β-VAE as a Flexible and Light Codec Alternative for Video Conferencing

CS 236 Final Project

Developed by Colin Sullivan at Stanford University

Intro

Most of the previous ML research on facial expression data compression has been centered on facial expression recognition (FER).

This paper explores a new potential use case: facial expression data compression for video conferencing.

Using generative models, we were able to model the compressed latent representation of a single user's facial expression data which could, potentially, be transmitted and decoded in real time.

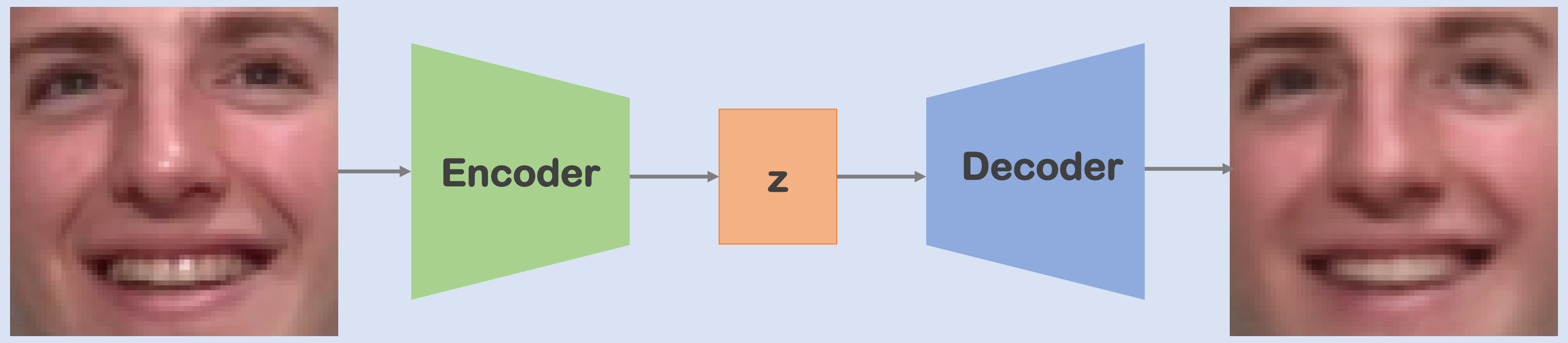

Fig 1. Diagram of General Autoencoder Structure

Model and Structure

The Convolutional β-VAE restricts the latent space such that it can be easily sampled from, and meaningful interpolations can be performed between compressed data points.

Using a lightweight network, we were able to compress the data to the size of a single IP payload (<64 KB) which can be transmitted efficiently on virtually any modern network.

