Over-the-Air Speech Recognition Attack

UCLA EE209AS Winter 2018 by Professor. Mani srivastava
Midterm Report
Team members: Weikun Han, Zhengshuang Ren

Custom Training Data

To train on our own data, we should make sure that you have at least several hundred recordings of each sound you would like to recognize and arrange them into folders by class. For example, if we were trying to recognize a voice record after transmitting in the air, you would create a root folder called sample_1, and then within that many voice records with different noise for the same label. We would then organize our audio files into the appropriate folders.

Convolutional Neural Network with U-Net

Objective

To mimic the acoustic channel through Speaker - Air - Microphone to produce expected input audio files for effective adversary examples received at victim side.
Further combine the model with Moustafa's/Carlini's work to achieve Over-the-Air attack on home assistant devices.

Approach

Preprocessing

1. Process audio .wav files with librosa
    a. Possibly pair matching, offset removal or alignment
    b. Make input & target pairs
2. Trim and splice
    a. Trim all files into equal-length chunks of files
    b. Stitching sequentially/randomly into batches
3. Separation into Train, valid, test
    a. ⅔, ⅙, ⅙ or 60%, 20%, 20%

Training

1. Feed batches sequentially into neural network
2. Deep residual blocks with downsampling and upsampling using conv1D
3. Possibly Subpixel reshuffle/restack
4. Loss functions MSE/L2/BER
5. Save best model (best validation loss)

Testing

Use received audio files as input to produce anticipated input file to speaker

Reference Links:
Blog:
https://blog.insightdatascience.com/using-deep-learning-to-reconstruct-high-resolution-audio-29deee8b7ccd
U-Net:
https://arxiv.org/abs/1505.04597
Subpixel convolutions:
https://arxiv.org/abs/1609.05158
Github-Repo:
https://github.com/jhetherly/EnglishSpeechUpsampler

Requirements and Dependencies

The following packages are required (the version numbers that have been tested are given for reference):

- Python 2.7 or 3.6
- Tensorflow 1.0.1
- Numpy 1.12.1
- Librosa 0.5.0
- tqdm 4.11.2 (only for preprocessing training data)
- Sox 1.2.7 (only for preprocessing training data)