EXTRACTION OF FACIAL FEATURES FROM SPEECH

(BASED ON SPEECH2FACE CVPR 2019 PAPER)

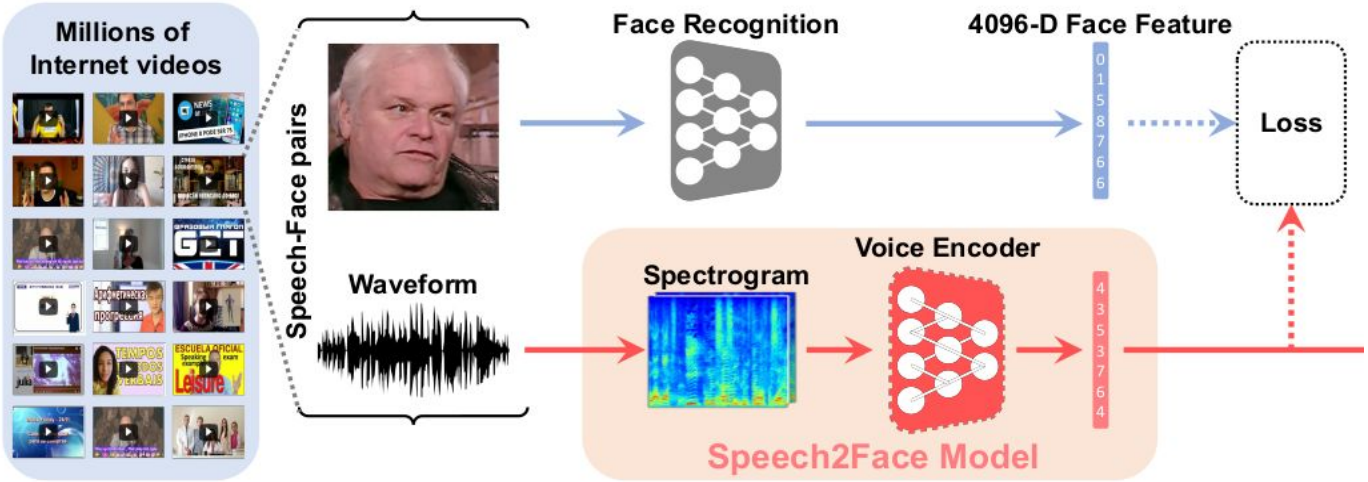
Neelesh Verma (160050062)
Ankit (160050044)
Saiteja Talluri (160050098)

ABSTRACT

- The main motivation was to infer about a person’s look from the way they speak.
- We split this task to two parts
  - First learn the facial features of a person from the speech (**SpeechToFace Model**)
  - Produce the face image from the features (FaceDecoder Model)

- During training of SpeechToFace, our model learns voice-face correlations and then we used this for **voice recognition** (as evaluation metric !!)
- This done in a self-supervised manner, by utilizing the natural co-occurrence of faces and speech in Internet videos, without the need to model attributes explicitly.

TRAINING PIPELINE



Pipeline is taken from the Speech2Face CVPR 2019 Paper [Tae-Hyun Oh et al.]

PREPROCESSING DATA

- AVSpeech Dataset (https://looking-to-listen.github.io/avspeech/download.html)
- Used **youtube-dl** library to download the videos from the csv files corresponding to start and end times.
- Used **ffmpeg** to extract audio and frames separately from the video.
- Used **librosa** and **tensorflow** libraries to compute stft and power law compression.
- Used **face_recognition** and keras **vgg-facenet** to find face boxes and compute 4096 face embedding vector
- Saved the audio spectrogram and embedding as pickle files to speed up the training process.

PREPROCESSING PIPELINE

AVSpeech Video Clip

Audio Clip (6 sec from start of the clip, repeat if less than 6 sec)

Resampled at **16kHz** and only **single** channel is used

Spectrogram S computed using STFT with Hann window of 25 mm, the hop length of 10 ms, and 512 FFT frequency bands.

S goes through the **power-law compression**, resulting sgn(S)|S|^0.3 for real and imaginary independently and then saved in pickle file

Used CNN based Dlib face detector to crop the face images

Crop the face image to (224,224) and pass to VGG FaceNet

4096 face feature vector extracted from fc7 layer is saved in pickle file to be used to compute loss.

