More Detailed steps for data preprocessing

(How to get face and lip in this work?)

Face region

Lip region

Some notes

- First of all, We sincerely appreciate Dr. Qi Chen (The University of Adelaide), and Dr. Chenxu Hu (Tsinghua University). We are very very grateful to the original authors for providing us with their excellent datasets 🎉🎉🎉.
- Numerous methods exist for extracting regions of lips and faces; in this instance, we share the more detailed preprocessing steps of our methods (HPMDubbing 🎬) to obtain the mouth and lip region images.
- If you have more convenient and faster tools or methods (like Deep Visual Tracking or FaceTracking), you can also use them to process the original video.
- We will share the image areas of the mouth and face we extracted for readers to use conveniently. (If you want to extract features from other models, you can feed these images directly)
- Below we will take the chem dataset as an example to show you our processing steps in this work.

1. Download all list .txt

- Please note that we first removed five samples from the training set because we found the **Montreal Forced Aligner (MFA)** cannot obtain their TextGrid files for the five samples provided by the author, as follows:

```
Setting up corpus information...
Number of speakers in corpus: 1, average number of utterances per speaker: 6640.0
Creating dictionary information...
Setting up corpus_data directory...
Generating base features (mfcc)...
Calculating CMVN...
Done with setup.
There were 5 segments/files not aligned. Please see /data/conggaoxiang/Chem_wav16_feature/TextGrid/unaligned.txt for more details on why
 alignment failed for these files.
Done! Everything took 712.3856081962585 seconds
(v2c) conggaoxiang@aa-ESC8000-G4:~/MFA_tools$ vim /data/conggaoxiang/Chem_wav16_feature/TextGrid/unaligned.txt
```

conggaoxiang — co

```
PyIiTK0eXrA-003 Could not decode (beam too narrow)
X0nCoOX0LAg-022 Could not decode (beam too narrow)
ZwsWjelzqDA-099 Could not decode (beam too narrow)
hXLiubyd58Y-004 Could not decode (beam too narrow)
wFY9ZvUmM5o-039 Could not decode (beam too narrow)
```

all list number: 6635

All_list.txt

```
6633  myN3PqD38Ds-001
6634  myN3PqD38Ds-002
6635  myN3PqD38Ds-004
```

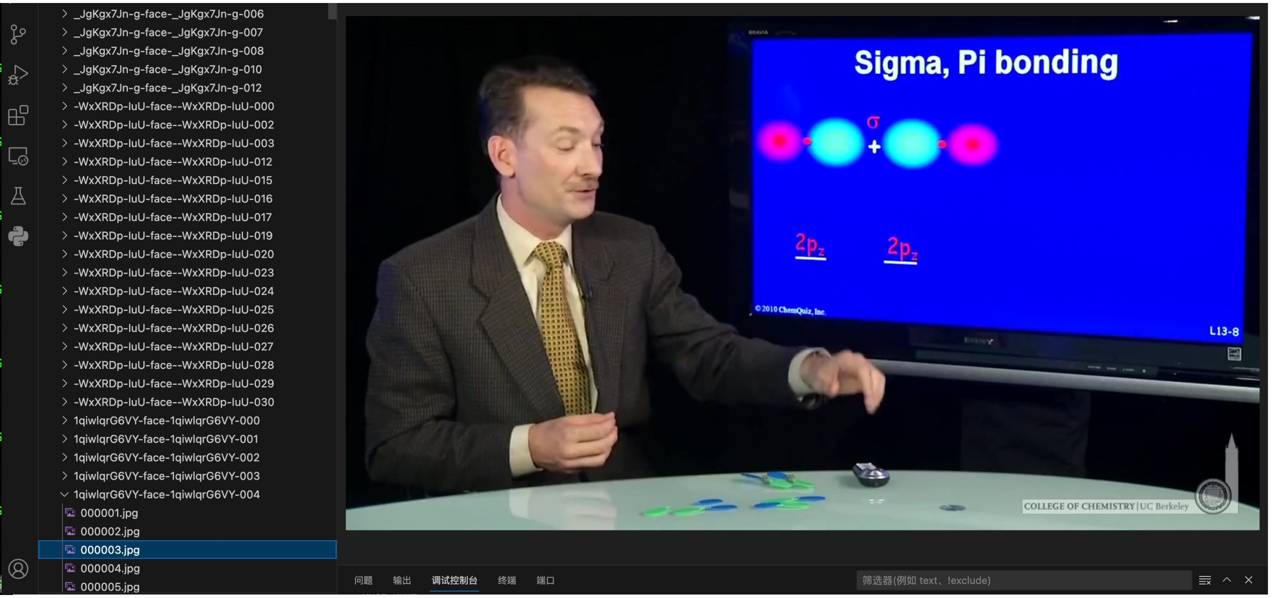
We share the all list .txt you can download it by Google Drive: https://drive.google.com/file/d/13xJW7rRKl4uE0qZhz0EgOvO_5VpbVzzG/view?usp=drive_link

