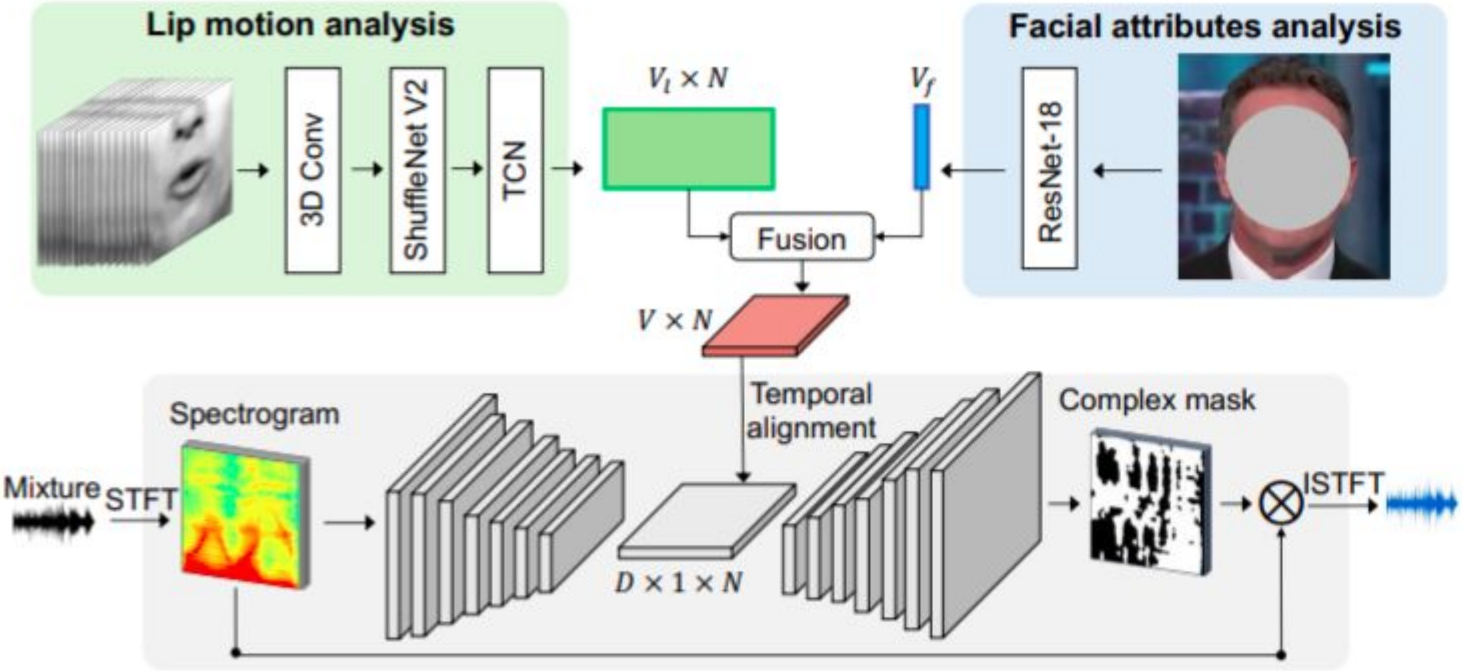
VISUALVOICE: Audio-Visual Speech Separation with Cross-Modal Consistency

VISUALVOICE: Audio-Visual Speech Separation with Cross-Modal Consistency

Objective : A multi-task learning framework to jointly learn audio-visual speech separation.


Lip motion analysis
3D Conv
ShuffleNet V2
TCN
$V_l \times N$
V_f
Facial attributes analysis
ResNet-18
Fusion
$V \times N$
Temporal alignment
Spectrogram
Mixture
STFT
$D \times 1 \times N$
Complex mask
ISTFT

Synthesising Training data

Two speech segments $s_{\mathcal{A}_1}(t),\ s_{\mathcal{A}_2}(t)$ f￼rom video V_A for speaker A, an$s_{\mathcal{B}}(t)$ from video V_B for speaker B. (Speaker consistency loss is used)

Let $F_{\mathcal{A}_1}, F_{\mathcal{A}_2}, F_{\mathcal{B}}$ denote the face tracks associated with the speech segments $s_{\mathcal{A}_1}(t),\ s_{\mathcal{A}_2}(t)$, $s_{\mathcal{B}}(t)$ respectively.

We create two mixture signals x_1(t) and x_2(t):

$$x_1(t) = s_{\mathcal{A}_1}(t) + s_{\mathcal{B}}(t), \quad x_2(t) = s_{\mathcal{A}_2}(t) + s_{\mathcal{B}}(t).$$

The mixture speech signals are then transformed into complex audio spectrograms X_1 and X_2.

Mixing two audio samples

1. **Get audio timeseries** (1D array representing amplitudes) using librosa.core.load()

2. **Average amplitudes** both audio samples , it represents audio time series of mixture

3. Get **complex frequency-time spectrogram** of mixed audio using librosa.load.stft()

4. This spectrogram is fed into audio-visual speech separation network

Visual Features Extraction

Face tracks $F_{\mathcal{A}_1}, F_{\mathcal{A}_2}, F_{\mathcal{B}}$ are given a input to lip motion and facial attributes analysis network.

