Face Reconstruction from Speech

Shubham Nemani(203050011)
Pranjal Saini(203050014)
Ashwani Kumar Jha(20305R001)

Introduction

The aim of our project is to learn voice-face correlations that allow it to produce images that capture various physical attributes of the speakers such as age, gender and ethnicity.

Input: An audio spectrogram

Output: Reconstructed face image

We divide this task into 2 main components:

- An **voice encoder** that aims to learn the the facial features from audio,i.e., it learns voice face correlations.
- A **face decoder** that takes the predicted feature vector and generates a face image corresponding to the input feature vector.

Training pipeline

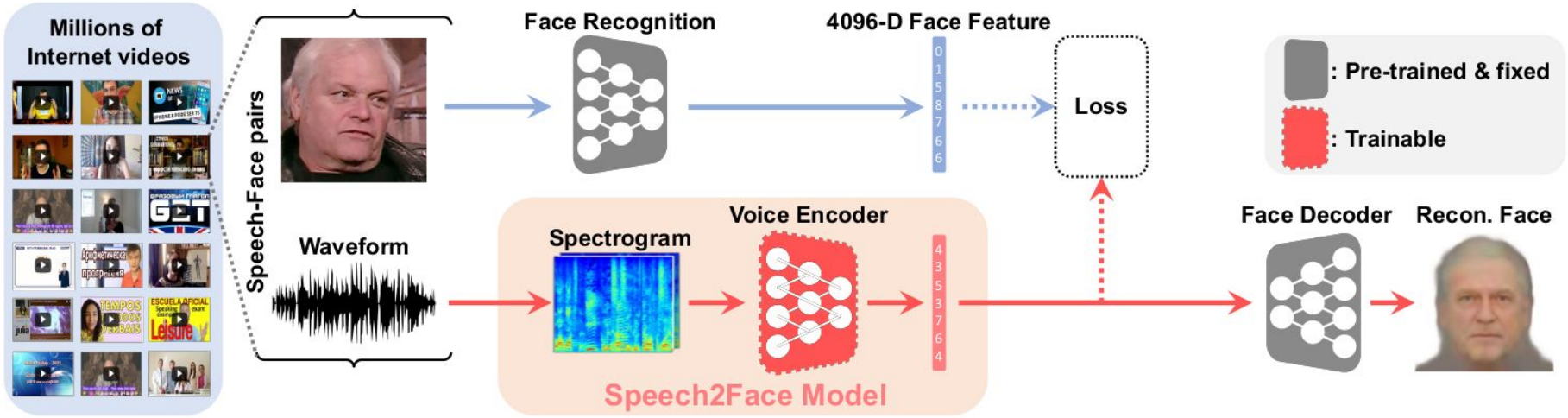
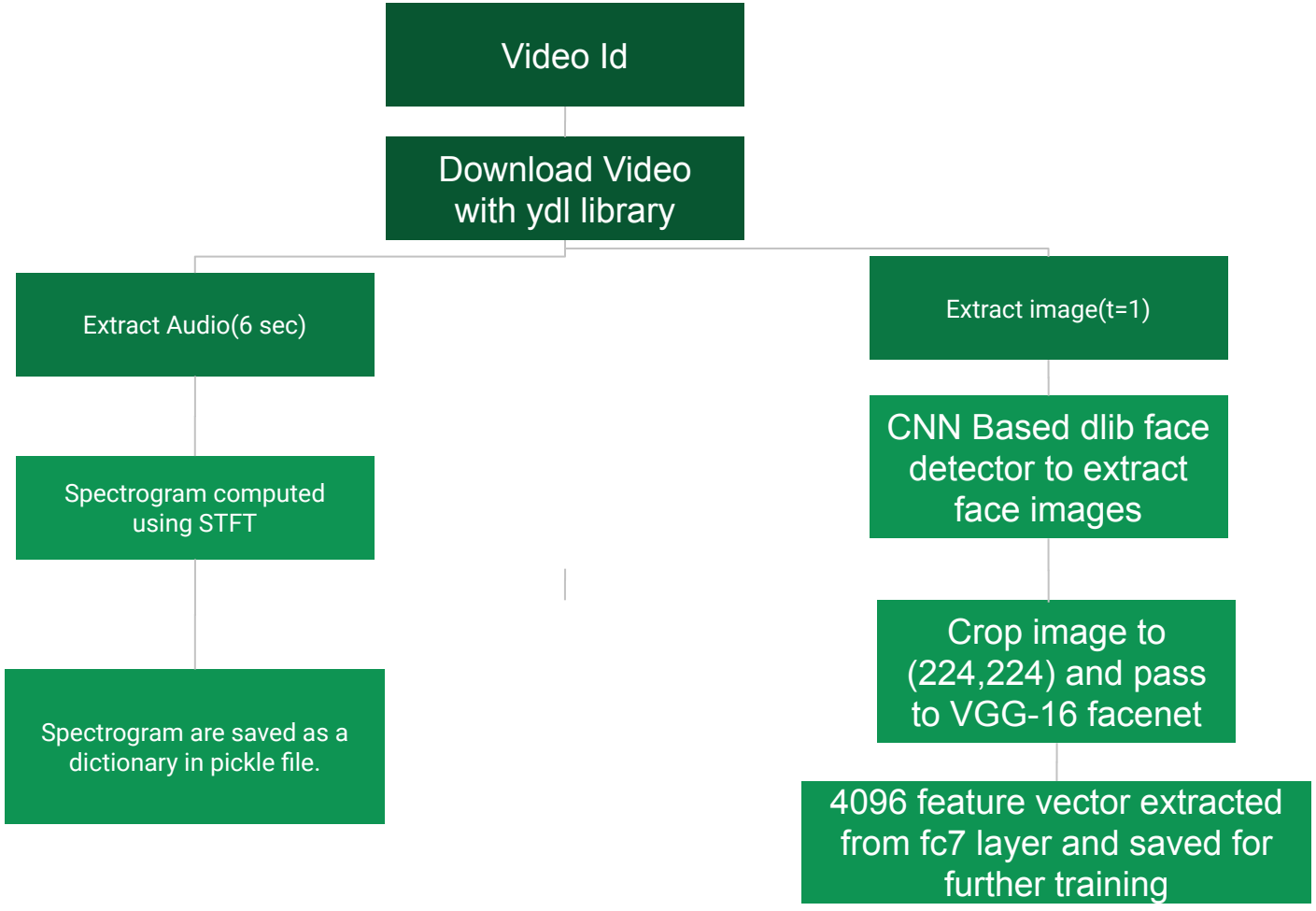


