MoCoGan: Decomposing Motion and Content for Video Generation

As the title says, the main point of this work is the separation of a video in two subspaces: Content and Motion. This paper is proposing a GAN based framework that generates **video frames** from random vectors.

Each vector consists of a content part and a motion part, and because the content doesn't change much in a short term video, that part remains the same for the whole generation (extracted from a Gaussian distribution), while the motion part is stochastic generated (using a RNN).

Motion and Content Decomposed GAN

To be able to generate videos of different lengths and videos with the same length but executed with different speeds, in this paper is used a latent space Zi of images. A video of length K is represented by a path [z1,..,zk] in this latent space. This space is further decomposed into the content and motion subspaces.
The content subspace is modelled using a Gaussian (normal) distribution, and the zc variable is generated only once for the whole video.

$$\mathbf{z}_{\mathrm{C}} \sim p_{Z_{\mathrm{C}}} \equiv \mathcal{N}(\mathbf{z}|0, I_{d_{\mathrm{C}}})$$

The motion subspace is modelled by a path in the Zm subspace. Since not all the paths in this subspace correspond to plausible motion, a recurrent neural network is used to generate valid paths. At each time step, this network takes a vector sampled from a Gaussian distribution and outputs a vector in the Zm subspace. The RNN consists of one GRU layer.

Networks

This work uses 4 neural networks: the recurrent network that generates **motion paths**, the **image generator** Gi, the **image discriminator** Di and the **video discriminator** Dv.
Gi generates a video clip by sequentially mapping vectors from Zi to images, from a sequence of vectors like : $[\begin{bmatrix}\mathbf{z}_{\mathrm{C}} \\ \mathbf{z}_{\mathrm{M}}^{(1)}\end{bmatrix}, ..., \begin{bmatrix}\mathbf{z}_{\mathrm{C}} \\ \mathbf{z}_{\mathrm{M}}^{(K)}\end{bmatrix}]$ to a sequence of images : $\tilde{\mathbf{v}} = [\tilde{\mathbf{x}}^{(1)}, ..., \tilde{\mathbf{x}}^{(K)}],$ where

zM is from the recurrent network. Both DI and DV play the judge role, providing criticisms to GI and RM. Di is specialised in criticising Gi based on individual images, while Dv provides criticism based on the generated video clip. Dv is also based on a spatio-temporal CNN architecture and evaluates the generated motion. Because Gi has no concept of motion, the criticism from Dv goes to the recurrent network.
The authors note that Dv should be enough because it provides feedback on both static image appearance and video dynamics but Di improves the convergence of the adversarial training.

Objective function

The learning problem of MoCoGAN is:

$$\max_{G_{\mathrm{I}}, R_{\mathrm{M}}} \min_{D_{\mathrm{I}}, D_{\mathrm{V}}} \mathcal{F}_{\mathrm{V}}(D_{\mathrm{I}}, D_{\mathrm{V}}, G_{\mathrm{I}}, R_{\mathrm{M}})$$

The objective function is:

$$\mathbb{E}_{\mathbf{v}}[-\log D_{\mathrm{I}}(S_1(\mathbf{v}))] + \mathbb{E}_{\tilde{\mathbf{v}}}[-\log(1 - D_{\mathrm{I}}(S_1(\tilde{\mathbf{v}})))] +$$
$$\mathbb{E}_{\mathbf{v}}[-\log D_{\mathrm{V}}(S_{\mathrm{T}}(\mathbf{v}))] + \mathbb{E}_{\tilde{\mathbf{v}}}[-\log(1 - D_{\mathrm{V}}(S_{\mathrm{T}}(\tilde{\mathbf{v}})))],$$

In this formula, the first and second terms encourage Di to output 1 for a video frame from a real video and 0 for a generated one. Similarly, the third and fourth terms encourage Dv to output 1 or 0 depending if the frame sequence is real or generated.

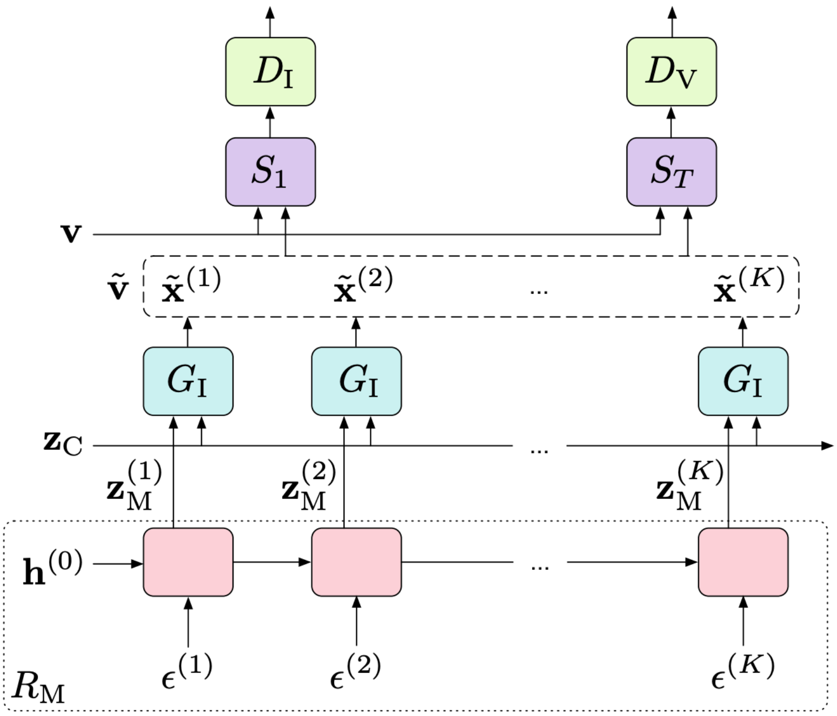

Figure 2: The MoCoGAN framework for video generation. For a video, the content vector, $\mathbf{z}_{\mathrm{C}}$, is sampled once and fixed. Then, a series of random variables $[\epsilon^{(1)}, ..., \epsilon^{(K)}]$ is sampled and mapped to a series of motion codes $[\mathbf{z}_{\mathrm{M}}^{(1)}, ..., \mathbf{z}_{\mathrm{M}}^{(K)}]$ via the recurrent neural network R_{M}. A generator G_{I} produces a frame, $\tilde{\mathbf{x}}^{(k)}$, using the content and the motion vectors $\{\mathbf{z}_{\mathrm{C}}, \mathbf{z}_{\mathrm{M}}^{(k)}\}$. The discriminators, D_{I} and D_{V}, are trained on real and fake images and videos, respectively, sampled from the training set $\mathbf{v}$ and the generated set $\tilde{\mathbf{v}}$. The function S_1 samples a single frame from a video, S_{T} samples T consequtive frames.