AV-GNNformer: Graph Neural Network and Conformer-Based Model for Audio-Visual Speech Enhancement (4th Edition COG-MHEAR AVSE Challenge)

Nasir Saleem[a], Fazal E Wahab[b], Arif Reza Anwary[a], Kia Dashtipour[a], Khubaib Ahmed[c], Adeel Hussain[a], Amir Hussain[a]

[a]*Edinburgh Napier University, EH10 5DT, Edinburgh, UK*
[b]*University of Science and Technology of China, , Hefei, China*
[c]*University of Stirling, , Stirling FK9 4LA, UK*

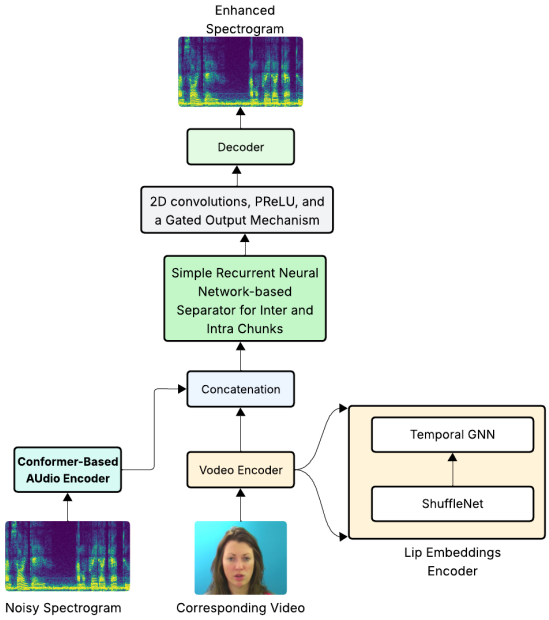

Figure 1: Architecture of the AVSE

Table 1: System Description

Parameter Type	Observations
Trainable Parameters	10.38M
Training Epochs	48 (Early Stop)
RTF	0.651 Seconds (On CPU)
	0.0806 Seconds (On GPU)
Model-Size (MB)	41.52
Training Hardware	NVIDIA TITAN X (Pasxal)
	Core™ i7-8700 CPU@3.20GHz (12 CPUs)
Time Per Epoch	255 Minutes (Approx)
Total Training Steps	431.5K
Batch Size	4
Training Time	212.5 Hours
Total Epochs	50
Optimiser	Adam
MACs/FLOPs (G)	32.45/64.94
Learning Rate	0.0001
DNN Framework	PyTorch, PyTorch Lightning

Audio Encoder: Conformer-Based Representation

The audio encoder in this architecture is based on a stack of Conformer blocks, designed to extract rich and robust features from the noisy audio input. Initially, a 1D convolutional layer is employed to project the raw single-channel audio waveform into a higher-dimensional embedding space defined by the encoder dimension. This projection, implemented using a `nn.Conv1d` layer, serves as a front-end feature extractor and allows the subsequent layers to operate in a more expressive feature space.

Visual Encoder: ShuffleNet + ST-GCN

The visual encoder is designed to efficiently extract spatial features from video frames while maintaining a low computational footprint. Each input frame, representing the speaker's facial region or mouth area, is independently processed using a lightweight convolutional neural network, specifically ShuffleNet. ShuffleNet leverages pointwise group convolutions and channel shuffling operations to dramatically reduce computational complexity while preserving representational power. Following feature extraction, a temporal graph is constructed in which each node corresponds to a time step, representing the audio-visual features at that moment. Edges are created between temporally adjacent frames, forming a structured and time-aware graph that preserves sequential relationships in the data. This graph is then passed through a Temporal Graph Neural Network (GNN), such as a Spatio-Temporal Graph Convolutional Network (ST-GCN) or a Temporal Graph Attention Network (TGAT). These networks are designed to model complex temporal dependencies by propagating and aggregating information across time through the graph structure.

After temporal modeling, the visual and audio embeddings obtained from the GNN are fused. This is done through simple concatenation operation. The fused features are then processed by a lightweight sequential model—Simple Recurrent Unit (SRU)—to generate the estimates for masking and finally the enhanced speech output from audio decoder.