Controllable Image-to-Video Translation: A Case Study on Facial Expression Generation

Lijie Fan[1], **Wenbing Huang**[2], **Chuang Gan**[3], **Junzhou Huang**[2], **Boqing Gong**[2]
[1]Massachusetts Institute of Technology, [2]Tencent AI Lab, [3]MIT-Waston Lab
lijiefan@mit.edu, hwenbing@126.com, chuangg@mit.edu, jzhuang@uta.edu, boqinggo@outlook.com

Abstract

The recent advances in deep learning have made it possible to generate photo-realistic images by using neural networks and even to extrapolate video frames from an input video clip. In this paper, for the sake of both furthering this exploration and our own interest in a realistic application, we study image-to-video translation and particularly focus on the videos of facial expressions. This problem challenges the deep neural networks by another temporal dimension comparing to the image-to-image translation. Moreover, its single input image fails most existing video generation methods that rely on recurrent models. We propose a user-controllable approach so as to generate video clips of various lengths from a single face image. The lengths and types of the expressions are controlled by users. To this end, we design a novel neural network architecture that can incorporate the user input into its skip connections and propose several improvements to the adversarial training method for the neural network. Experiments and user studies verify the effectiveness of our approach. Especially, we would like to highlight that even for the face images in the wild (downloaded from the Web and the authors' own photos), our model can generate high-quality facial expression videos of which about 50% are labeled as real by Amazon Mechanical Turk workers.

Introduction

Upon observing the accomplishments of deep neural networks in a variety of subfields of AI, researchers have gained keen interests in pushing its boundaries forward. Among the new domains in which they have recently achieved remarkable results, photo-realistic image generation (Goodfellow et al. 2014; Karras et al. 2017) and image-to-image translation (Isola et al. 2016; Zhu et al. 2017) are two well-known examples — they were considered very difficult in general as the desired output is extremely high-dimensional, incurring the curse of dimensionality to conventional generative models. In this paper, for the sake of both furthering this exploration and our interest in a realistic application, we propose to study image-to-video translation which challenges the deep models by yet another temporal dimension. We focus on a special case study: how to generate video clips of rich facial expressions from a single profile photo of the neutral expression.

The image-to-video translation might seem like an ill-posed problem because the output has much more unknowns to fill in than the input values. Although there have been some works on video generation (Oh et al. 2015; Mathieu, Couprie, and LeCun 2015; Lotter, Kreiman, and Cox 2016; Villegas et al. 2017; Tulyakov et al. 2017; Liu et al. 2017; Vondrick and Torralba 2017), they usually take as input multiple video frames and then extrapolate the future from the recurrent pattern inferred from the input, preventing them from tackling the image-to-video translation whose input supplies no temporal cue at all. Moreover, it is especially difficult to generate satisfying video clips of facial expressions for the following two reasons. One is that humans are familiar with and sensitive about the facial expressions. Any artifacts, no matter in the spatial dimensions or along the temporal dimension, could be noticed by users. The other is that the face identity is supposed to be preserved in the generated video clips. In other words, the neural network cannot remember the faces seen in the training stage but instead learn the "imagination" capabilities so as to handle new faces in the deployment stage.

Despite the difficulties discussed above, we believe it is feasible to tackle the image-to-video translation at least in the particular domain of facial expression generation. First, different people express emotions in similar manners. For instance, one often opens its mouth when s/he becomes excited or surprised. Second, the expressions are often "unimodal" for a fixed type of emotion. In other words, there exists a procedure of gradual change from the neutral mode to the peak state of an expression. For instance, one increases her/his degree of happiness monotonically until s/he reaches the largest degree of expression. Third, the human face of a profile photo draws a majority of users' attention, leaving the quality of the generated background less important. All these characteristics significantly reduce the variability of the video frames, making the image-to-video translation plausible.

In this paper, we propose a user-controllable approach to the image-to-video translation. Given a single profile photo as input and a target expression (e.g., happiness), our model generates several video clips of various lengths. We allow users to conveniently control the length of a video clip by specifying an array of real numbers between 0 and 1. Each number indicates the expression degree (e.g., 0.6 out of 1) the corresponding frame is supposed to depict.