Figure 1. Model and training pipeline

Data Preprocessing pipeline



Data Preparation Details

1) **Dataset** -

AVSpeech Dataset (https://looking-to-listen.github.io/avspeech/download.html)
Train Size ~ 8K , Test Size ~ 400 , Time - 3-4 hrs per 1K

2) **Video Preprocessing** -

- We used youtube-dl to download the video and moviepy to extract corresponding audio and frame.
- We used CNN face detector from dlib for face recognition and VGG16-Face to get 4096-D face features.
- Used librosa and torchvision to compute complex spectrogram of input audio.
- Saved the audio spectrograms and VGG feature embeddings as pickle files to speed up the training process.

Voice Encoder

Our voice encoder module is a convolutional neural network which generates a 4096-dim feature vector corresponding to the input audio spectrogram.

We train the encoder in a self-supervised manner where the ground truth features are generated from corresponding face image of the person using a pre-trained VGG16-Face model.

We train the model to minimize the **L1- distance** between normalized true and predicted features.

Voice Encoder Architecture

Layer	Input	CONV RELU BN	CONV RELU BN	CONV RELU BN	MAXPOOL	CONV RELU BN	MAXPOOL	CONV RELU BN	MAXPOOL	CONV RELU BN	MAXPOOL	CONV RELU BN	CONV RELU BN	CONV	AVGPOOL RELU BN	FC RELU	FC
Channels	2	64	64	128	–	128	–	128	–	256	–	512	512	512	–	4096	4096
Stride	–	1	1	1	2×1	1	2×1	1	2×1	1	2×1	1	2	2	1	1	1
Kernel size	–	4×4	4×4	4×4	2×1	4×4	2×1	4×4	2×1	4×4	2×1	4×4	4×4	4×4	$\infty \times 1$	1×1	1×1

Total Trainable Parameters = 46593664

Voice Encoder has a CNN based architecture. We did not use original Voice Encoder proposed in Speech2Face paper because it has around **760 million** parameters with which it was not possible to train.