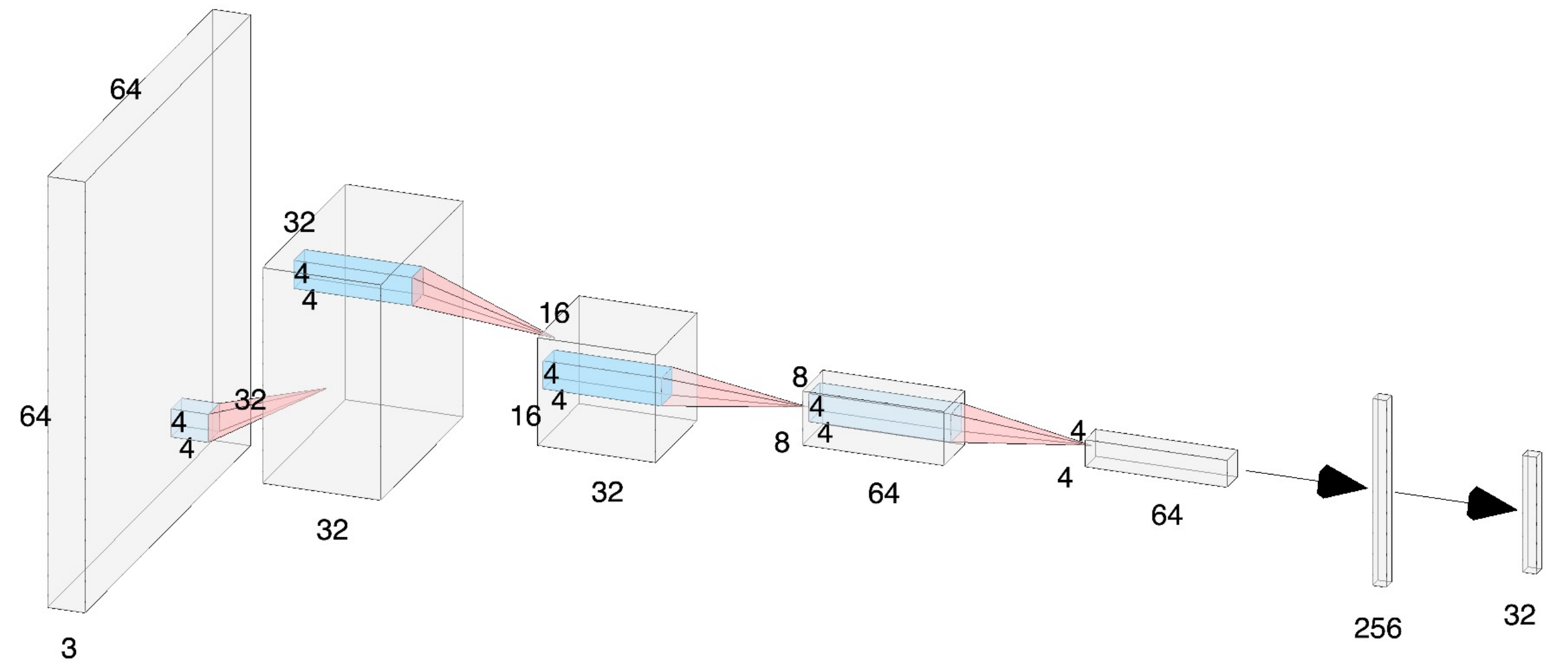

Fig 2. β-VAE Encoder Network Structure (Decoder is reversed)

The input was a half hour video of a single user. Every eighth frame was kept and on each frame a face cropping was created using HAAR cascades. These cropped images were reduced in size to 64 by 64 pixels and fed into the autoencoder as input.

The output was the modeled latent distribution of the facial image data.

