MakeItTalk README

队员 & 分工

- **陈顺章 1120212450**

  负责代码框架的编写，`main_end2end.py` 的测试与运行，接口的重写

- **李沅臻 1120210631**

  负责本地模型的训练以及 `train_content.py` 的运行

- **张司睿 1120211750**

  负责 `main_train_image_translation.py` 的运行以及代码框架的阅读，README 文档编写

- **夏诗航 1120212365**

  负责 `main_end2end_cartoon.py` 的运行

- **黎书萌 1120212093**

  负责 `main_gen_new_puppet` 的运行

项目简介

MakeItTalk 是一个由马萨诸塞大学阿默斯特分校、Adobe 研究院等机构提出的新方法。这种方法不仅能让真人头像说话，还可以让卡通、油画、素描、日漫中的人像说话。与之前的方法不同，MakeItTalk **将输入音频信号中的内容和说话人身份信息分离开来**。**音频内容用于稳健地控制嘴唇及周围区域的运动，而说话人信息则决定了面部表情的细节和人物的头部动态**。
其具体工作流程包括：给定一段音频和一张面部图像，MakeItTalk 可以生成说话人的头部状态动画，且声画同步。在训练阶段，研究者使用现成可用的人脸特征点检测器对输入图像进行预处理，提取面部特征点。然后使用输入音频和提取到的特征点直接训练使语音内容动态化的基线模型。为了达到高保真动态效果，研究者尝试将输入音频信号的语音内容和说话人嵌入分离开来，进而实现面部特征点的预测，其特征点-图像合成算法分为两种：对于非真人图像，如油画或矢量图，该研究使用基于德劳内三角剖分的简单换脸方法。对于真人图像，则使用图像转换网络将真人面部图像和底层特征点预测动态化。

项目网址链接：https://people.umass.edu/~yangzhou/MakeItTalk/

原论文链接：https://people.umass.edu/~yangzhou/MakeItTalk/MakeItTalk_SIGGRAPH_Asia_Final_round-5.pdf

论文仓库链接：GitHub - yzhou359/MakeItTalk

运行环境

环境要求集成在Makeittalk_requirement.txt中，通过下列操作安装所需环境：

```
pip install -r Makeittalk_requirement.txt
```

安装 `ffmpeg`

```
sudo apt-get install ffmpeg
```

运行前准备

- 安装运行环境
- 从 https://drive.google.com/drive/folders/1EwuAy3j1b9Zc1MsidUfxG_pJGc_cV60O?usp=sharing 下载 .pickle 文件到 `\checkpoints\dump` 文件夹
- 下载剩余三个需要的模型到 `\checkpoints\ckpt` 文件夹中

  https://drive.google.com/file/d/1i2LJXKp-yWKIEEgJ7C6cE3_2NirfY_0a/view?usp=sharing

  https://drive.google.com/file/d/1rV0jkyDqPW-aDJcj7xSO6Zt1zSXqn1mu/view?usp=sharing

  https://drive.google.com/file/d/1ZiwPp_h62LtjU0DwpelLUoodKPR85K7x/view?usp=sharing

是否还有别的运行前准备需要做?

项目功能

在我们的复现中，实现了一键训练、生成视频与评估：

执行指令：

```
python run_talkingface.py --model=audio2landmark_content --dataset=audio2landmark_content
```