2. Use FFmpeg to obtain each frame of the video, and set the FPS to 25

```python
os.makedirs(args.pyframesPath, exist_ok=True)  # Save all the video frames
command = ("ffmpeg -y -i %s -qscale:v 2 -threads %d -r %d -f image2 %s -loglevel panic" % \
        (args.videoPath, args.nDataLoaderThread, 25, os.path.join(args.pyframesPath, '%06d.jpg')))
subprocess.call(command, shell=True, stdout=None)
```

Note that set -r to 25

```
(v2c) conggaoxiang@aa-ESC8000-G4:~/TalkNet-ASD-main$ python demo_chem_last.py
  0%|                                                                    | 0/6635 [00:00<?, ?it/s]
============NUM:1---ID:myN3PqD38Ds-005---SUCCESS========
2023-12-01 16:11:59 Extract the video and save in /data/conggaoxiang/TalkNet-ASD-main/Chem_buwan/Chem_frame/myN3PqD38Ds-face-myN3PqD38Ds
-005
  0%|                                                          | 1/6635 [00:00<1:12:24,  1.53it/s]
============NUM:2---ID:myN3PqD38Ds-014---SUCCESS========
2023-12-01 16:12:00 Extract the video and save in /data/conggaoxiang/TalkNet-ASD-main/Chem_buwan/Chem_frame/myN3PqD38Ds-face-myN3PqD38Ds
-014
  0%|                                                          | 2/6635 [00:02<1:36:44,  1.14it/s]
============NUM:3---ID:myN3PqD38Ds-015---SUCCESS========
2023-12-01 16:12:01 Extract the video and save in /data/conggaoxiang/TalkNet-ASD-main/Chem_buwan/Chem_frame/myN3PqD38Ds-face-myN3PqD38Ds
-015
  0%|                                                          | 3/6635 [00:03<1:40:01,  1.11it/s]
============NUM:4---ID:myN3PqD38Ds-016---SUCCESS========
2023-12-01 16:12:02 Extract the video and save in /data/conggaoxiang/TalkNet-ASD-main/Chem_buwan/Chem_frame/myN3PqD38Ds-face-myN3PqD38Ds
-016
  0%|                                                          | 4/6635 [00:03<1:39:39,  1.11it/s]
============NUM:5---ID:myN3PqD38Ds-017---SUCCESS========
2023-12-01 16:12:03 Extract the video and save in /data/conggaoxiang/TalkNet-ASD-main/Chem_buwan/Chem_frame/myN3PqD38Ds-face-myN3PqD38Ds
-017
  0%|                                                          | 5/6635 [00:04<1:22:36,  1.34it/s]
============NUM:6---ID:myN3PqD38Ds-022---SUCCESS========
2023-12-01 16:12:04 Extract the video and save in /data/conggaoxiang/TalkNet-ASD-main/Chem_buwan/Chem_frame/myN3PqD38Ds-face-myN3PqD38Ds
-022
  0%|                                                          | 6/6635 [00:05<1:44:05,  1.06it/s]
============NUM:7---ID:myN3PqD38Ds-024---SUCCESS========
2023-12-01 16:12:04 Extract the video and save in /data/conggaoxiang/TalkNet-ASD-main/Chem_buwan/Chem_frame/myN3PqD38Ds-face-myN3PqD38Ds
-024
  0%|                                                          | 7/6635 [00:06<1:25:52,  1.29it/s]
============NUM:8---ID:myN3PqD38Ds-025---SUCCESS========
2023-12-01 16:12:05 Extract the video and save in /data/conggaoxiang/TalkNet-ASD-main/Chem_buwan/Chem_frame/myN3PqD38Ds-face-myN3PqD38Ds
```



Please run 1_get_your_frames.py

You will get

3. Detect and crop the face from the video frames using S^3FD face detection model.

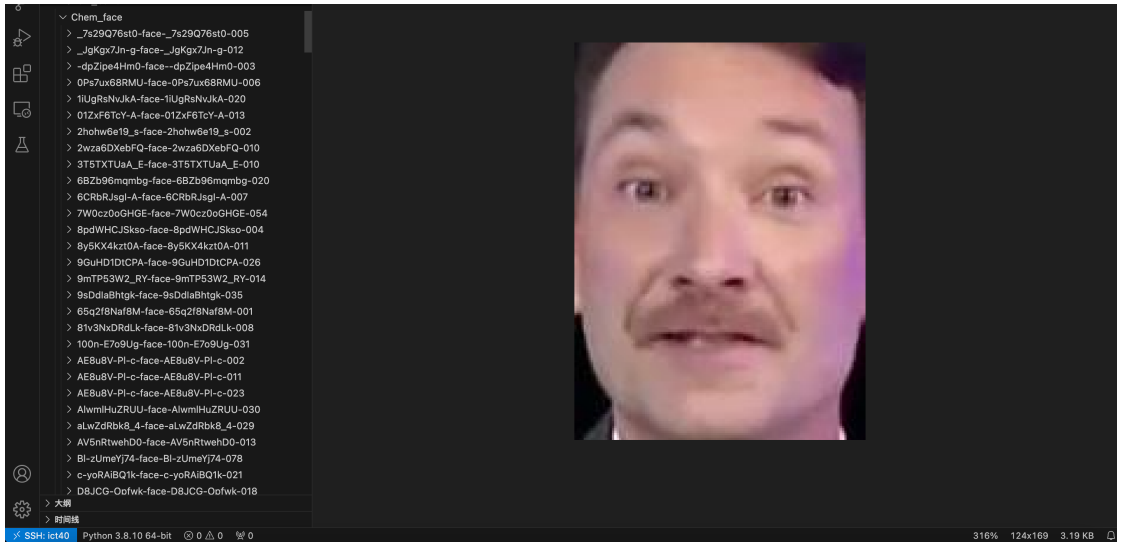
```python
class S3FD():
    def __init__(self, device='cuda'):
        tstamp = time.time()
        self.device = device
        self.net = S3FDNet(device=self.device).to(self.device)
        PATH = os.path.join(os.getcwd(), PATH_WEIGHT)
        state_dict = torch.load(PATH, map_location=self.device)
        self.net.load_state_dict(state_dict)
        self.net.eval()
    def detect_faces(self, image, conf_th=0.8, scales=[1]):
        w, h = image.shape[1], image.shape[0]
        bboxes = np.empty(shape=(0, 5))
        with torch.no_grad():
            for s in scales:
                scaled_img = cv2.resize(image, dsize=(0, 0), fx=

                scaled_img = np.swapaxes(scaled_img, 1, 2)
                scaled_img = np.swapaxes(scaled_img, 1, 0)
                scaled_img = scaled_img[[2, 1, 0], :, :]
                scaled_img = scaled_img.astype('float32')
                scaled_img -= img_mean
                scaled_img = scaled_img[[2, 1, 0], :, :]
                x = torch.from_numpy(scaled_img).unsqueeze(0).to
                y = self.net(x)
```