Moreover, our approach can generate a video frame of a particular degree of laughing, for example, without the need of rendering the frames before it. In contrast, most existing video generation methods (Oh et al. 2015; Mathieu, Couprie, and LeCun 2015; Lotter, Kreiman, and Cox 2016; Villegas et al. 2017; Tulyakov et al. 2017; Liu et al. 2017; Vondrick and Torralba 2017) cannot due to their recurrent generators. Two notable exceptions are (Xue et al. 2016) and (Hao, Huang, and Belongie 2018). However, their goals differ from ours; the former predicts the probabilistic future of the input while the latter takes as input both a video frame and sparse trajectories.

We design our deep neural network and the training losses in the following manner in order to achieve the aforementioned properties. The frame generator consists of three modules: a base encoder, a residual encoder, and a decoder taking as input from both encoders. We weigh the skip connections between the residual encoder and the decoder using the expression degrees supplied by users in the test stage. In the training stage, we infer the degrees by assigning 0 to the neutral expression frame, 1 to the frame of the peak expression, and then numbers between 0 and 1 to the frames in between in proportion to their distances to the neutral frame. We train our model following the practice of generative adversarial nets (Goodfellow et al. 2014) with the following improvements. Noting the importance of the mouth region in expressing emotions, we use a separate discriminator to take care of it. Besides, we regularize the change between adjacent frames to ensure smoothness along the temporal dimension. Finally, we augment the main task of frame generation by predicting the face landmarks.

Extensive experiments and user studies verify that the video clips generated by our approach are of superior quality over those by the competing methods. We would like to highlight that, by even inputting the face images in the wild (downloaded from the Web and the authors' own photos), our model can generate almost realistic facial expression videos, of which around 50% are labeled as real by Amazon Mechanical Turk workers.

Related Work

Image-to-Image Translation. Image-to-image translation has re-gained much attention due to the recent advances of deep generative models (Goodfellow et al. 2014). Earlier, researchers usually formulate this task as per-pixel classification or regression (Long, Shelhamer, and Darrell 2015), where the training loss conditioning on the input image is applied to each pixel such as conditional random fields (Chen et al. 2018) and nonparametric loss (Li and Wand 2016). More recent approaches apply the conditional GAN as a structured loss to penalize the joint configuration of the output, such as the Pixe2Pixel framework by (Isola et al. 2017). Subsequently, the translation between two unpaired domains is also studied as CycleGAN (Zhu et al. 2017) and the unsupervised domain adaption method in (Liu, Breuel, and Kautz 2017). Comparing with them, our image-to-video task is more challenging because the temporal dynamics have to be captured in our task.

Video Generation. Predicting the future may benefit many applications, such as learning feature representations (Goroshin, Mathieu, and LeCun 2015; Ranzato et al. 2014; Srivastava, Mansimov, and Salakhudinov 2015) and interactions (Finn, Goodfellow, and Levine 2016). Previous works on video generation can be roughly divided into two categories: unconditional video generation and video prediction. The first focuses mainly generates short video clips from random vectors sampled from a prior distribution (Vondrick, Pirsiavash, and Torralba 2016; Tulyakov et al. 2017). VGAN (Vondrick, Pirsiavash, and Torralba 2016) does this by separately generating the static background and the foreground. MoCoGAN (Tulyakov et al. 2017) decomposes the motion and content into two subspaces where the motion trajectory is learned by a Recurrent Neural Network (RNN). The second category, i.e., video prediction, aims at extrapolating or interpolating video frames from the observed frames (Oh et al. 2015; Mathieu, Couprie, and LeCun 2015; Lotter, Kreiman, and Cox 2016; Villegas et al. 2017). Early work focuses on small patches (Sutskever, Hinton, and Taylor 2009). Owing to the development of deep learning, recent approaches in video prediction have shifted from predicting patches to full frame prediction (Oh et al. 2015). For example, (Mathieu, Couprie, and LeCun 2015) proposed an adversarial loss for video prediction and a multi-scale network architecture that results in high quality prediction for a few time steps in natural video. Upon observing that the frame prediction quality by (Mathieu, Couprie, and LeCun 2015) degrades quickly, the HP method by (Villegas et al. 2017) generates the long-term feature frames by first learning the evolution of the high-level structure (*e.g.* the pose) with a RNN and then constructing the current image frame conditioned on the predicted high-level structure and a image in the pass. A more recent work by (Hao, Huang, and Belongie 2018) attempts to control the video prediction by using user-defined sparse trajectories. Our image-to-video translation is in the same vein as the video prediction, but we emphasize some of its unique characteristics. First, our task requires one single input image other than multiple video frames, opening the door for more potential applications. Second, unlike (Oh et al. 2015; Mathieu, Couprie, and LeCun 2015; Lotter, Kreiman, and Cox 2016; Villegas et al. 2017) where recurrent models are applied, our method can skip an arbitrary number of frames during inference and training.