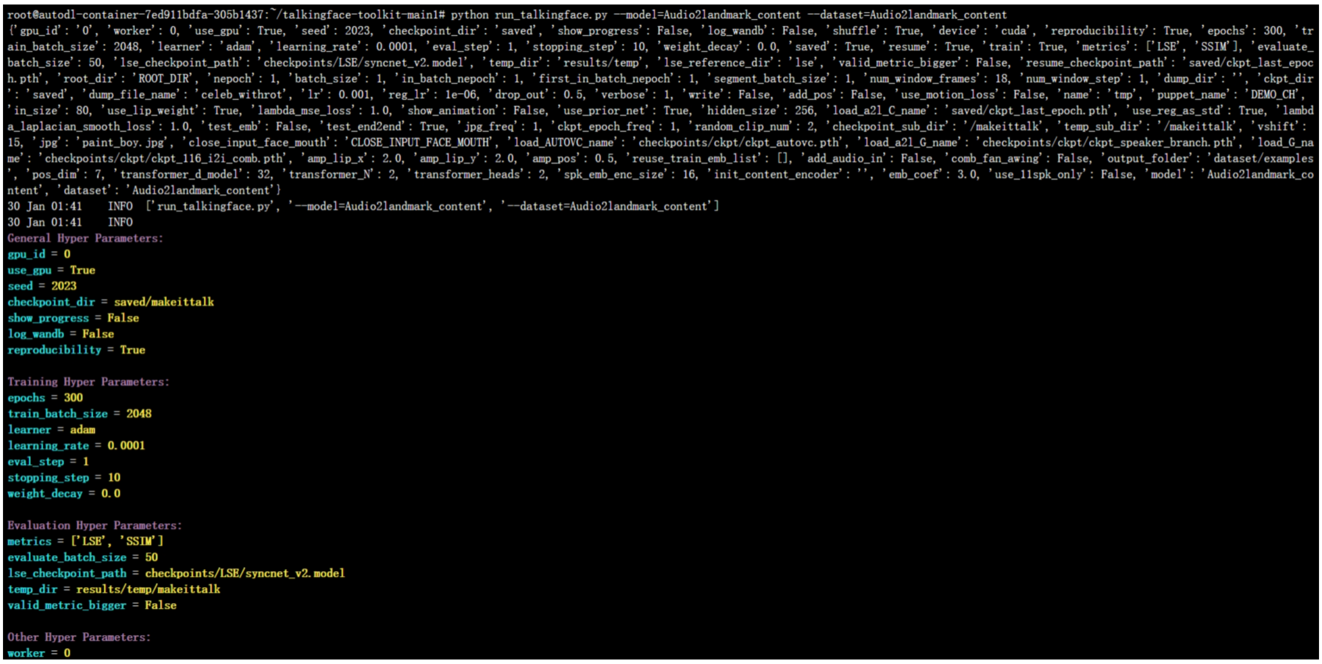
训练后的**视频**将自动存储在 `dataset/examples` 里面

训练结束后的**模型**将存储在 `saved/` 里面

实验结果截图

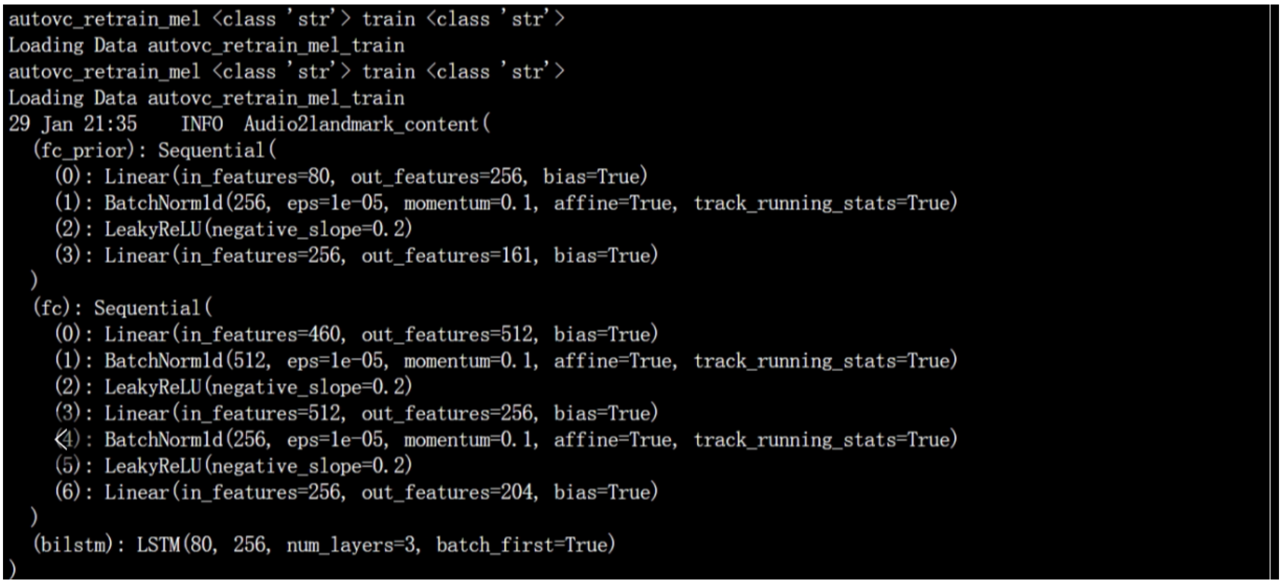
该部分是提供了在我们的服务器上跑通整个实验部分的截图：

- 输入命令后开始读取 yaml 中的内容：



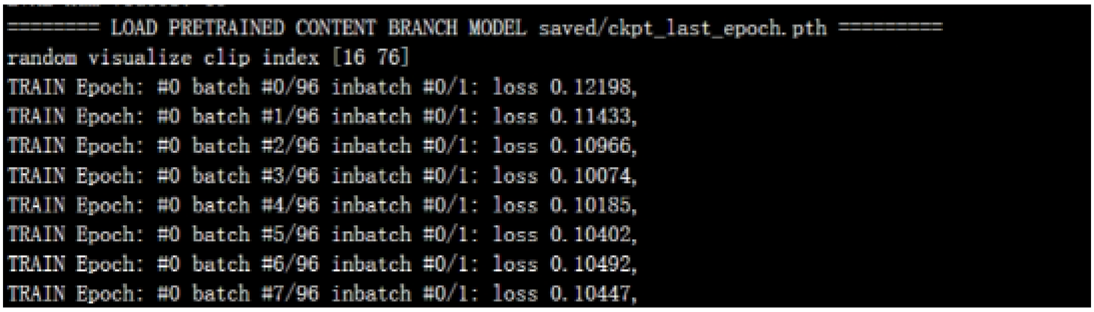
```
root@autodl-container-7ed911bdfa-305b1437:~/talkingface-toolkit-main1# python run_talkingface.py --model=Audio2landmark_content --dataset=Audio2landmark_content
{'gpu_id': '0', 'worker': 0, 'use_gpu': True, 'seed': 2023, 'checkpoint_dir': 'saved', 'show_progress': False, 'log_wandb': False, 'shuffle': True, 'device': 'cuda', 'reproducibility': True, 'epochs': 300, 'tr
ain_batch_size': 2048, 'learner': 'adam', 'learning_rate': 0.0001, 'eval_step': 1, 'stopping_step': 10, 'weight_decay': 0.0, 'saved': True, 'resume': True, 'train': True, 'metrics': ['LSE', 'SSIM'], 'evaluate_
batch_size': 50, 'lse_checkpoint_path': 'checkpoints/LSE/syncnet_v2.model', 'temp_dir': 'results/temp', 'lse_reference_dir': 'lse', 'valid_metric_bigger': False, 'resume_checkpoint_path': 'saved/ckpt_last_epoc
h.pth', 'root_dir': 'ROOT_DIR', 'nepoch': 1, 'batch_size': 1, 'in_batch_nepoch': 1, 'first_in_batch_nepoch': 1, 'segment_batch_size': 1, 'num_window_frames': 18, 'num_window_step': 1, 'dump_dir': '', 'ckpt_dir
': 'saved', 'dump_file_name': 'celeb_withrot', 'lr': 0.001, 'reg_lr': 1e-06, 'drop_out': 0.5, 'verbose': 1, 'write': False, 'add_pos': False, 'use_motion_loss': False, 'name': 'tmp', 'puppet_name': 'DEMO_CH',
'in_size': 80, 'use_lip_weight': True, 'lambda_mse_loss': 1.0, 'show_animation': False, 'use_prior_net': True, 'hidden_size': 256, 'load_a2l_C_name': 'saved/ckpt_last_epoch.pth', 'use_reg_as_std': True, 'lambd