- This script is based on TalkNet-ASD : https://github.com/TaoRuijie/TalkNet-ASD
- For the face process, we use Parallel to accelerate.
- We also save unrecognized images as JSON files for checking.

Please run `2_get_your_face.py`

```
gaoxiang_cong@aa-ESC8000-G4:~/TalkNet-ASD-main$ python 2_get_your_face.py
[Parallel(n_jobs=4)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=4)]: Done  42 tasks      | elapsed:  1.1min
[Parallel(n_jobs=4)]: Done 192 tasks      | elapsed:  4.5min
[Parallel(n_jobs=4)]: Done 442 tasks      | elapsed:  9.7min
[Parallel(n_jobs=4)]: Done 792 tasks      | elapsed: 17.4min
[Parallel(n_jobs=4)]: Done 1242 tasks      | elapsed: 27.3min
```

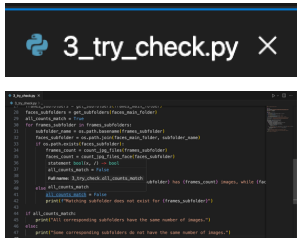
You will get



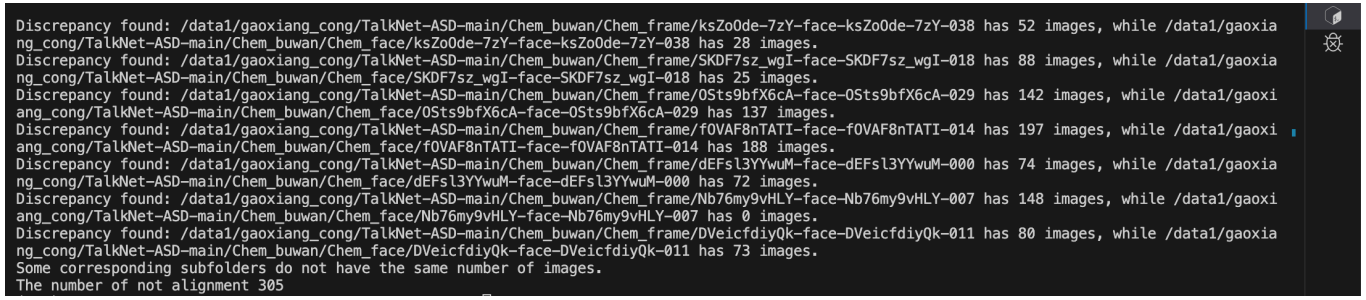
[Supplement] Why can't it be detected?

Script to check if faces are all aligned:

