Project Course L13 Documentation

Daniele Ugo Leonzio, Simone Mariani

2020/2021

1 Introduction

This document is meant to be a guide to whoever wants to start working on our project. In the following sections we discuss in detail the dataset that we used in our experiments and the implementation of the scripts related to the application of the inpainting methods on audio spectrograms. We wrote three scripts, one for every used inpainting methods, because each one of them requires different pre-processing procedures. The entire project has been developed in Google Colaboratory enviroment using Python language. The machine learning frameworks adopted for the CNN based methods are Keras for the Pix2Pix method and PyTorch for the Deep Image Prior.

2 Dataset

The dataset we chose is the MOBIPHONE dataset, which is composed by different mobile phone audio recordings. Due to different durations among the various audio recordings, we decided to set a fixed length of 20 seconds for all the files. The dataset can be found at `https://drive.google.com/drive/folders/1vLAFl9oe20v4stUtOKLtHdzJO2t2oBAc?usp=sharing`.

3 Code

In this section we focus on the three scripts implementations. For each one of these scripts, the first step is to import the various libraries and to load the dataset.
Then the user must select the type of time-frequency transform and the type of reconstruction by keyboard input.
We mapped the spectrogram values into 0,255 range obtaining a 8-bit grayscale image. After that the parameters regarding the mask need to be set. In particular the user has to select the starting point for the region selected by the mask ('starting_point' variable) and the increment applied to that region at every iteration ('increment' variable). The last thing is to set the type of mask (acting on the 'mask' variable).

In every script the evaluation metrics are computed using the same functions. Also the function used to convert the inpainted spectrogram image back to audio ('convert' function) is replicated in all the scripts. In particular this convertion function simply computes the inverse short time fourier transform applying the reconstruction selected by the user.
In order to save the final results we write them into csv files through the csv library of Python that allows a user to read/write those files.

3.1 Model Based

The model based script follows the exact pipeline descripted above (and also inside our paper). The code runs the inpainting procedure on 24 files of 'Vodafone joy 845' folder which will be also used as test folder for the CNN based methods, just to compare the obtained results on the exact same recordings.

3.2 Pix2Pix

Before applying the inpainting pipeline, this method requires a training process. First of all we split the MOBIPHONE dataset in two parts, a test folder (the same as the other two methods) and a train folder (containing all the remaining files). These two datasets are populated with a couple of images for each audio recoding. This couple is composed by the original spectrogram image and the masked one. We train the network for 5 epochs ('EPOCHS' variable), to reconstruct a specific region that will be the same used for the testing files. In this case at each cycle step the mask is incremented as in the other methods, but now we need to train again the network on these new conditions. The generator and discriminator models composing the conditional GAN, and further details about the net implementation are defined according to `https://arxiv.org/pdf/1611.07004v3.pdf`. To extract the images from the testing and training datasets we use a specific function ('load' function) which uncouples the original and the masked spectrogram images.

3.3 Deep Image Prior

The inpainting procedure is the same descripted also for the other methods. The only things one could do is to vary the training parameters, such as: learning rate ('LR' variable), number of epochs ('num_iter' variable), standard deviation of noise ('reg_noise_std' variable). According to the Deep Image Prior implementation (which can be found at `https://github.com/DmitryUlyanov/deep-image-prior`) different net types can be used, in our case we chose the 'skip depth' 6 net. The other available types are: 'skip depth 4', 'skip depth 2', 'UNET' and 'ResNet'. The user can change net acting on 'NET_TYPE' variable.

4 GitHub

You can find the complete project including paper, scripts, results and useful links at our GitHub repository `https://github.com/mindofsimon/PC_L13_Audio_Inpainting`.