a_laplacian_smooth_loss': 1.0, 'test_emb': False, 'test_end2end': True, 'jpg_freq': 1, 'ckpt_epoch_freq': 1, 'random_clip_num': 2, 'checkpoint_sub_dir': '/makeittalk', 'temp_sub_dir': '/makeittalk', 'vshift':
15, 'jpg': 'paint_boy.jpg', 'close_input_face_mouth': 'CLOSE_INPUT_FACE_MOUTH', 'load_AUTOVC_name': 'checkpoints/ckpt/ckpt_autovc.pth', 'load_a2l_G_name': 'checkpoints/ckpt/ckpt_speaker_branch.pth', 'load_G_na
me': 'checkpoints/ckpt/ckpt_116_i2i_comb.pth', 'amp_lip_x': 2.0, 'amp_lip_y': 2.0, 'amp_pos': 0.5, 'reuse_train_emb_list': [], 'add_audio_in': False, 'comb_fan_awing': False, 'output_folder': 'dataset/examples
', 'pos_dim': 7, 'transformer_d_model': 32, 'transformer_N': 2, 'transformer_heads': 2, 'spk_emb_enc_size': 16, 'init_content_encoder': '', 'emb_coef': 3.0, 'use_11spk_only': False, 'model': 'Audio2landmark_co
ntent', 'dataset': 'Audio2landmark_content'}
30 Jan 01:41    INFO  ['run_talkingface.py', '--model=Audio2landmark_content', '--dataset=Audio2landmark_content']
30 Jan 01:41    INFO
General Hyper Parameters:
gpu_id = 0
use_gpu = True
seed = 2023
checkpoint_dir = saved/makeittalk
show_progress = False
log_wandb = False
reproducibility = True

Training Hyper Parameters:
epochs = 300
train_batch_size = 2048
learner = adam
learning_rate = 0.0001
eval_step = 1
stopping_step = 10
weight_decay = 0.0

Evaluation Hyper Parameters:
metrics = ['LSE', 'SSIM']
evaluate_batch_size = 50
lse_checkpoint_path = checkpoints/LSE/syncnet_v2.model
temp_dir = results/temp/makeittalk
valid_metric_bigger = False

Other Hyper Parameters:
worker = 0
```