Please run



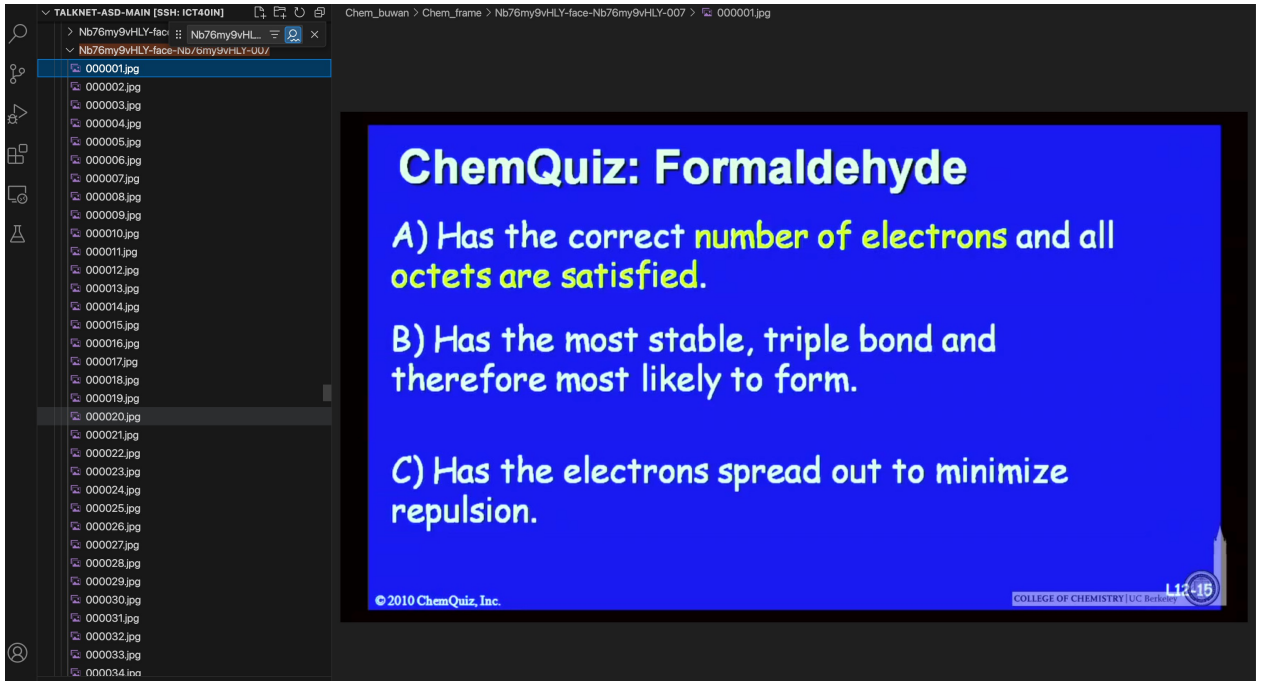
You will get

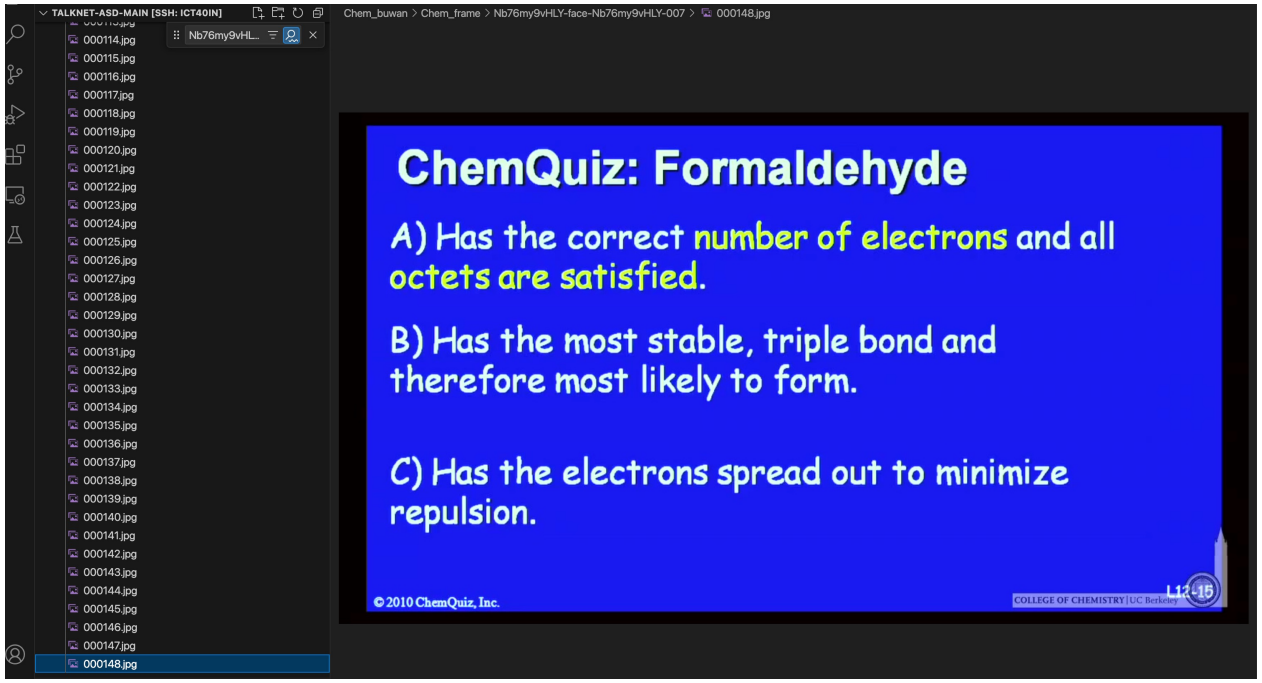


- We share the face regions extracted by ourselves. You can download it to check:
- https://drive.google.com/drive/folders/16Kl320RMZ3_ft1SiqpJyPmOeJFL20m7c?usp=drive_link

- Besides, we give some failed examples (PPT on the following two pages) for you to check.
- We provide our failed list, i.e. JSON file of the previously mentioned, which you can download here: https://drive.google.com/file/d/11TXnjkdVVdAPagXkSlg1uPrye5f3NEyS/view?usp=drive_link

For example, the sample **Nb76my9vHLY-face-Nb76my9vHLY-007** is all “Chemistry PPT”, and there is no “chemistry teacher”.