The **lip motion** takes mouth **regions of interest (ROIs)** as input and uses ShuffleNet-v2 network to extract the final lip motion features of dimension **V_l × N** (ROI box size = 88*88 and V_l = 512 & N = 64)

For the **facial attributes** analysis network, ResNet-18 is used that takes a single face image randomly sampled from face track, extracts face embedding of dimension V_f that encodes the facial attributes of speaker (V_f=128)

Replicate the facial attributes feature along the time dimension to concatenate with the lip motion feature map (640*64-dimensional feature map)

V_{lp}, where $V = V_l \oplus V_f$ of dimension

of dimension 640*64.

So for each of the face track $F_{\mathcal{A}_1}, F_{\mathcal{A}_2}, F_{\mathcal{B}}$, we have a 2D visual feature map

Audio Feature Extraction

Encoder and a decoder network.

Encoder consists of convolutional and frequency pooling layer to downsample frequency dimension to 1 and time dimension to N = 64

The input to the encoder is the complex spectrogram of the mixture signal of dimension **2 × F × T** , here T is time = 256

Encoder Output is a audio feature map of dimension **D × 1 × N,** where D is the channel dimension.

Concatenate the visual and audio features along the channel dimension to generate an audio-visual feature map of dimension **(V + D) × 1 × N.**

Decoder takes the concatenated audio-visual feature as input and **predicts** a **complex mask** of dimension **2 × F × T**.

So using mixture spectrogram X_1 and F_{A1}, mask M_{A1} predicted for speaker A
using X_1 and F_B, mask M_{B1} predicted for speaker B
using X_2 and F_{A2}, mask M_{A2} predicted for speaker A
using X_2 and F_B, mask M_{B2} predicted for speaker B

Using **Tanh** layer to map the output feature map values to the range of [-1, 1].
And then using scaling by 5 so that output mask values are in range [-5,5]
because the real and imaginary parts of the ground-truth complex mask typically lie between -5 and 5.

Speech separation by Masking

The predicted **spectrograms for the separated speech** signals can be obtained by complex masking the mixture spectrograms:

$$S_{\mathcal{A}_i} = X_i * M_{\mathcal{A}_i}, \;\; S_{\mathcal{B}_i} = X_i * M_{\mathcal{B}_i}, \;\; i \in \{1, 2\},$$

Finally, using the inverse short-time Fourier transform (ISTFT) , we reconstruct the separated speech signals.

Mask Prediction Loss -

$$L_{mask\text{-}prediction} = \sum_{i\in\{\mathcal{A}_1,\mathcal{A}_2,\mathcal{B}_1,\mathcal{B}_2\}} \|M_i - \mathcal{M}_i\|_2,$$

$\mathcal{M}_i$ denotes the ground-truth complex masks, which are obtained by taking the complex ratio of the spectrogram of the clean speech to the corresponding mixture spectrogram.

Cross-modal matching loss-

The predicted spectrograms $\mathbf{S}_{A1}$, $\mathbf{S}_{B1}$, $\mathbf{S}_{A2}$, $\mathbf{S}_{B2}$ for separated speech signals are fed into ResNet-18 to extract audio embeddings $\mathbf{a}^{\mathcal{A}_1}, \mathbf{a}^{\mathcal{A}_2}, \mathbf{a}^{\mathcal{B}_1}, \mathbf{a}^{\mathcal{B}_2}$

This loss is to link voice and face.

Let i^A and i^B denote the face image embeddings e [illegible] al attribute network for speakers A and B, respectively. Using the following **triplet loss**:

$$L_t(\mathbf{a}, \mathbf{i}^+, \mathbf{i}^-) = \max\{0, D(\mathbf{a}, \mathbf{i}^+) - D(\mathbf{a}, \mathbf{i}^-) + \mathtt{m}\},$$

The distance between the embedding of the separated speech and the face embedding of the corresponding speaker is smaller than the distance between the separated speech embedding and the face embedding for the other speaker, by a margin.

The **cross-modal matching loss** is defined as follows:

$$\begin{aligned} L_{cross\text{-}modal} = {} & L_t(\mathbf{a}^{\mathcal{A}_1}, \mathbf{i}^{\mathcal{A}}, \mathbf{i}^{\mathcal{B}}) + L_t(\mathbf{a}^{\mathcal{A}_2}, \mathbf{i}^{\mathcal{A}}, \mathbf{i}^{\mathcal{B}}) \\ & + L_t(\mathbf{a}^{\mathcal{B}_1}, \mathbf{i}^{\mathcal{B}}, \mathbf{i}^{\mathcal{A}}) + L_t(\mathbf{a}^{\mathcal{B}_2}, \mathbf{i}^{\mathcal{B}}, \mathbf{i}^{\mathcal{A}}). \end{aligned}$$

Speaker Consistency Loss-

The audio embeddings for the separated speech segments for speaker A

should be more similar compared to that of speaker B.

speaker consistency loss on the audio embeddings of the separated speech; a

$$L_{consistency} = L_t(\mathbf{a}^{\mathcal{A}_1}, \mathbf{a}^{\mathcal{A}_2}, \mathbf{a}^{\mathcal{B}_1}) + L_t(\mathbf{a}^{\mathcal{A}_1}, \mathbf{a}^{\mathcal{A}_2}, \mathbf{a}^{\mathcal{B}_2}).$$