Total Data (Training, Validation and Test)

Training Data

~ 4800 (80%)

Validation Data

~600 (10%)

Test Data

~600 (10%)

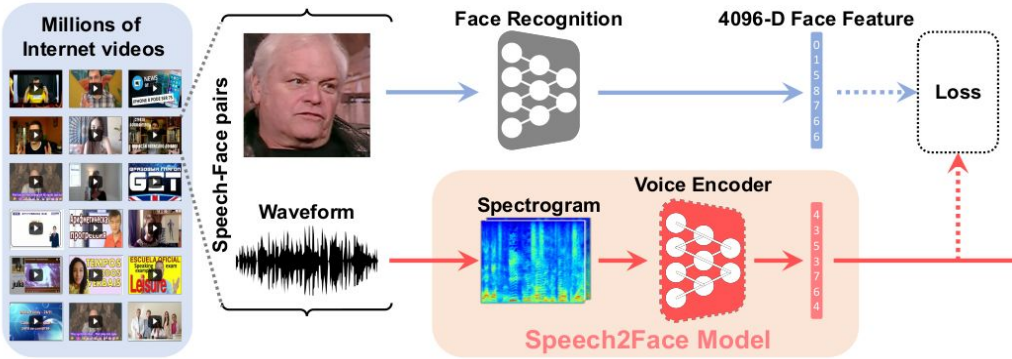
SPEECH ENCODER ARCHITECTURE

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	∞ × 1	1 × 1	1 × 1

Total params: 148,067,584 **Trainable params: 148,062,976** Non-trainable params: 4,608

Architecture is taken from the Speech2Face CVPR 2019 Paper [Tae-Hyun Oh et al.]

Loss Calculations



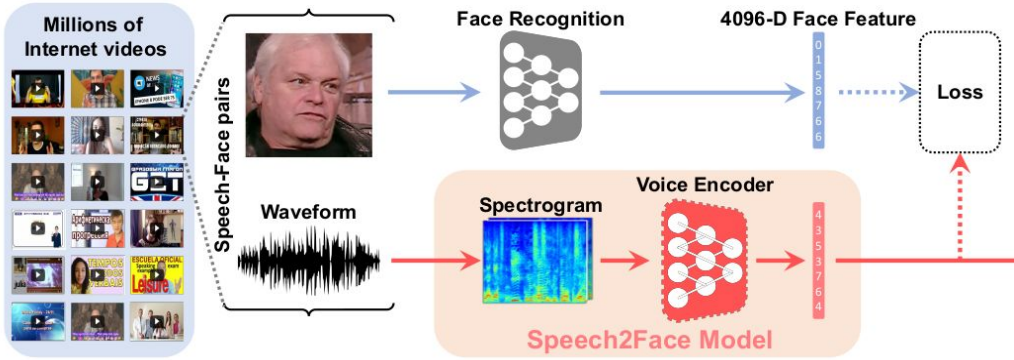
Some of the loss functions which can be explored.

- L1 loss - $\|\mathbf{v}_f - \mathbf{v}_s\|_1$
  - The author mentions that training undergoes slow and unstable progression with this loss.

- L2 loss of normalised features $\mathcal{L}_{\text{total}} = \left\| \frac{\mathbf{v}_f}{\|\mathbf{v}_f\|} - \frac{\mathbf{v}_s}{\|\mathbf{v}_s\|} \right\|_2^2$
  - **We used this loss function in our setup.**

Loss Functions taken from the Speech2Face CVPR 2019 Paper [Tae-Hyun Oh et al.]