Facial Attribute Manipulation. Several works (Shen and Liu 2017; Lu, Tai, and Tang 2017; Pumarola et al. 2018) have been conducted for facial images manipulation. The study by (Shen and Liu 2017) addresses the face attribute manipulation by modifying a face image according to attributes. The approach by (Lu, Tai, and Tang 2017) performs attribute-guided face image generation on unpaired image data. Since both of the above methods are mainly for static face generation, they are not naturally applicable for our task to generate continues videos of facial expressions. The method by (Pumarola et al. 2018) foucus on the task of generating facial expression videos. They applie a cycle-system for face generation where the action units are extracted by some off-line approach. Also, their pipeline requires an external tool to crop out the faces in the image. While in our appraoch, we

do not need perform any face cropping.

Approach

We first formalize the image-to-video translation problem and then describe our approach in detail.

Problem formulation

Given an input image $\boldsymbol{I} \in \mathbb{R}^{H\times W\times 3}$ where H and W are respectively the height and width of the image, our goal is to generate a sequence of video frames $\{\boldsymbol{V}(a) := f(\boldsymbol{I}, a); a \in [0, 1]\}$, where $f(\boldsymbol{I}, a)$ denotes the model to be learned. Note that the variable a, called an action variable, takes continuous values between 0 and 1, implying that there could be an infinite number of frames in the generated video clip. In practice, we allow users to give an arbitrary number of values to a and, for each of them, our model generates a frame. For simplicity, we use a separate model $f(\boldsymbol{I}, a)$ for each type of facial expressions to describe our approach.

We demand the following properties from the model $f(\boldsymbol{I}, a)$. It is supposed to reconstruct the input image when $a = 0$, *i.e.*, $f(\boldsymbol{I}, 0) = \boldsymbol{I}$. Besides, the function $f(\boldsymbol{I}, a)$ has to be smooth with respect to the input a. In other words, the generated video frames $\boldsymbol{V}(a)$ and $\boldsymbol{V}(a + \Delta a)$ should be visually similar when Δa is small. The larger a is, the bigger change the generated frame $\boldsymbol{V}(a)$ is from the original image $\boldsymbol{I}$. In the case of facial expression generation, we let $\boldsymbol{V}(1)$ be the peak state of the expression (e.g., the state when one's mouth opens to the most when s/he laughs).

The way we formalize the frame generator $f(\boldsymbol{I}, a)$ implies several advantages over the popular recurrent models for video generation. First, the generation process is controllable. One may control the total number of frames by supplying the proper number of values for the action variable a. One may also tune the position of the peak state of the expression in the video. For instance, an array of monotonically increasing values let the subject of the input image express his emotion from mild to strong, while a unimodal array like $\boldsymbol{a} = [0, 0.1, \cdots, 1, 0.9, \cdots, 0]$ makes the subject express to the most and then cool down. Besides, the frames to be generated are independent of each other, taxing less over the format of the training data — temporal smoothness is enforced by a regularization term. Finally, this model structure also benefits the optimization procedure because we do not need backpropagate gradients through time, avoiding the potential caveat of vanishing gradients.