- 输出模型信息



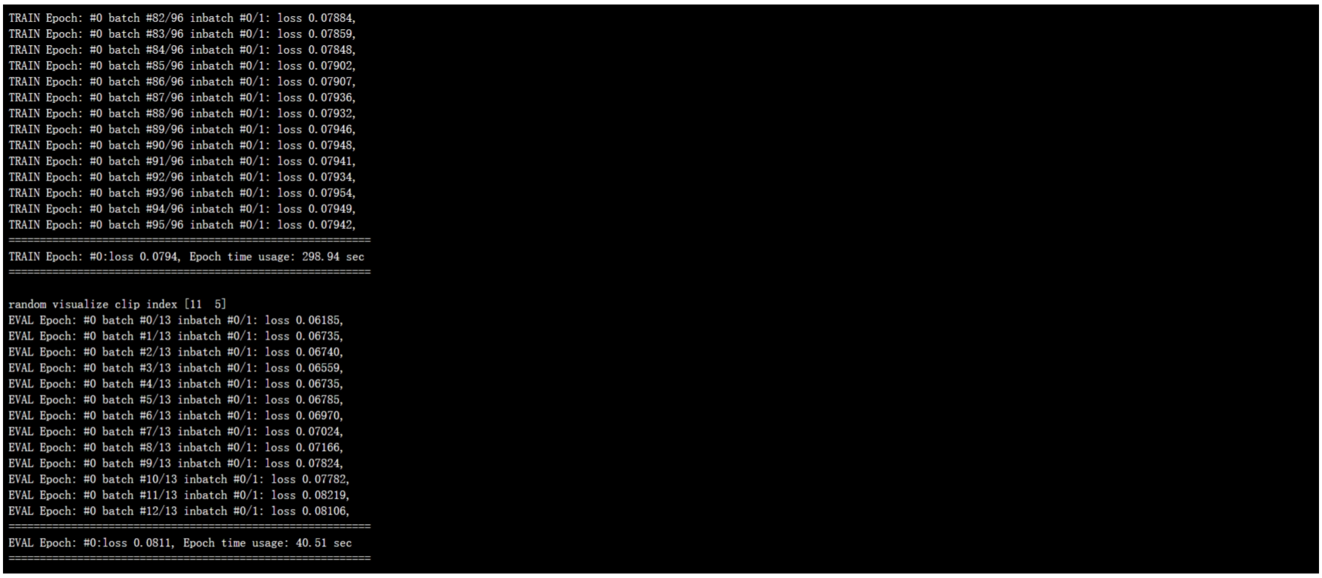
```
autovc_retrain_mel <class 'str'> train <class 'str'>
Loading Data autovc_retrain_mel_train
autovc_retrain_mel <class 'str'> train <class 'str'>
Loading Data autovc_retrain_mel_train
29 Jan 21:35    INFO  Audio2landmark_content(
  (fc_prior): Sequential(
    (0): Linear(in_features=80, out_features=256, bias=True)
    (1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): LeakyReLU(negative_slope=0.2)
    (3): Linear(in_features=256, out_features=161, bias=True)
  )
  (fc): Sequential(
    (0): Linear(in_features=460, out_features=512, bias=True)
    (1): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): LeakyReLU(negative_slope=0.2)
    (3): Linear(in_features=512, out_features=256, bias=True)
    (4): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): LeakyReLU(negative_slope=0.2)
    (6): Linear(in_features=256, out_features=204, bias=True)
  )
  (bilstm): LSTM(80, 256, num_layers=3, batch_first=True)
)
```