We decreased the model complexity and reduced the parameters to **46 million**.

Loss Function

We experimented with different types of losses:

1. L1 loss of normalised features (l1)- $\left\| \frac{\mathbf{v}_f}{\|\mathbf{v}_f\|} - \frac{\mathbf{v}_s}{\|\mathbf{v}_s\|} \right\|$
2. L2 loss of normalised features(l2) - $\left\| \frac{\mathbf{v}_f}{\|\mathbf{v}_f\|} - \frac{\mathbf{v}_s}{\|\mathbf{v}_s\|} \right\|_2^2$
3. Combination of L1 and L2 -

   Loss = l1 + lamda* l2 ; lamda = 10
4. Contrastive loss - $-\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{\exp\left(q_i^\top \cdot v_i\right)}{\sum_{j\neq i}\exp\left(q_i^\top \cdot v_j\right)}\right)$

- **We got the best performance on using L1 loss.**

Decoder

We have trained on face dataset **UTKface** . It contains face images from all age groups .

The images cover large variation in pose, facial expression , illumination, occlusion etc.

We have used **5900** images for training.

It is then fine-tuned using vgg-features generated from AVSpeech Dataset.

Architecture : It consists of 4 transpose convolutional layers, thus converting a 4096 face feature vector into a 224*224*3 RGB face image.

Loss Metrics - L1 loss between true and predicted face images.

Evaluation Metric(Encoder)

1. We calculated **L1-distance, L2-distance and cosine similarity** between normalised true and predicted features on train and test data.

	L2-Distance	**L1-Distance**	**Cosine Similarity**
Train Set(8K)	0.56	18.47	0.83
Test Set(400)	0.77	20.23	0.68

Results on different losses (Encoder)

Evaluation / Loss Metrics	L1 Distance	L2 Distance	Cosine Similarity
L1 Loss	20.23	0.77	0.68
L2 Loss	34.12	1.42	0.62
L1 loss + lambda * L2 loss	28.45	1.32	0.63
Contrastive Loss	43.67	2.35	0.49

2. Recall@K(Face Retrieval performance):

We measure the retrieval performance on reference(test) database of 400 samples.

For each sample in ref. set, we generate its feature vector using voice encoder and retrieve K topmost matching true feature vectors. Recal@K denotes the percent of times queried image is retrieved in top K.

	Train set(8K)					Test set(400)				
Similarity Measure	**R@1**	**R@5**	**R@10**	**R@20**	**R@50**	**R@1**	**R@5**	**R@10**	**R@20**	**R@50**
L1 Distance	11.96	29.26	41.73	55.47	72.26	0.51	1.77	3.03	6.83	14.18
L2 Distance	11.70	29.52	41.48	55.72	73.03	0.76	2.03	3.15	6.33	14.68
Cosine similarity	11.20	27.23	39.44	54.45	73.03	0.25	1.01	2.53	4.56	14.94