Network design for the video frame generator

Figure 1 sketches the neural network modules we designed for the video frame generator $f(\boldsymbol{I}, a)$. It is mainly composed of three modules: a base encoder, a residual encoder, and a decoder. In addition, there are two two discriminators for the purpose of generative adversarial training. We employ the Convolution-BatchNorm-ReLu layers in these modules (Ioffe and Szegedy 2015).

Generator. Considering that $f(\boldsymbol{I}, 0) = I$, a straightforward construction of f is to linearly combine the input image with a residual term. However, it would incur severe artifacts to add the two in the pixel space. Instead, we perform linear aggregation in the feature space. Denote by $e_0(\cdot)$ and $e_1(\cdot)$ the base encoder and the residual encoder, respectively, where the former is to extract the feature hierarchy for self-reconstruction and the latter takes care of the change that is useful for constructing the future frames. Concretely, we have the following,

$$\boldsymbol{F}(\boldsymbol{I}, a) \quad = \quad e_0(\boldsymbol{I}) + a \cdot e_1(\boldsymbol{I}), \tag{1}$$

where the variate a explicitly determines the intensity of shift off the base encoder. Note that the summation in eq. 1 is layer by layer (cf. Figure 1). The resulting feature hierarchy $\boldsymbol{F}$ is then fed to the decoder d for video frame generation, *i.e.*,

$$\boldsymbol{V}(a) := f(\boldsymbol{I}, a) = d(\boldsymbol{F}(\boldsymbol{I}, a)) = d(e_0(\boldsymbol{I}) + a \cdot e_1(\boldsymbol{I})), \tag{2}$$

where the decoder mirrors the base encoder's architecture and takes as input the feature hierarchy in the reverse order (cf. Figure 1).

Discriminators. We use two discriminators for the purpose of adversarial training: a global discriminator D_g and a local one D_l. The global discriminator contrasts the generated video frame $\boldsymbol{V}(a)$ to the groundtruth frame. This is a standard and effective practice in video generation (Vondrick, Pirsiavash, and Torralba 2016; Tulyakov et al. 2017) and image-to-image translation (Isola et al. 2017). In addition, we employ a local discriminator to take special account of certain local parts of interest. Taking the smile expression for example, the mouth region is the most active part and deserves more detailed synthesis than the others. We first compute a mask as a convex closure of the detected facial landmarks around the subject's mouth and then filter out the mouth regions by the mask for both groundtruth and the generated frames. The local discriminator D_l is then applied to the filtered pairs.

Training loss

We prepare training data in the following manner. Given a video clip of length T, assume it has been labeled such that the 1st frame is in the neutral expression state and the T-th is at the peak of the expression. We assign coefficient $a = (t-1)/(T-1)$ to the t-th frame of this clip. Denote by $\boldsymbol{Y}(a)$ one of these groundtruth frames. We train our neural network using the adversarial loss

$$\mathcal{L}_g \quad := \quad -\log(1 - D_g(\boldsymbol{V}(a))) - \log D_g(\boldsymbol{Y}(a))), \tag{3}$$

for the global discriminator D_g and

$$\begin{aligned}\mathcal{L}_l \quad := \quad & -\log(1 - D_l\left(\boldsymbol{V}(a) \circ \boldsymbol{M}(a)\right)) \\ & -\log D_l\left(\boldsymbol{Y}(a) \circ \boldsymbol{M}(a)\right),\end{aligned} \tag{4}$$

for the local discriminator D_l, where $\boldsymbol{M}(a)$ is the mask to crop out the local patch of interest and $\circ$ denotes element-wise multiplication. Joining the previous work (Isola et al. 2017), we find that it is beneficial to augment the adversarial loss with the reconstruction error: $\mathcal{L}_r = \|\boldsymbol{Y}(a) - \boldsymbol{V}(a)\|_1$.