- 完成前置步骤并开始训练（为了节省测试时间，训练的 epoch 数设为 1）：



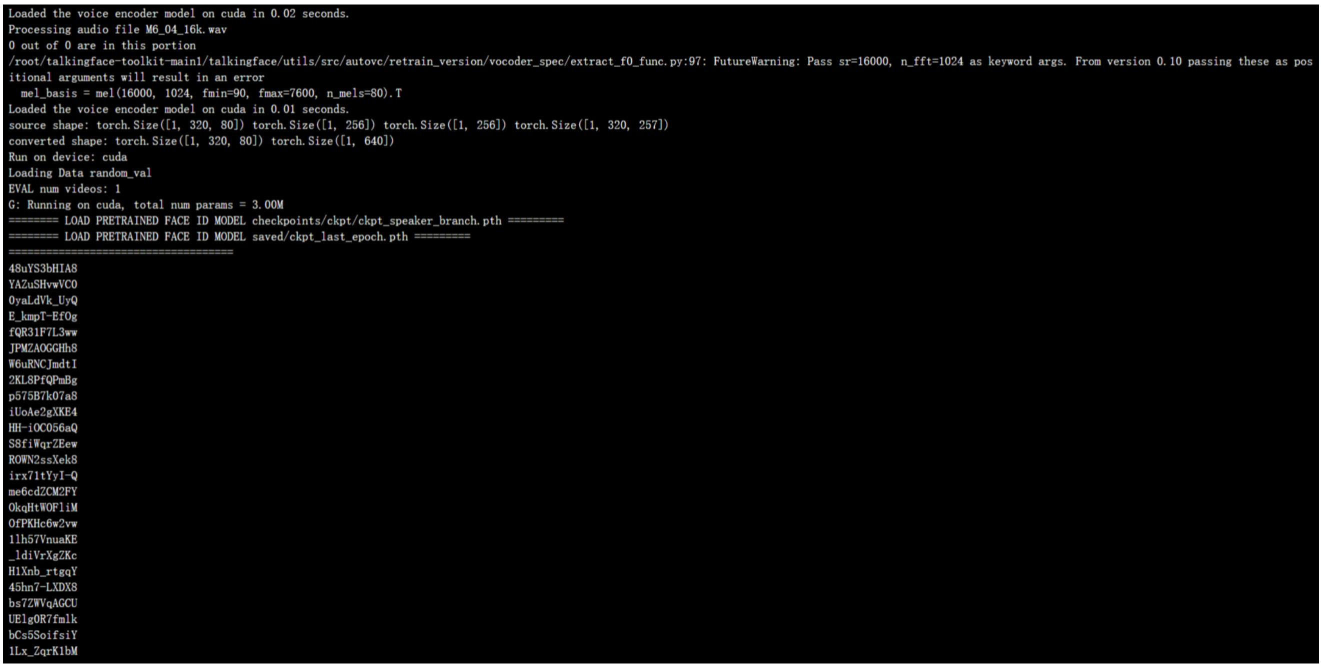
```
======== LOAD PRETRAINED CONTENT BRANCH MODEL saved/ckpt_last_epoch.pth =========
random visualize clip index [16 76]
TRAIN Epoch: #0 batch #0/96 inbatch #0/1: loss 0.12198,
TRAIN Epoch: #0 batch #1/96 inbatch #0/1: loss 0.11433,
TRAIN Epoch: #0 batch #2/96 inbatch #0/1: loss 0.10966,
TRAIN Epoch: #0 batch #3/96 inbatch #0/1: loss 0.10074,
TRAIN Epoch: #0 batch #4/96 inbatch #0/1: loss 0.10185,
TRAIN Epoch: #0 batch #5/96 inbatch #0/1: loss 0.10402,
TRAIN Epoch: #0 batch #6/96 inbatch #0/1: loss 0.10492,
TRAIN Epoch: #0 batch #7/96 inbatch #0/1: loss 0.10447,
```