000001.jpg ——— 000148.jpg

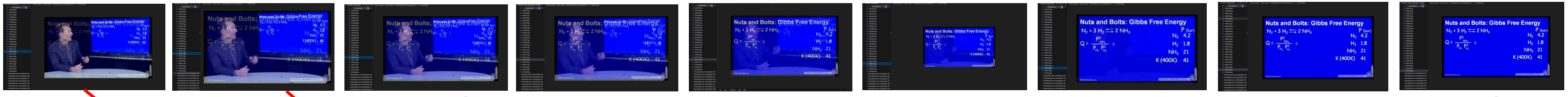
All frames are all “Chemistry PPT”, so the face detection is 0

```
Discrepancy found: /data1/gaoxiang_cong/TalkNet-ASD-main/Chem_buwan/Chem_frame/Nb76my9vHLY-face-Nb76my9vHLY-007 has 148 images, while /data1/gaoxi
ang_cong/TalkNet-ASD-main/Chem_buwan/Chem_face/Nb76my9vHLY-face-Nb76my9vHLY-007 has 0 images.
```

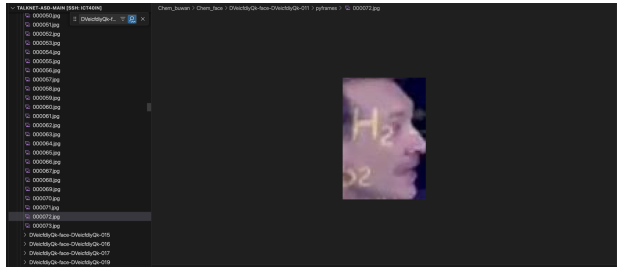
For example, in the later part, the screen gradually switches to “Chemistry PPT”, and the face of the chemistry teacher is no longer there.

Discrepancy found: /data1/gaoxiang_cong/TalkNet-ASD-main/Chem_buwan/Chem_frame/**DVeicfdiyQk-face-DVeicfdiyQk-011** has 80 images, while /data1/gaoxiang_cong/TalkNet-ASD-main/Chem_buwan/Chem_face/**DVeicfdiyQk-face-DVeicfdiyQk-011** has 73 images.

000072.jpg 000073.jpg 000074.jpg 000080.jpg



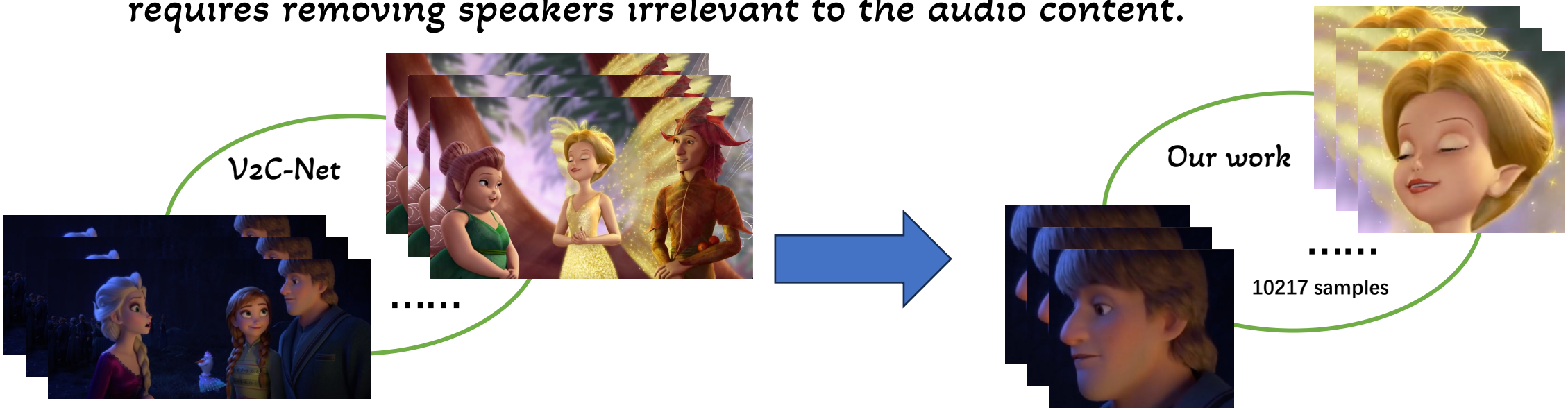
000072.jpg 000073.jpg 000074.jpg-000080.jpg



S3FD model cannot capture human faces anymore

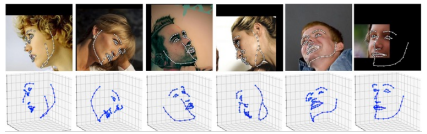
[More details] Face detection on V2C dataset

- The difficulty in scene processing of the V2C dataset is that unlike the chem dataset, which only has one speaker, there may be many speakers in a movie shot.
- The original V2C dataset (V2C: Visual Voice Clone [CVPR 2022]) did not take this into account did not differentiate between speakers, and used the entire scene as a visual cue. No corresponding annotation.
- One of the contributions of this work is the specification of movie speakers, which requires removing speakers irrelevant to the audio content.

