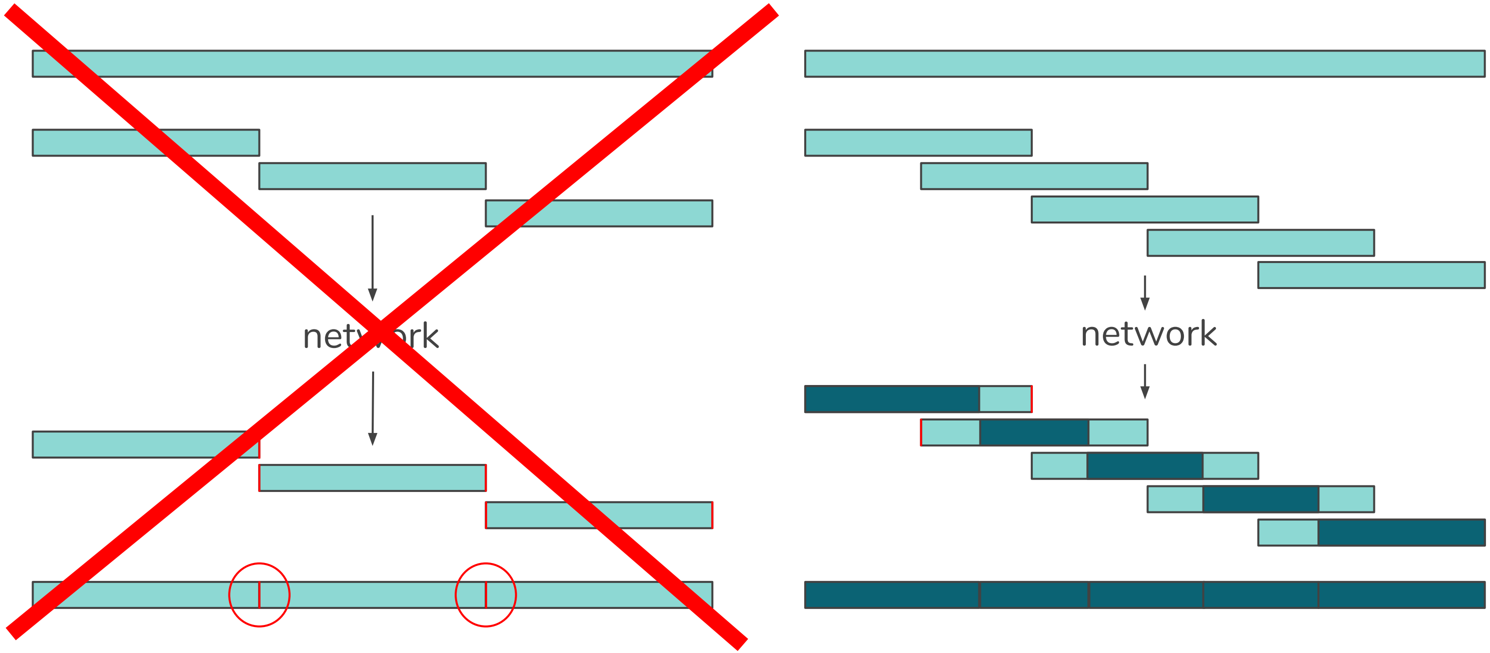


Proposed method

- Implement the network proposed by *Kuleshov*, *Enam*, *Ermo*
- Once it is working as expected, try to improve it using GANs
    - *Sung Kim*, *Visvesh Sathe*, **Adversarial Audio Super-Resolution with Unsupervised Feature Losses**
- Not only for super-resolution, but also for denoising and reverberation removal

Re-constructing the audio

We want it to work for any audio file



Experiment

Datasets

Maestro dataset :	122GB	44.1kHz	1411 kbps	wav
Beethoven dataset :	330MB	44.1kHz	112 kbps	ogg

Metrics

- SNR (Signal to noise ratio), higher = better
- LSD (Log-spectral distance), lower = better
- MOS (Mean opinion score)

$$\mathrm{SNR}\left(x, x_{ref}\right) = 10 \log_{10} \frac{\|x_{ref}\|_2^2}{\|x - x_{ref}\|_2^2}$$

$$\mathrm{LSD}\left(X, X_{ref}\right) = \frac{1}{W} \sum_{w=1}^{W} \sqrt{\frac{1}{K} \sum_{k=1}^{N} \left(\log_{10} \frac{|X(w,k)|^2}{|X_{ref}(w,k)|^2}\right)^2}$$

Work done so far

- Implementation using pytorch
- Complete pipeline
  - Data preparation
  - Training
  - Generate improved audio file
  - Evaluation with metrics
- Modular code
  - Any dataset
  - Any preprocessing on the audio
- Trained on noisy, downsampled and reverberated audio

Results Super-resolution

5kHz to 10kHz, 10 epochs, mini-batch of 32 samples, sliding window of 1024 with stride 512, depth 4, dropout of 0.5. Metrics on 1’000’000 points.

	SNR		LSD	
	Low quality	Improved	Low quality	Improved
Default	27.42	1.65	2.32	**1.39**
4 -> 8 layers	27.42	1.65	2.32	***1.34***
Dropout 0.5 -> 0.2	27.42	1.65	2.32	**1.41**

SNR worse, but LSD better
When listening : hear saturation, but also hear frequencies back

Results Denoising and Dereverberation

	SNR		LSD	
	Low quality	Improved	Low quality	Improved
Denoising	6.02	3.55	1.05	1.22
Dereverberation	7.26	3.08	0.41	0.90

Always worse, probably need more training data / deeper network

Planned work

Using inspiration from other papers (**Adversarial Audio Super-Resolution with Unsupervised Feature Losses**)

- Turn into a GAN
- Add loss computed by features of another network
- Try more realistic pre-processing

References

- *Volodymyr Kuleshov, S. Zayd Enam, and Stefano Ermon*, **Audio Super-resolution using neural networks**, arXiv:1708.00853, 2017
- *Sung Kim, Visvesh Sathe*, **Adversarial Audio Super-Resolution with Unsupervised Feature Losses**, ICLR 2019 Conference Blind Submission, https://openreview.net/forum?id=H1eH4n09KX
- *François G. Germain, Qifeng Chen, and Vladlen Koltun*, **Speech Denoising with Deep Feature Losses**, arXiv:1806.10522, 2018
- *Santiago Pascual , Antonio Bonafonte , Joan Serra*, **SEGAN: Speech Enhancement Generative Adversarial Network**, arXiv:1703.09452, 2017
- *Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck*. "**Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset**." In International Conference on Learning Representations, 2019, [link]