- 训练完成，自动评估是否过拟合：

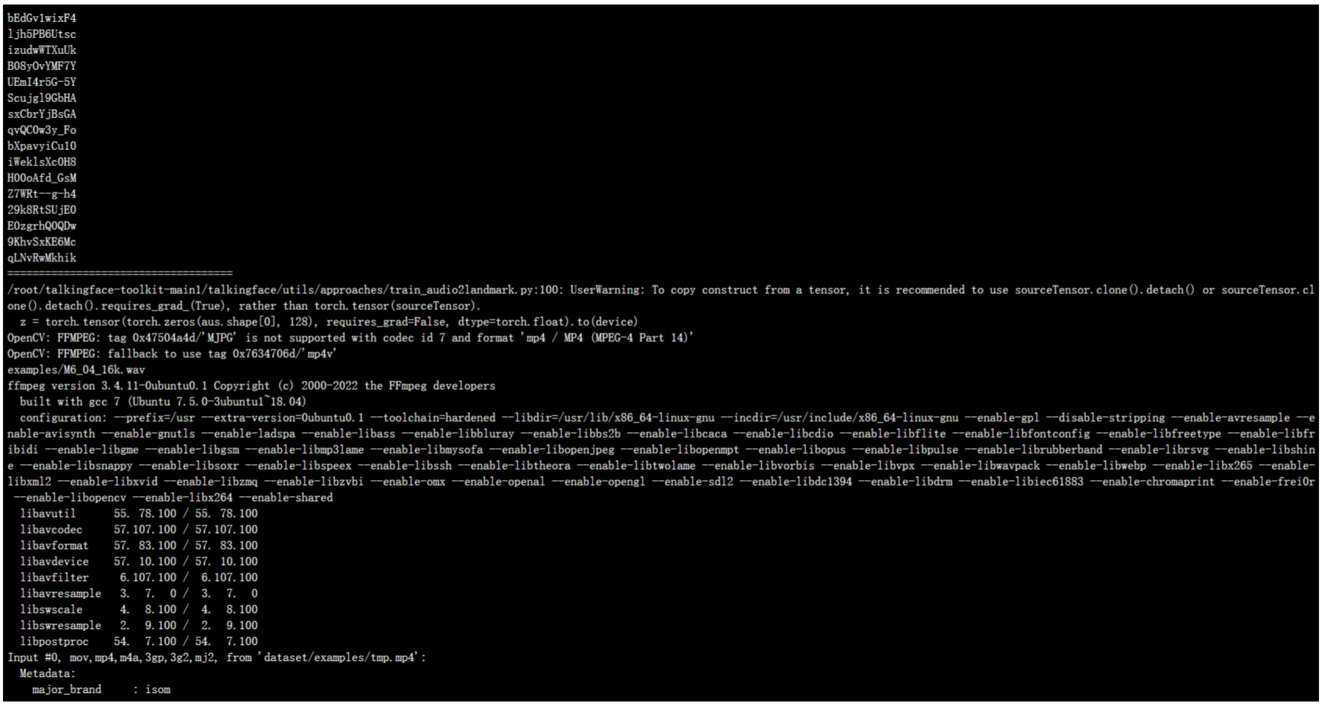


```
TRAIN Epoch: #0 batch #82/96 inbatch #0/1: loss 0.07884,
TRAIN Epoch: #0 batch #83/96 inbatch #0/1: loss 0.07859,
TRAIN Epoch: #0 batch #84/96 inbatch #0/1: loss 0.07848,
TRAIN Epoch: #0 batch #85/96 inbatch #0/1: loss 0.07902,
TRAIN Epoch: #0 batch #86/96 inbatch #0/1: loss 0.07907,
TRAIN Epoch: #0 batch #87/96 inbatch #0/1: loss 0.07936,
TRAIN Epoch: #0 batch #88/96 inbatch #0/1: loss 0.07932,
TRAIN Epoch: #0 batch #89/96 inbatch #0/1: loss 0.07946,
TRAIN Epoch: #0 batch #90/96 inbatch #0/1: loss 0.07948,
TRAIN Epoch: #0 batch #91/96 inbatch #0/1: loss 0.07941,
TRAIN Epoch: #0 batch #92/96 inbatch #0/1: loss 0.07934,
TRAIN Epoch: #0 batch #93/96 inbatch #0/1: loss 0.07954,
TRAIN Epoch: #0 batch #94/96 inbatch #0/1: loss 0.07949,
TRAIN Epoch: #0 batch #95/96 inbatch #0/1: loss 0.07942,
==========================================================
TRAIN Epoch: #0:loss 0.0794, Epoch time usage: 298.94 sec
==========================================================


random visualize clip index [11  5]
EVAL Epoch: #0 batch #0/13 inbatch #0/1: loss 0.06185,
EVAL Epoch: #0 batch #1/13 inbatch #0/1: loss 0.06735,
EVAL Epoch: #0 batch #2/13 inbatch #0/1: loss 0.06740,
EVAL Epoch: #0 batch #3/13 inbatch #0/1: loss 0.06559,
EVAL Epoch: #0 batch #4/13 inbatch #0/1: loss 0.06735,
EVAL Epoch: #0 batch #5/13 inbatch #0/1: loss 0.06785,
EVAL Epoch: #0 batch #6/13 inbatch #0/1: loss 0.06970,
EVAL Epoch: #0 batch #7/13 inbatch #0/1: loss 0.07024,
EVAL Epoch: #0 batch #8/13 inbatch #0/1: loss 0.07166,
EVAL Epoch: #0 batch #9/13 inbatch #0/1: loss 0.07824,
EVAL Epoch: #0 batch #10/13 inbatch #0/1: loss 0.07782,
EVAL Epoch: #0 batch #11/13 inbatch #0/1: loss 0.08219,
EVAL Epoch: #0 batch #12/13 inbatch #0/1: loss 0.08106,
==========================================================
EVAL Epoch: #0:loss 0.0811, Epoch time usage: 40.51 sec
==========================================================
```