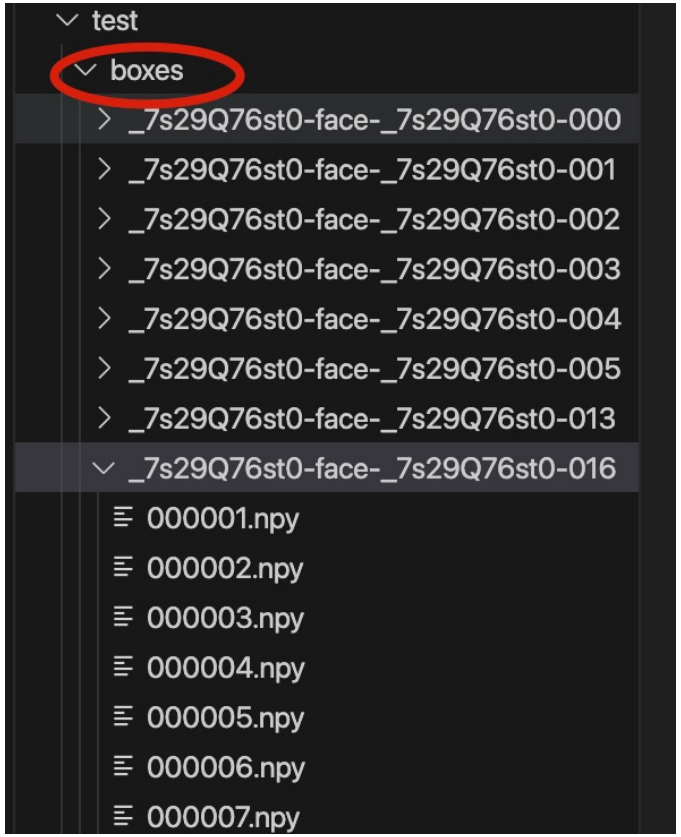


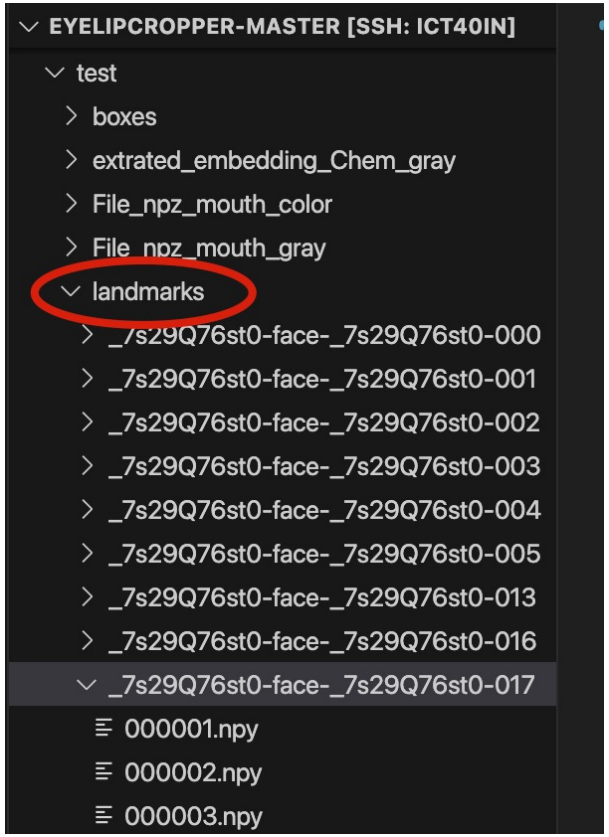
4. Get the mouth ROIs from face frames

- 4.1) Align faces to generate 68 landmarks and bounding boxes.



Please run `4_face_alignment.py` You will get





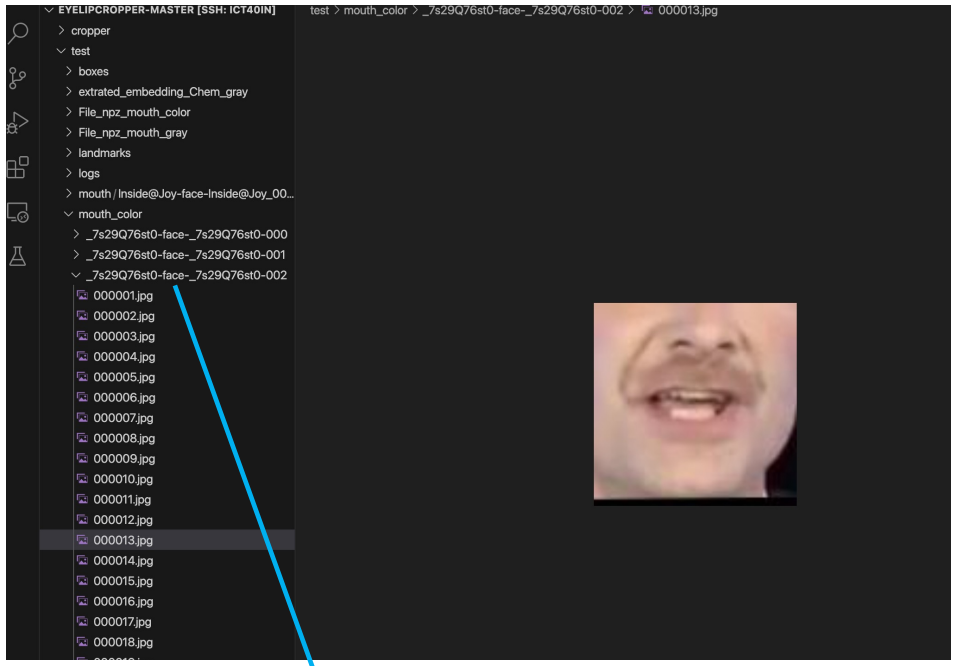
- This script is based on EyeLipCropper: https://github.com/zhliuworks/EyeLipCropper
- Facial landmarks refer to specific key points on the face that are used to help identify and analyze the structure of the face.
- A bounding box is a rectangular box used to define the location of a specific object in an image. In face recognition tasks, bounding boxes usually surround the detected face, indicating its position and size in the image.