Loss Calculations



Another interesting loss function to implement

$$\mathcal{L}_{\text{total}} = \lambda_1 \left\| \frac{\mathbf{v}_f}{\|\mathbf{v}_f\|} - \frac{\mathbf{v}_s}{\|\mathbf{v}_s\|} \right\|_2^2 + \lambda_2 L_{\text{distill}}\left(f_{\text{VGG}}(\mathbf{v}_f), f_{\text{VGG}}(\mathbf{v}_s)\right)$$

where $L_{\text{distill}}(\mathbf{a}, \mathbf{b}) = -\sum_i p_{(i)}(\mathbf{a}) \log p_{(i)}(\mathbf{b})$ $\quad p_{(i)}(\mathbf{a}) = \frac{\exp(a_i/T)}{\sum_j \exp(a_j/T)}$

- This loss additionally penalises the difference in activation of last layer of VGG Facenet (i.e., fc8)
  - L distill as an alternative of cross entropy loss, which encourages the output of Speech2Face to approximate the VGG.
  - Ensures stabilisation and little improvement
  - Could not implement due to memory constraints :(

Loss Functions taken from the Speech2Face CVPR 2019 Paper [Tae-Hyun Oh et al.]

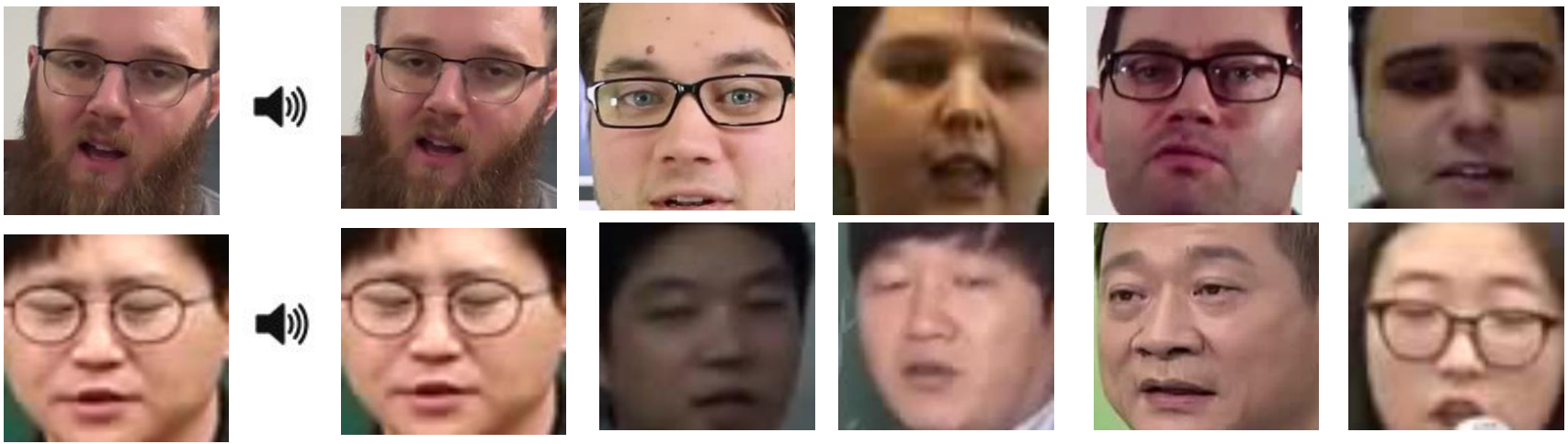
RESULTS (FACE RETRIEVAL PERFORMANCE)

	R@1	R@5	R@10	R@25	R@50	R@75	R@100
Train Data	45	52	55	58	62	64	66
Test Data	51	61	66	70	75	77	81

Table : SpeechToFace → Face retrieval performance. We measure retrieval performance by recall at K (R@K, in %), which indicates the chance of retrieving the true image of a speaker within the top-K results

Train Data contains a database of 4800 images on which the model is trained and Test Data contains 600 completely new images.