- 训练结束后自动生成关键点与用于评估的视频

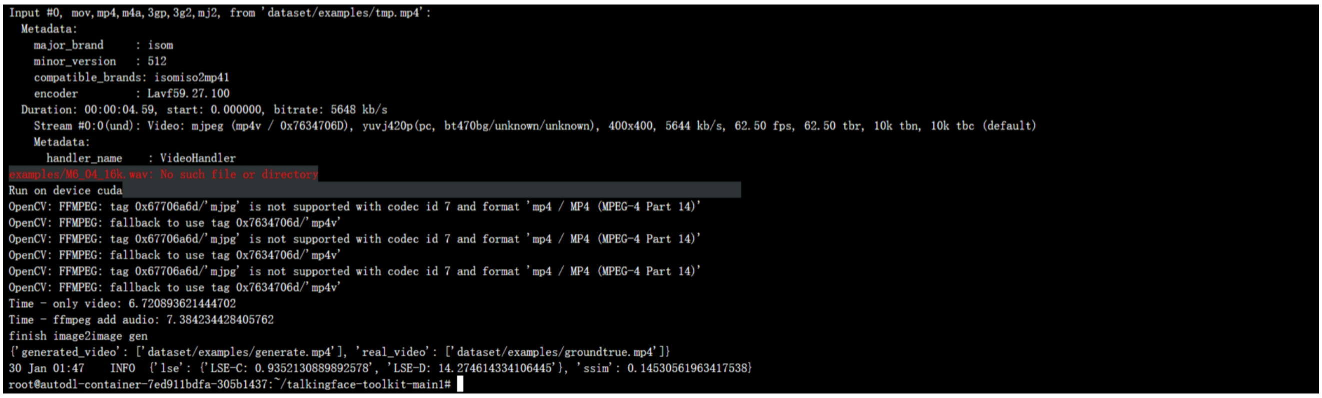


```
Loaded the voice encoder model on cuda in 0.02 seconds.
Processing audio file M6_04_16k.wav
0 out of 0 are in this portion
/root/talkingface-toolkit-main1/talkingface/utils/src/autovc/retrain_version/vocoder_spec/extract_f0_func.py:97: FutureWarning: Pass sr=16000, n_fft=1024 as keyword args. From version 0.10 passing these as pos
itional arguments will result in an error
  mel_basis = mel(16000, 1024, fmin=90, fmax=7600, n_mels=80).T
Loaded the voice encoder model on cuda in 0.01 seconds.
source shape: torch.Size([1, 320, 80]) torch.Size([1, 256]) torch.Size([1, 256]) torch.Size([1, 320, 257])
converted shape: torch.Size([1, 320, 80]) torch.Size([1, 640])
Run on device: cuda
Loading Data random_val
EVAL num videos: 1
G: Running on cuda, total num params = 3.00M
======== LOAD PRETRAINED FACE ID MODEL checkpoints/ckpt/ckpt_speaker_branch.pth =========
======== LOAD PRETRAINED FACE ID MODEL saved/ckpt_last_epoch.pth =========
====================================
48uYS3bHIA8
YAZuSHvwVC0
0yaLdVk_UyQ
E_kmpT-Ef0g
fQR31F7L3ww
JPMZAOGGHh8
W6uRNCJmdtI
2KL8PfQPmBg
p575B7k07a8
iUoAe2gXKE4
HH-iOC056aQ
S8fiWqrZEew
ROWN2ssXek8
irx71tYyI-Q
me6cdZCM2FY
OkqHtWOFliM
OfPKHc6w2vw
1lh57VnuaKE
_ldiVrXgZKc
H1Xnb_rtgqY
45hn7-LXDX8
bs7ZWVqAGCU
UElg0R7fmlk
bCs5SoifsiY
1Lx_ZqrK1bM
```



```
bEdGv1wixF4
ljh5PB6Utsc
izudwWTXuUk
B08yOvYMF7Y
UEmI4r5G-5Y
Scujgl9GbHA
sxCbrYjBsGA
qvQCOw3y_Fo
bXpavyiCu10
iWeklsXcOH8
H0OoAfd_GsM
Z7WRt--g-h4
29k8RtSUjEO
EOzgrhQOQDw
9KhvSxKE6Mc
qLNvRwMkhik
====================================
/root/talkingface-toolkit-main1/talkingface/utils/approaches/train_audio2landmark.py:100: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.cl
one().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
  z = torch.tensor(torch.zeros(aus.shape[0], 128), requires_grad=False, dtype=torch.float).to(device)
OpenCV: FFMPEG: tag 0x47504a4d/'MJPG' is not supported with codec id 7 and format 'mp4 / MP4 (MPEG-4 Part 14)'
OpenCV: FFMPEG: fallback to use tag 0x7634706d/'mp4v'
examples/M6_04_16k.wav
ffmpeg version 3.4.11-0ubuntu0.1 Copyright (c) 2000-2022 the FFmpeg developers
  built with gcc 7 (Ubuntu 7.5.0-3ubuntu1~18.04)
  configuration: --prefix=/usr --extra-version=0ubuntu0.1 --toolchain=hardened --libdir=/usr/lib/x86_64-linux-gnu --incdir=/usr/include/x86_64-linux-gnu --enable-gpl --disable-stripping --enable-avresample --e
nable-avisynth --enable-gnutls --enable-ladspa --enable-libass --enable-libbluray --enable-libbs2b --enable-libcaca --enable-libcdio --enable-libflite --enable-libfontconfig --enable-libfreetype --enable-libfr
ibidi --enable-libgme --enable-libgsm --enable-libmp3lame --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --enable-libpulse --enable-librubberband --enable-librsvg --enable-libshin
e --enable-libsnappy --enable-libsoxr --enable-libspeex --enable-libssh --enable-libtheora --enable-libtwolame --enable-libvorbis --enable-libvpx --enable-libwavpack --enable-libwebp --enable-libx265 --enable-
libxml2 --enable-libxvid --enable-libzmq --enable-libzvbi --enable-omx --enable-openal --enable-opengl --enable-sdl2 --enable-libdc1394 --enable-libdrm --enable-libiec61883 --enable-chromaprint --enable-frei0r
 --enable-libopencv --enable-libx264 --enable-shared
  libavutil      55. 78.100 / 55. 78.100
  libavcodec     57.107.100 / 57.107.100
  libavformat    57. 83.100 / 57. 83.100
  libavdevice    57. 10.100 / 57. 10.100
  libavfilter     6.107.100 /  6.107.100
  libavresample   3.  7.  0 /  3.  7.  0
  libswscale      4.  8.100 /  4.  8.100
  libswresample   2.  9.100 /  2.  9.100
  libpostproc    54.  7.100 / 54.  7.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'dataset/examples/tmp.mp4':
  Metadata:
    major_brand     : isom
```

