Supplementary Material for "End-to-End Speech-Driven Facial Animation with Temporal GANs"

Konstantinos Vougioukas[1]
k.vougioukas@imperial.ac.uk
Stavros Petridis[12]
stavros.petridis04@imperial.ac.uk
Maja Pantic[12]
m.pantic@imperial.ac.uk

[1] iBUG Group
Dept. Computing
Imperial College London
London, UK

[2] Samsung AI Centre
Cambridge, UK

1 Audio Preprocessing

The sequence of audio samples is divided into overlapping audio frames in a way that ensures a one-to-one correspondence with the video frames. In order to achieve this we pad the audio sequence on both ends and use the following formula for the stride:

$$stride = \frac{rate_{audio}}{rate_{video}} \tag{1}$$

2 Network Architecture

This section describes, in detail, the architecture of the networks used in our temporal GAN. All our networks use *ReLU* activations except for the final layers. The encoders and generator use the hyperbolic tangent activation to ensure that their output lies in the set $[-1,1]$ and the discriminator uses a Sigmoid activation.

2.1 Audio Encoder

The *Audio Encoder* network obtains features for each audio frame. It is made up of 7 Layers and produces an encoding of size 256. This encoding is fed into a 2 layer GRU which will produce the final context encoding.

2.2 Noise Generator

The *Noise Generator* is responsible for producing noise that is sequentially coherent. The network is made up of GRUs which take as input at every instant a 10 dimensional vector sampled from a Gaussian distribution with mean 0 and variance of 0.6. The *Noise Generator* is shown in Figure 2.

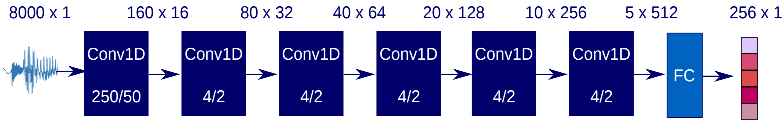

Figure 1: The deep audio encoder used to extract 256 dimensional features from audio frames containing 8000 samples. Convolutions are described using the notation *kernel / stride*. The feature dimensions after each layer are shown above the network using the notation *feature size* $\times$ *number of feature maps*.

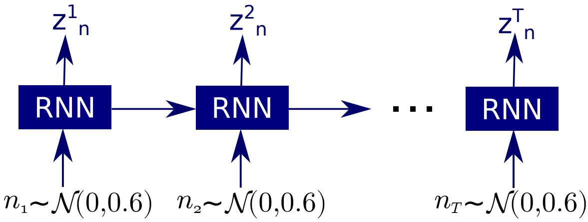

Figure 2: The network that generates the sequential noise

2.3 Identity Encoder and Frame Decoder

The *Identity Encoder* is responsible for capturing the identity of the speaker from the still image. The *Identity Encoder* is a 6 layer CNN which produces an identity encoding z_{id} of size 50. This information is concatenated to the context encoding z_c and the noise vector z_n at every instant and fed as input to the *Frame Decoder*, which will generate a frame of the sequence. The *Frame Decoder* is a 6 layer CNN that uses strided transpose convolutions to generate frames. The *Identity Encoder* - *Frame Decoder* architecture is shown in Figure 3

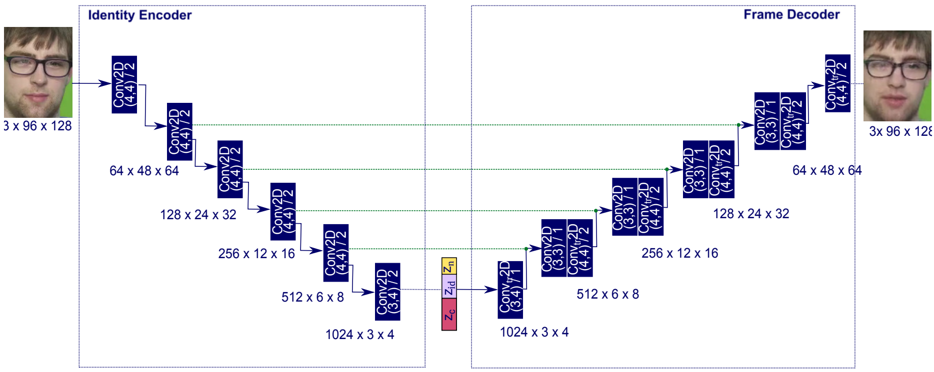

Figure 3: The U-Net architecture used in the system with skip connections from the hidden layers of the *Identity Encoder* to the *Frame Decoder*. Convolutions are denoted by *Conv2D* and transpose convolutions as $Conv_{tr}2D$. We use the notation *(kernel$_x$, kernel$_y$) / stride* for 2D convolutional layers.

3 Datasets

The model is evaluated on the GRID and TCD TIMIT datasets. The subjects used for training, validation and testing are shown in Table 1

Dataset	Training	Validation	Testing
GRID	1, 3, 5, 6, 7, 8, 10, 12, 14, 16, 17, 22, 26, 28, 32	9, 20, 23, 27, 29, 30, 34	2, 4, 11, 13, 15, 18, 19, 25, 31, 33
TCD TIMIT	1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 16, 17, 19, 20, 21, 22, 23, 24, 26, 27, 29, 30, 31, 32, 35, 37, 38, 39, 40, 42, 43, 46, 47, 48, 50, 51, 52, 53, 57, 59	34, 36, 44, 45, 49, 54, 58	8, 9, 15, 18, 25, 28, 33, 41, 55, 56

Table 1: The subject IDs for the training, validation and test sets for the GRID and TCD TIMIT datasets.