Face retrieval performance Results

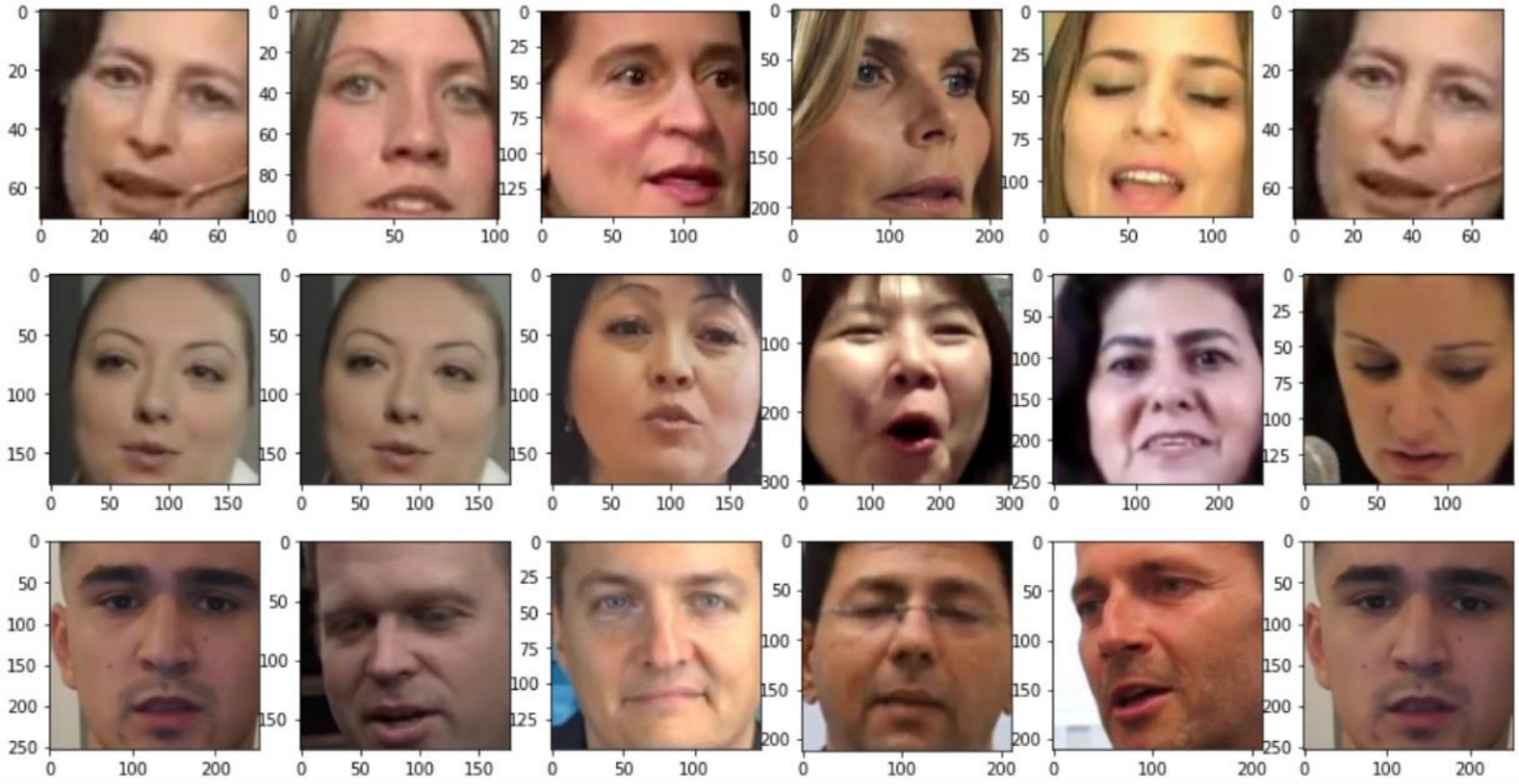


Figure. 3

Face Retrieval performance Results (cont.)

In figure 3, we obtained the top 5 matching images for some feature vectors generated by our voice encoder.

As we can see, our voice encoder has learned various physical attributes such as age and gender.

Because in every row, same gender images are extracted.

Also, in Row2, in almost all the retrievals, same expression images are extracted.

Qualitative Analysis of predicted images

For every example we show:

(left) the original image,

(middle) face decoder reconstruction from the VGG-Face feature extracted from the original image,

(right) face decoder reconstruction from the predicted VGG-Face feature from audio

Cosine Similarity between image predicted using VGG features and using audio waveform ranges between **0.25 - 0.50**

Qualitative Analysis of predicted images

References

- Speech2Face: Learning the Face Behind a Voice- https://arxiv.org/pdf/1905.09773.pdf
- Wav2Pix: speech-conditioned face generation using generative adversarial networks. - https://arxiv.org/pdf/1903.10195.pdf
- AVSpeech Dataset- https://looking-to-listen.github.io/avspeech/download.html
- UTKFace Dataset- https://susanqq.github.io/UTKFace/