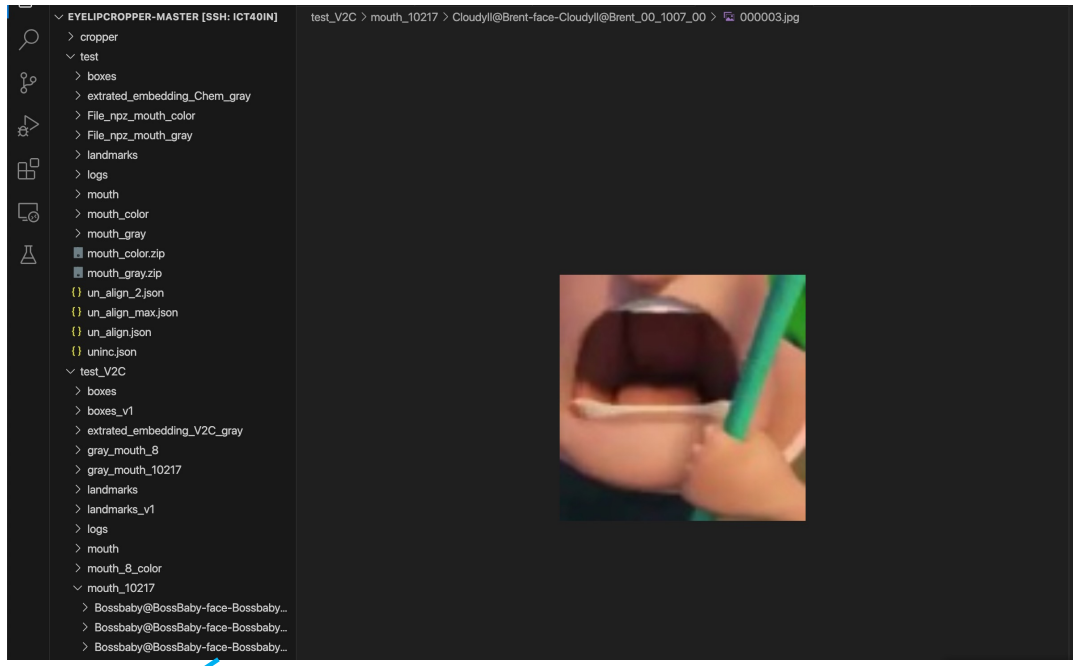
4. Get the mouth ROIs from face frames

- 4.2) Next, we crop the mouth ROIs based on the bounding boxes

Please run `5_eye_mouth_crop.py`

You will get **Lip region**





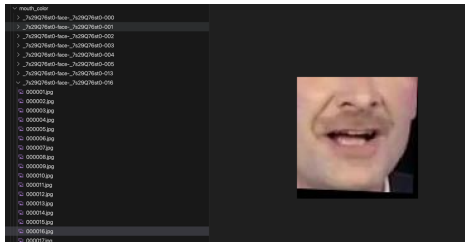
Chem dataset

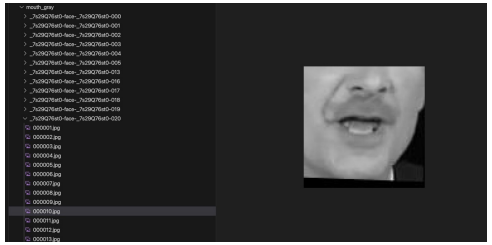
V2C dataset

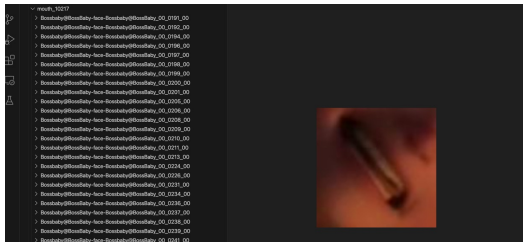
We accidentally named it wrong here. Writing "-face-" may be misleading, but the content inside is indeed the lip region (.jpg) we extracted.

[More details] Color or Gray?

