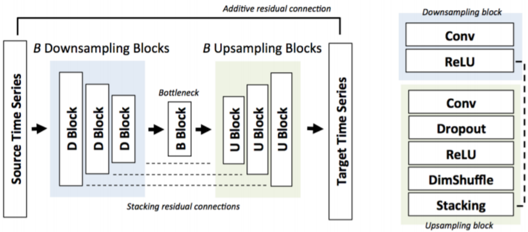
AUDIO SUPER-RESOLUTION USING NEURAL NETS

Volodymyr Kuleshov, S. Zayd Enam, and Stefano Ermon

Architecture



Audio processing

We represent an audio signal as a function s(t): $[0, T] \rightarrow R$, where T is the duration of the signal (in seconds) and $s(t)$ is the amplitude at t. Taking a digital measurement of s requires us to discretize the continuous function $s(t)$ into a vector $x(t)$ $:\{\frac{1}{R}, \frac{2}{R}, \cdot, \frac{RT}{R}\}$. In this work R as the sampling rate of x (in Hz).Goal is to increase the resolution of audio samples by predicting x from a fraction of its samples taken at $\{\frac{1}{R}, \frac{2}{R}, \cdot, \frac{RT}{R}\}$

Method

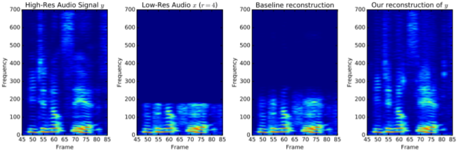


Рис.: Audio super-resolution visualized using spectrograms.

$x = \{\frac{x_1}{R_1}, \ldots \frac{x_{R_1 T_1}}{R_1}\}$ - low resolution signal, $y = \{\frac{y_1}{R_1}, \ldots, \frac{y_{R_2 T_2}}{R_2}\}$ - high-resolution version of x that has a sampling rate $R_2 > R_1$. We use $r = R2/R1$ to denote the upsampling ratio of the two signals.We learn a model $p(y|x)$.

$y = f_\theta(x) + \epsilon$, $\epsilon \sim \mathcal{N}(0, 1)$ is Gaussian noise and f_θ is a model parametrized by θ.The above formulation naturally leads to a mean squared error (MSE) objective:

$$\mathcal{L}(\mathcal{D}) = \frac{1}{n}\sqrt{\sum_{i=1}^{n} ||y_i - f_\theta(x_i)||_2^2}$$

Experiments

- **Dataset**. We use the VCTK dataset (Yamagishi) — which contains 44 hours of data from 108 different speakers — and the Piano dataset.We generate low-resolution audio signal from the 16 KHz originals by applying an order 8 Chebyshev type I low-pass filter before subsampling the signal by the desired scaling ratio.
- **Evaluating modes**. We evaluate our method in three regimes. The SINGLESPEAKER task trains the model on the first 223 recordings of VCTK Speaker 1 (about 30 mins) and tests on the last 8 recordings.The MULTISPEAKER task assesses our ability to generalize to new speakers. We train on the first 99 VCTK speakers and test on the 8 remaining ones. Lastly, the PIANO task extends audio-super resolution to non-vocal data.

Experiments

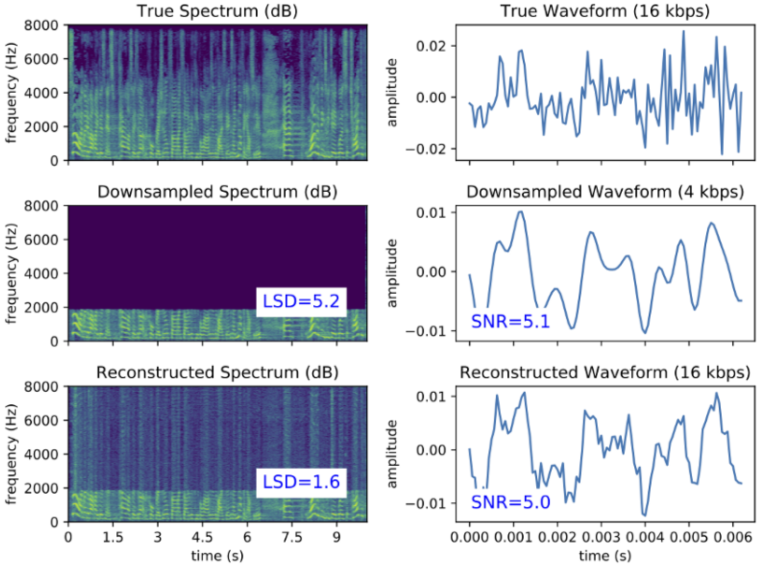
- **Method** We compare our method relative to two baselines: a cubic B-spline — which corresponds to the bicubic upsampling baseline used in image super-resolution — and the recent neural network-based technique.
- **Metrics**

$$SNR(x, y) = 10 log \frac{||y||_2^2}{||x - y||_2^2}$$

$$LSD(x, y) = \frac{1}{L} \sum_{l=1}^{L} \sqrt{\frac{1}{K} \sum_{k=1}^{K} (X(l, k) - \hat{X}(l, k))^2}$$

where X and $\hat{X}$ are the log-spectral power magnitudes of y and x, respectively. $X = log|S|^2$ where S is the short-time Fourier transform (STFT) of the signal, l and k index frames and frequencies, respectively;

About metrics



Results

		SingleSpeaker			MultiSpeaker			Piano		
Ratio	Obj.	Spline	DNN	Ours	Spline	DNN	Ours	Spline	DNN	Ours
$r = 2$	SNR	20.3	20.1	21.1	19.7	19.9	20.7	29.4	29.3	30.1
	LSD	4.5	3,7	3.2	4.4	3.6	3.1	3.5	3.4	3.4
$r = 4$	SNR	14.8	15.9	17.1	13.0	14.9	16.1	22.2	23.0	23.5
	LSD	8.2	4.9	3.6	8.0	5.8	3.5	5.8	5.2	3.6
$r = 6$	SNR	10.4	n/a	14.4	9.1	n/a	10.0	15.4	n/a	16.1
	LSD	10.3	n/a	3.4	10.1	n/a	3.7	7.3	n/a	4.4

Results. MUSHRA

MUSHRA - is a methodology for conducting a codec listening test to evaluate the perceived quality of the output from lossy audio compression algorithms.

MultiSpeaker Sample

	1	2	3	4	Average
Ours	69	75	64	37	61.3
DNN	51	55	66	53	56.3
Spline	31	25	38	47	35.3

Results. Domain adaptation

	LPF (Test)		No LPF (Test)	
	SNR	LSD	SNR	LSD
LPF (Train)	30.1	3.4	0.42	4.5
No LPF (Train)	0.43	4.4	33.2	3.3

Thank you for your attention!