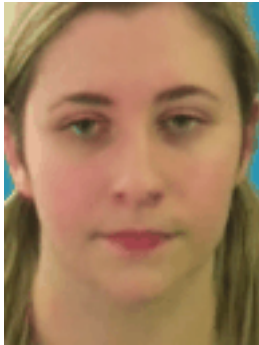

Speech-Driven Animation

This library implements the end-to-end facial synthesis model described in this paper.

This library is maintained by Konstantinos Vougioukas, Honglie Chen and Pingchuan Ma.



Downloading the models

The models were hosted on git LFS. However the demand was so high that I reached the quota for free gitLFS storage. I have moved the models to GoogleDrive. Models can be found [here](#). Place the model file(s) under `sda/data/`

Installing

To install the library do:

```
1 $ pip install .
```

Running the example

To create the animations you will need to instantiate the VideoAnimator class. Then you provide an image and audio clip (or the paths to the files) and a video will be produced.

Choosing the model

The model has been trained on the GRID, TCD-TIMIT, CREMA-D and LRW datasets. The default model is GRID. To load another pretrained model simply instantiate the VideoAnimator with the following arguments:

```
1 import sda
2 va = sda.VideoAnimator(gpu=0, model_path="crema") # Instantiate the
  animator
```

The models that are currently uploaded are: - [x] GRID - [x] TIMIT - [x] CREMA - [] LRW

Example with image and audio paths

```
1 import sda
2 va = sda.VideoAnimator(gpu=0) # Instantiate the animator
3 vid, aud = va("example/image.bmp", "example/audio.wav")
```

Example with numpy arrays

```
1 import sda
2 import scipy.io.wavfile as wav
3 from PIL import Image
4
5 va = sda.VideoAnimator(gpu=0) # Instantiate the animator
6 fs, audio_clip = wav.read("example/audio.wav")
7 still_frame = Image.open("example/image.bmp")
8 vid, aud = va(frame, audio_clip, fs=fs)
```

Saving video with audio

```
1 va.save_video(vid, aud, "generated.mp4")
```

Using the encodings

The encoders for audio and video are made available so that they can be used to produce features for classification tasks.

Audio encoder

The Audio encoder (which is made of Audio-frame encoder and RNN) is provided along with a dictionary which has information such as the feature length (in seconds) required by the Audio Frame encoder and the overlap between audio frames.

```
1 import sda
2 encoder, info = sda.get_audio_feature_extractor(gpu=0)
```

Citation

If you find this code useful in your research, please consider to cite the following papers:

```
1 @inproceedings{vougioukas2019end,
2   title={End-to-End Speech-Driven Realistic Facial Animation with
3     Temporal GANs.},
4   author={Vougioukas, Konstantinos and Petridis, Stavros and Pantic,
5     Maja},
6   booktitle={CVPR Workshops},
7   pages={37--40},
8   year={2019}
9 }
```