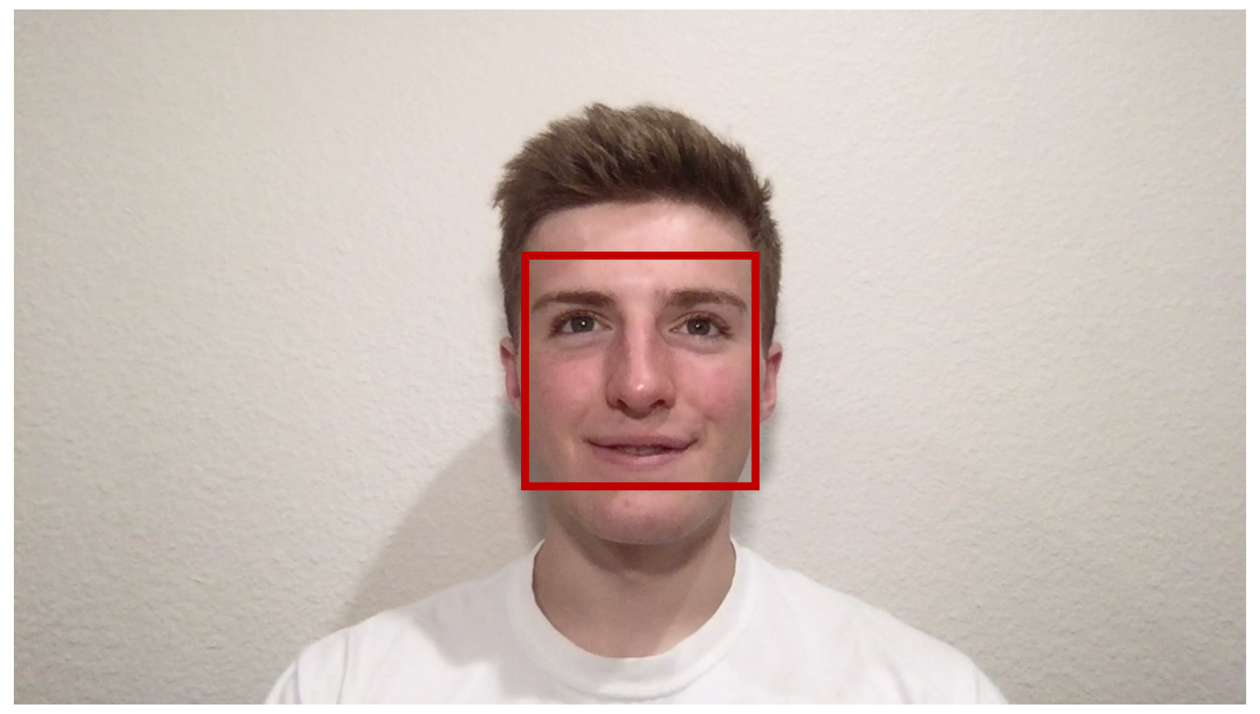

Fig 3. Face Image Cropping Using HAAR Cascades

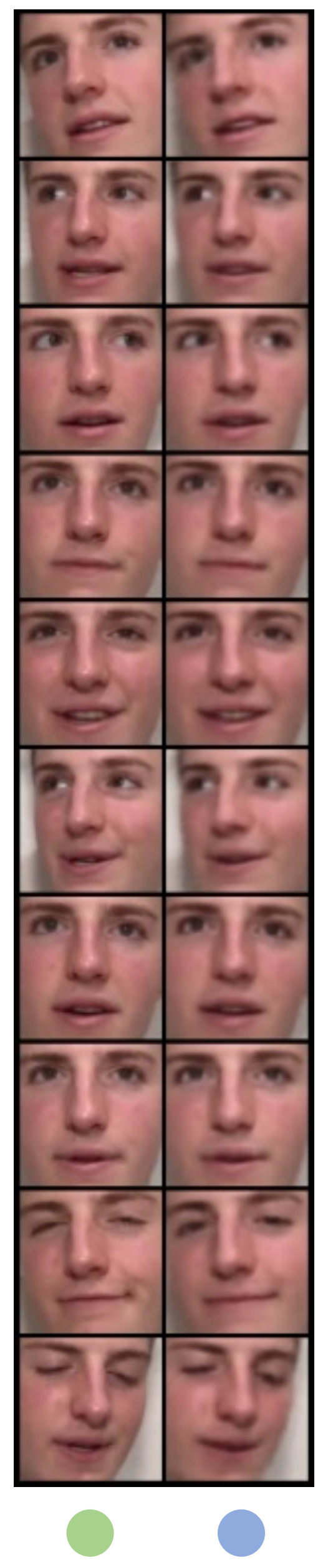

Fig 4. Side-by-Side Real/Generated Facial Images

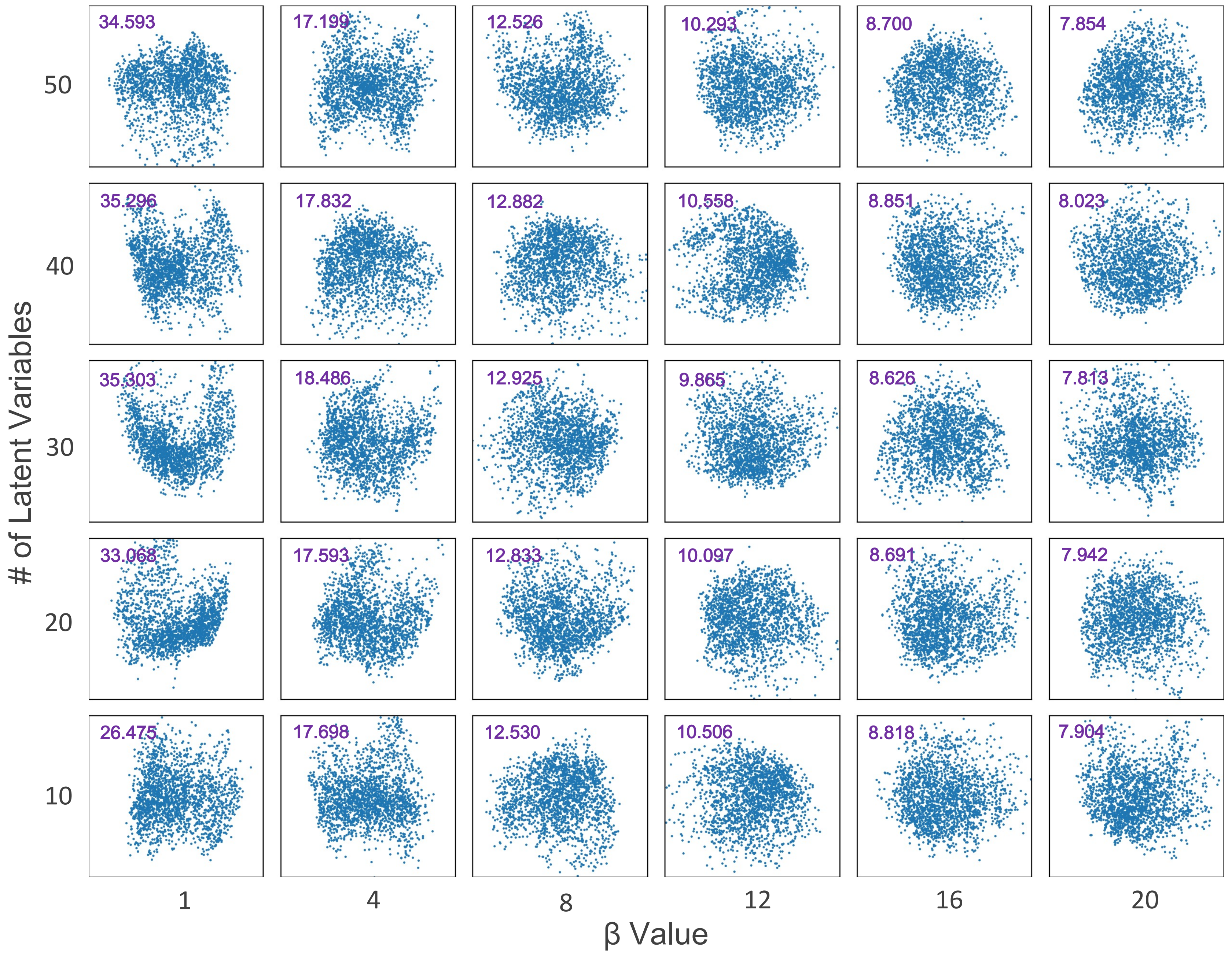

Fig 5. PCA-Compressed Latent Distributions of β-VAE Models With KL Divergence

The models were able to achieve a test PSNR of up to **33dB** which is mediocre for images of this size. The models captured most facial cues (e.g. gaze, head tilt, creasing, etc) with some blurriness, especially on less common images.

Changing β and the number of latent values above 20 had little effect on the reconstruction quality, which suggests that even fewer than 10 latent variables could have been used. The reconstruction qualities of our models increased slightly as β decreased and as the number of latent variables increased.

The normality of our models' latent spaces increased with β value, as expected.

One interesting and potentially useful interpolation is the smile interpolation. To perform the smile interpolation, we shift latent points by some vector v_s such that the resulting decoded image appears to smile more.

In order to find an optimal v_s several smile images were labeled, and the optimal vector was discovered using the steps below.

The results for five randomly sampled images are on the right with the real sampled images labeled green.