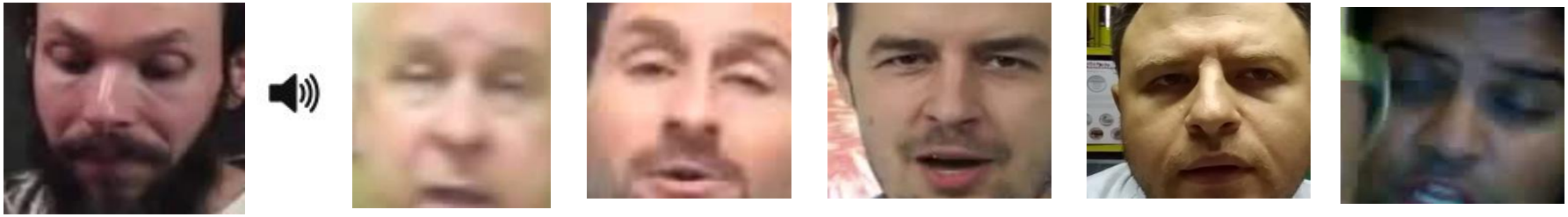
RESULTS (TOP 5 PREDICTIONS)



SpeechToFace→Face retrieval examples. We query a database of 600 face images by comparing our SpeechToFace prediction of input audio to all VGG-Face face features in the database. For each query, we show the top-5 retrieved samples.

First row (Perfect match i.e, top 1) - **Most of the predicted persons have spectacle and gender matches**.
Second row (Result in top 5) : **Speech suggests that the person is Chinese,** however gender mismatch in one of the result.

RESULTS (TOP 5 PREDICTIONS)



SpeechToFace→Face retrieval examples. We query a database of 600 face images by comparing our SpeechToFace prediction of input audio to all VGG-Face face features in the database. For each query, we show the top-5 retrieved samples. The above row is an example where the true face was not among the top results, this **may be attributed to too much beard** (which model doesn't learn properly owing to less such data) and **less quality of the cropped images due to which face features are not proper**

LIMITATIONS AND CHALLENGES

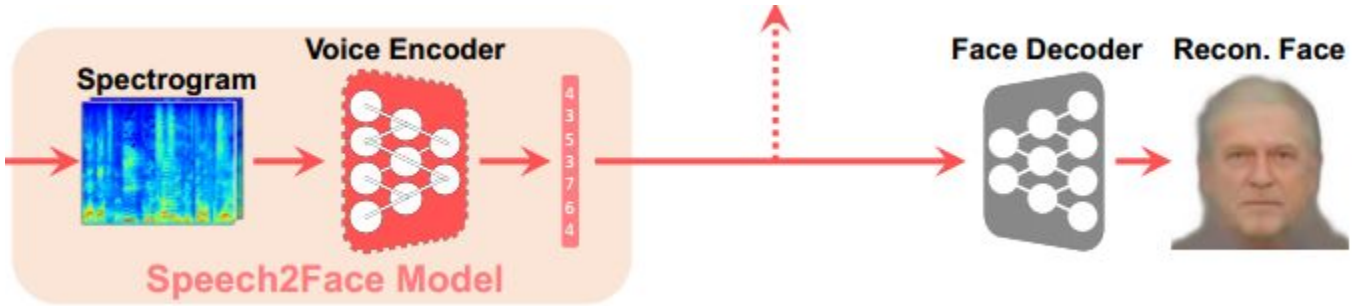
- Data Preprocessing for our task is very time taking for the AVSpeech Dataset (took almost 40-60 hrs to preprocess 6000 data)

- Requires multiple GPU as the model is very large and moreover we require vgg facenet during the loss calculation.

- More data (we used ~6000 samples whereas the paper mentions around ~150000), computation power and training time can increase the accuracy many fold !!

FUTURE WORK

Implementation of the Face Decoder Model, which takes as input the face features predicted by SpeechToFace model and produces an image of the face in a canonical form (frontal-facing and with neutral expression).



Pipeline is taken from the Speech2Face CVPR 2019 Paper [Tae-Hyun Oh et al.]

FUTURE WORK

- The pretrained Face Decoder Model used by the paper was not available and the model was based on another CVPR paper
  (Synthesizing Normalized Faces from Facial Identity Features)

- We tried implementing the model but this required lots of data for the model to train properly and the result was not even human recognizable.

- As the main focus of the project was on Speech Domain, we plan to complete this Vision task in the future.

References

- Speech2Face: Learning the Face Behind a Voice (https://arxiv.org/pdf/1905.09773.pdf)
    - We are very thankful to the authors [Tae-Hyun Oh et al.] for a wonderful paper.
    - Our work tries to implement the paper and make the code available.
- 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)

Thanks