Building Secure and Engaging Video Communication by Using Monitor Illumination

Supplementary Material

A. Overview of Appendices

Our appendices contain the following additional details and results:

- In Sec. B, we provide a more comprehensive overview of the model architecture for VS2, as illustrated in Sec. 4.
- The results for the predictions of the monitor, initially depicted in the main paper, are further elaborated upon in Sec. C.

B. Model Details

For the model architecture, a convolutional layer is denoted by conv x,y,z where x,y,z stands for the kernel width, stride and padding. conv* denotes a layer that has been followed by a pixel normalization (PixelNorm) and PReLU.

Feature extractor The feature extractor for a final layer ViT feature extractor is the final class token of size $3 \times 16 \times 16$.

The feature extractor for a multiple layer ViT feature extractor (in VS2) takes the 12 intermediate class tokens from the output of all 12 Transformer layers.

The ResNet models with a single layer feature extractor takes the output feature volume after the final convolutional layer, which is of dimensions $512 \times 7 \times 7$.

The ResNet models with a multiple layer feature extractor take four features from the 4th, 8th, 12th and 16th 3x3 convolutional layers. These features have dimensions $64 \times 56 \times 56$, $128 \times 28 \times 28$, $256 \times 14 \times 14$, and $512 \times 7 \times 7$ respectively.

Generator The convolutional generator in the RC model takes features of dimensions $512 \times 7 \times 7$ and passes it through multiple convolutional layers.

layer	output size
conv* 3,1,1	$512 \times 7 \times 7$
conv* 3,1,1	$1728 \times 7 \times 7$
avg pool	$1728 \times 1 \times 1$

Table 1. We show the architecture of the RC model generator.

The output is reshaped into a $3 \times 32 \times 18$ output which is the predicted monitor.

The conv generator in the VC uses a similar architecture, except the input volume is of size $3 \times 16 \times 16$.

The generator in each StyleGAN-based model is a pretrained StyleGAN-XL, which outputs a 64×64 image. We resize the actual monitor image to 64×64 when training the StyleGAN models

Mapping network Only the StyleGAN-based model have a mapping network. The output of the VS1 feature extractor is simply the final class token of size $3 \times 16 \times 16$. This is flattened and passed through a mapping network, which consists of a fully connected layer with 512 outputs.

In VS2, the output class token is flattened into a vector with $12 \cdot 3 \cdot 16 \cdot 16 = 9216$ dimensions, which is passed through a mapping network. This consists of a fully connected (FC) layer with 1024 outputs, a PixelNorm layer, followed by a PReLU layer and another FC layer with 512 outputs.

In RS1, the output feature volume of size $512 \times 7 \times 7$ is passed through a convolutional network.

layer	output size
conv* 3,1,1	$512 \times 7 \times 7$
conv* 3,1,1	$512 \times 7 \times 7$
conv 3,1,1	$512 \times 7 \times 7$
max pool	$512 \times 1 \times 1$

Table 2. We show the architecture of the RS1 model mapping network.

In RS2, the intermediate features are passed through four separate convolutional networks to yield feature vectors of size 128, 256, 512, and 512 which is then concatenated into a single vector, passed through a ReLU and another FC layer with 512 outputs. The architecture for the convolutional mapping networks are in Tab. 3.

Task classifier For task classifier TC1, the intended monitor is separately processed by a convolutional encoder, which contains one FC layer with 512 outputs and another FC layer with 256 outputs. Both layers are followed by a PixelNorm layer and PReLU activation.

The pretrained feature encoder (ViT or ResNet) from the face capture image is followed by a fully connected layer with 256 outputs, then a PixelNorm layer and PReLU activation.

Then, the two 256-dimensional feature vectors are concatenated. This is passed into a sequence of 3 FC layers with 128, 32, and 1 outputs respectively. The first two FC layers are each followed by a PixelNorm and ReLU activation.

TC2 has the same design, except that the predicted monitor image is used instead of the features from the pretrained

