This Matlab program implements a neural network based pitch tracker in noise described in:


The program was originally written by Kun Han, and the posted version was subsequently cleaned by Yuzhou Liu.

Please execute DNNPitch.m or RNNPitch.m in Matlab to get pitch estimates. For example:

1. DNN based pitch tracking:
   \[ \text{sig} = \text{audioread('examples/noisy0dB.wav')} \]
   \[ \text{F0} = \text{DNNPitch(sig)} \]

2. RNN based pitch tracking:
   \[ \text{sig} = \text{audioread('examples/noisy0dB.wav')} \]
   \[ \text{F0} = \text{RNNPitch(sig)} \]

Note that, a 16 kHz time domain signal is required for both functions. To run the RNN code, you have to access a GPU and the Parallel toolbox in Matlab.

Following is a description of the main steps in DNNPitch.m and RNNPitch.m:

1. Given the input signal, extract spectral domain features in each frame.
2. Employ neural networks (DNN/RNN) to compute the posterior probability of the pitch state for each frequency bin.
3. Use Viterbi decoding to connect pitch states, and use a moving average to generate final pitch contours.

To train the neural network models from scratch, we provide the following codes in ./train folder:

- gen_pefacfea.m: Generate features for a noisy sentence
- gen_praatfea.m: Generate desired outputs for a clean sentence
- trainNN: Given features and desired outputs for a corpus, train a DNN or RNN
Description of some other files and folders:

- feature_pefac.m: Feature extraction code
- cochleagram_toolbox: Signal processing toolbox
- DBNSuite: DNN toolbox
- examples: Example waveform files, including a noisy signal and a corresponding clean signal
- HMM: Viterbi decoding tool
- Models: Trained DNN and RNN models
- RNN: RNN toolbox
- utilities: Utility files
- voicebox_icl: Voicebox toolbox