- For the convenience of readers, we provide grayscale and color versions for the two datasets respectively. You can directly download our extracted mouth area through the connection:


color.jpg


gray.jpg


color.jpg

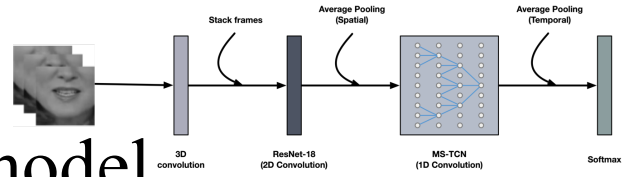

gray.jpg

https://drive.google.com/drive/folders/1d9rQjM7L7sfmTorrgnsNHe6Z1qiYT4yz?usp=drive_link

- Why are there grayscale and color versions?

  We first obtained the color version according to cropper (previous step), but we needed to follow the preprocessing process of Lipreading using Temporal Convolutional Networks (https://github.com/mpc001/Lipreading_using_Temporal_Convolutional_Networks), so we converted it to gray level. In short, We just used the gray mouth and got the lip feature by feeding it into Lipreading using Temporal Convolutional Networks. `extrated_embedding_Chem_gray`

4. Get the mouth ROIs from face frames



Lipreading using Temporal Convolutional Networks

- 4.3) Save as .npy (gray) or .npz(color) file to feed into model

Please run

6_create_npz.py

Aim to convert ".jpg"(previous step) to "Frame.npz"

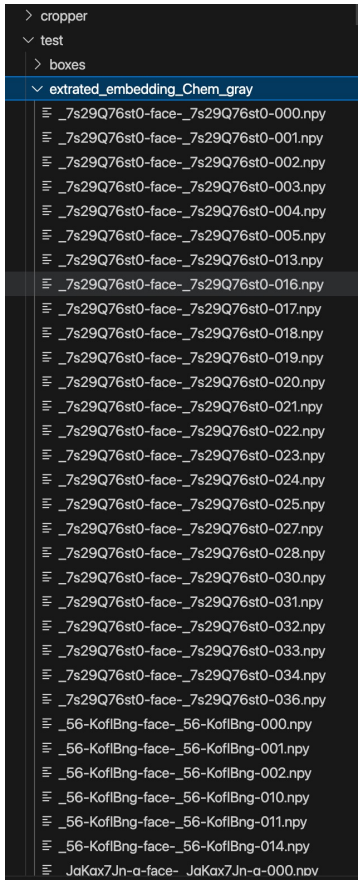
You will get

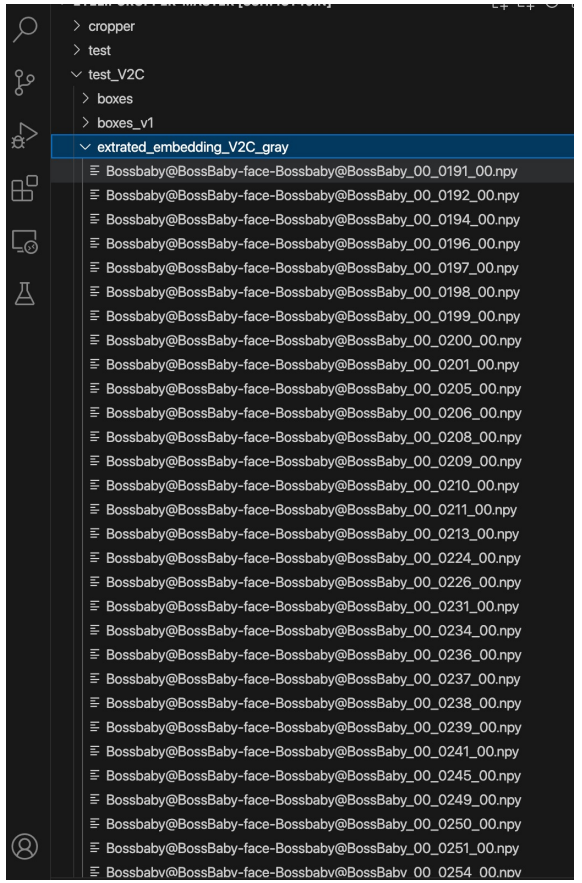
How to extract embeddings

We assume you have cropped the mouth patches and put them into *<MOUTH-PATCH-PATH>*. The mouth embeddings will be saved in the *.npz* format.

- To extract 512-D feature embeddings from the top of ResNet-18:

```
CUDA_VISIBLE_DEVICES=0 python main.py --modality video \
                                      --extract-feats \
                                      --config-path <MODEL-JSON-PATH> \
                                      --model-path <MODEL-PATH> \
                                      --mouth-patch-path <MOUTH-PATCH-PATH> \
                                      --mouth-embedding-out-path <OUTPUT-PATH>
```





Acknowledge

- We sincerely appreciate the help from these open-source projects and tool packages. Thanks to these authors to open source their code!
- https://github.com/TaoRuijie/TalkNet-ASD
- https://github.com/zhliuworks/EyeLipCropper
- https://github.com/face-analysis/emonet
- https://github.com/mpc001/Lipreading_using_Temporal_Convolutional_Networks
- https://montreal-forced-aligner.readthedocs.io/en/latest/
- https://github.com/FFmpeg/FFmpeg
- https://github.com/1adrianb/face-alignment
- https://github.com/yxlijun/S3FD.pytorch