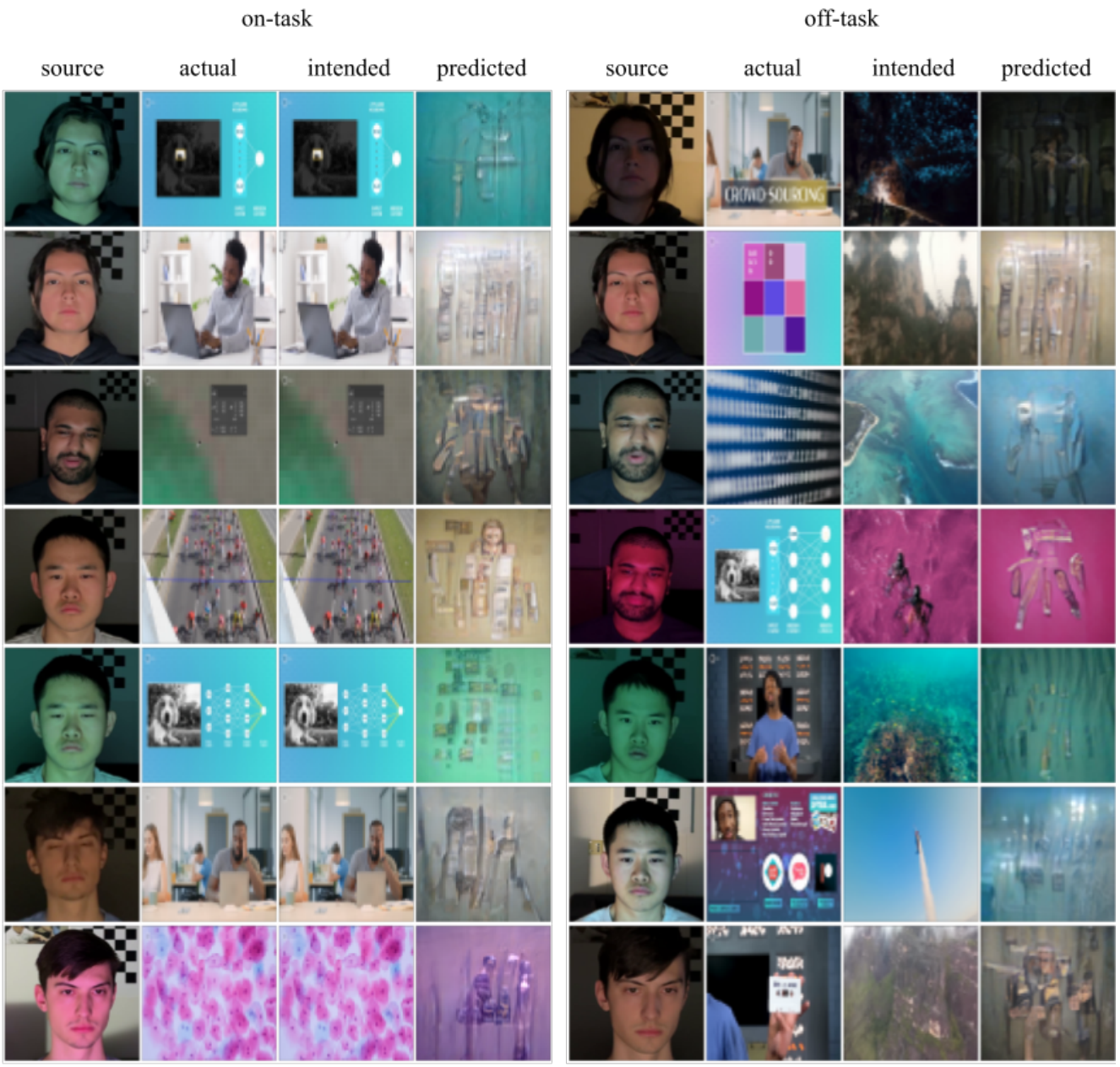

Figure 1. We show more samples of monitor images for the 'on-task' (left) and 'off-task' (right) case.

input size	$64 \times 56 \times 56$	input size	$128 \times 28 \times 28$	input size	$512 \times 14 \times 14$	input size	$512 \times 7 \times 7$
layer	output size	layer	output size	layer	output size	layer	output size
conv* 2,2,0	$128 \times 28 \times 28$	conv* 2,2	$128 \times 14 \times 14$	conv* 2,2,0	$512 \times 7 \times 7$	conv* 3,1,1	$512 \times 7 \times 7$
conv* 2,2,0	$128 \times 14 \times 14$	conv* 2,2	$256 \times 7 \times 7$	conv* 3,1,1	$512 \times 7 \times 7$	conv* 3,1,1	$512 \times 7 \times 7$
max pool	$128 \times 1 \times 1$	max pool	$256 \times 1 \times 1$	max pool	$512 \times 1 \times 1$	max pool	$512 \times 1 \times 1$

Table 3. We show the architecture for the RS2 intermediate features mapping network convolutional layers. From left to right, each subtable represents a convolutional network that takes in the first, second, third and fourth intermediate feature from the ResNet feature extractor respectively.

monitor. This predicted monitor image is passed through the same conv. encoder used for the intended monitor.

C. Reconstructed Monitor Result