- 使用框架中的 LSE 方法进行比对与评估，评估结果如下：

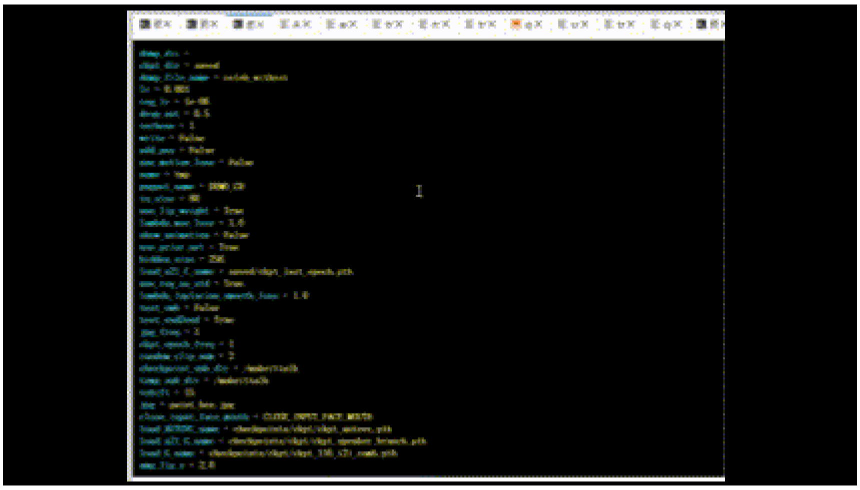


```
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'dataset/examples/tmp.mp4':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2mp41
    encoder         : Lavf59.27.100
  Duration: 00:00:04.59, start: 0.000000, bitrate: 5648 kb/s
    Stream #0:0(und): Video: mjpeg (mp4v / 0x7634706D), yuvj420p(pc, bt470bg/unknown/unknown), 400x400, 5644 kb/s, 62.50 fps, 62.50 tbr, 10k tbn, 10k tbc (default)
    Metadata:
      handler_name    : VideoHandler
examples/M6_04_16k.wav: No such file or directory
Run on device cuda
OpenCV: FFMPEG: tag 0x67706a6d/'mjpg' is not supported with codec id 7 and format 'mp4 / MP4 (MPEG-4 Part 14)'
OpenCV: FFMPEG: fallback to use tag 0x7634706d/'mp4v'
OpenCV: FFMPEG: tag 0x67706a6d/'mjpg' is not supported with codec id 7 and format 'mp4 / MP4 (MPEG-4 Part 14)'
OpenCV: FFMPEG: fallback to use tag 0x7634706d/'mp4v'
OpenCV: FFMPEG: tag 0x67706a6d/'mjpg' is not supported with codec id 7 and format 'mp4 / MP4 (MPEG-4 Part 14)'
OpenCV: FFMPEG: fallback to use tag 0x7634706d/'mp4v'
Time - only video: 6.720893621444702
Time - ffmpeg add audio: 7.384234428405762
finish image2image gen
{'generated_video': ['dataset/examples/generate.mp4'], 'real_video': ['dataset/examples/groundtrue.mp4']}
30 Jan 01:47    INFO  {'lse': {'LSE-C: 0.9352130889892578', 'LSE-D: 14.274614334106445'}, 'ssim': 0.14530561963417538}
root@autodl-container-7ed911bdfa-305b1437:~/talkingface-toolkit-main1#
```

- 评估结果：

```
30 Jan 01:47 INFO {'lse': {'LSE-C: 0.9352130889892578', 'LSE-D: 14.274614334106445'},'ssim': 0.14530561963417538 }
```

- 完整运行流程



备注

- 由于这个是一个视频生成框架，通过关键点将图片转换成视频，不存在真实视频，我们的对比视频是作者训练1001轮的模型之后处理出来的视频
- 作者的代码仓库中只提供了 `train_content` 部分的训练，其余的训练作者并没有提供训练方法以及数据集，因此只将模型 `ckpt_content_branch` 替换成了我们自己的，剩下的均用了作者预训练好的模型