Deriving Smile Interpolation Vector

1. $\text{argmax}_{v_s}\ (\sum_{z\sim smile} v_s \cdot z - \sum_{z\sim nosmile} v_s \cdot z)$ s.t. $|v_s|_2 = 1$
2. $\nabla(\sum_{z\sim smile} v_s \cdot z - \sum_{z\sim nosmile} v_s \cdot z - c[\![|v_s|_2^2]\!]) = \sum_{z\sim smile} z - \sum_{z\sim nosmile} z - 2cv_s$
3. $0 = \sum_{z\sim smile} z - \sum_{z\sim nosmile} z - 2cv_s$
4. $v_s \sim \sum_{z\sim smile} z - \sum_{z\sim nosmile} z$
5. $v_s = (\sum_{z\sim smile} z - \sum_{z\sim nosmile} z)\frac{1}{|v_s|_2}$
6. $\hat{z} = z + c \cdot v_s$

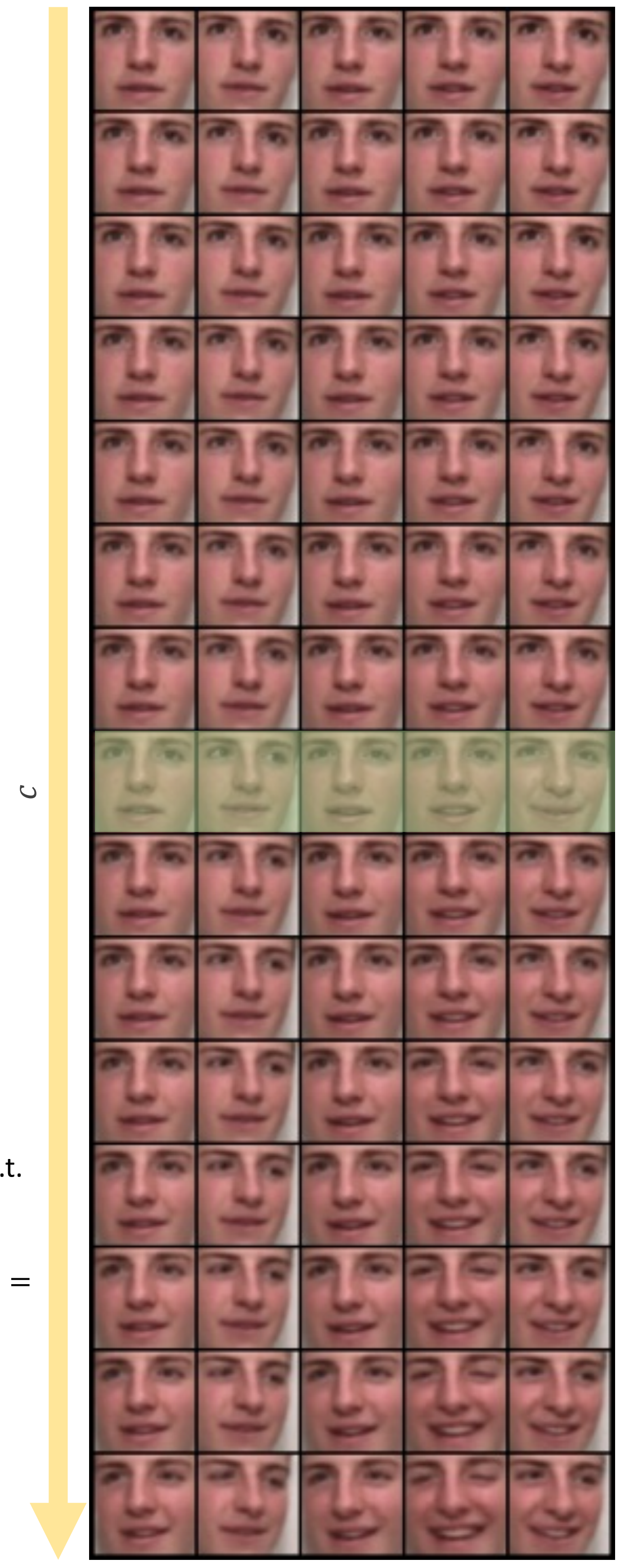

Fig 6. Smile Interpolation

Conclusion

While extremely restrictive and far from state-of-the-art image reconstruction quality, β-VAE's do offer an excitingly light and flexible alternative to the traditional video codec.

Further research could be done on encoding larger images, the relation between consecutive frames in the latent space, working with 3D data, and improving reconstruction quality.