Temporal continuity. The generative adversarial training of the neural network may result in mode collapse (different modes collapse to a mixed mode that does not exist in the real data) and mode dropping (the generator fails to capture some of the modes). Whereas the reconstruction loss $\mathcal{L}_r$ alleviates these issues to some extent, it is defined at a

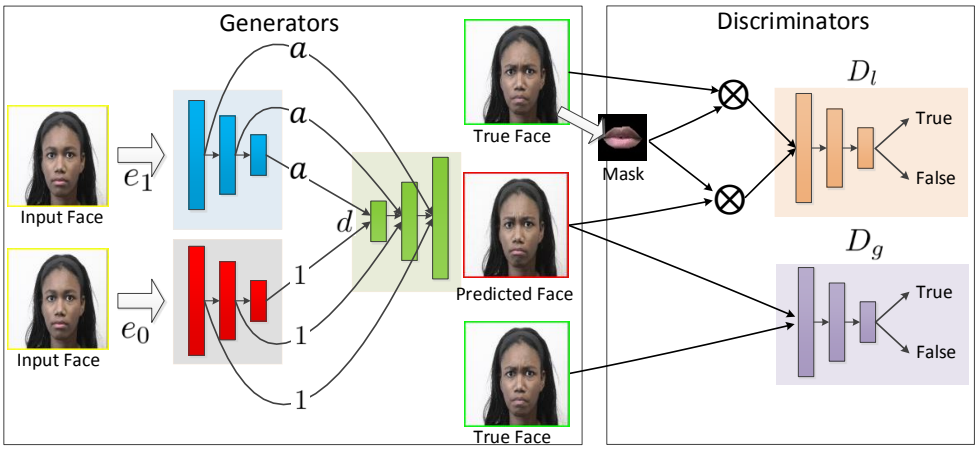

Figure 1: Illustration of our model. It consists of two encoders e_0 and e_1, one decoder d, and two discriminators D_l and D_g.

particular time step and does not track the temporal continuity in the video. We propose to regularize the difference between nearby video frames generated by the network. The regularization both helps prevent the mode dropping issue and makes the generated video clips smooth over time. It is defined as below,

$$\begin{aligned}\mathcal{R}_t &= \|\boldsymbol{V}(a) - \boldsymbol{V}(a-\Delta a)\|_1 + \\ &\quad \|\boldsymbol{V}(a) - \boldsymbol{V}(a+\Delta a)\|_1, \end{aligned} \tag{5}$$

where Δa is a small increment. The training is still frame-wise and efficient as the frames $\boldsymbol{V}(a)$, $\boldsymbol{V}(a+\Delta a)$, and $\boldsymbol{V}(a-\Delta a)$ are computed independently from each other.

Facial landmark prediction. As discussed earlier, we use facial landmarks to extract the local regions of interest for the local discriminator. In our experiments, we use the Dlib Library (King 2009) to detect 68 landmarks from any groundtruth video frame. These landmarks are supposed to be at the same locations for the correspondingly synthesized video frame. Therefore, we stack another 68-dimensional channel on the top of the second-to-last layer of the global decoder to predict the landmarks, enforcing the generator to provide details of the face. This loss is denoted by $\mathcal{L}_k := \|\bar{\boldsymbol{K}}(a) - \boldsymbol{K}(a)\|_2$, where $\bar{\boldsymbol{K}}$ and $\boldsymbol{K}(t)$ are the predicted and groundtruth landmarks, respectively.

Putting the above together, we train our neural networks by alternating between optimizing the generator and the discriminators in order to solve the following problem,

$$\min_{D_g, D_l} \max_{f} \sum_{a} \mathcal{L}_g + \mathcal{L}_l + \mathcal{L}_k + \mathcal{R}_t. \tag{6}$$

In the experiments, we use slightly different weights in front of the loss and regularization terms.

Jointly learning different facial expressions

Thus far, we have assumed a separate model $f(\boldsymbol{I}, a)$ for each type of facial expressions. It is straightforward to extend it to handle $n > 1$ types of emotions jointly:

$$\begin{aligned}\boldsymbol{V}(t) &= d(\boldsymbol{F}(\boldsymbol{I}, \boldsymbol{a}(t))), \quad \boldsymbol{F}(\boldsymbol{I}, \boldsymbol{a}(t)) \\ &= e_0(\boldsymbol{I}) + \sum_{i=1}^{n} a_i(t) \cdot e_i(\boldsymbol{I}), \end{aligned} \tag{7}$$

where $\boldsymbol{a}(t) \in [0,1]^n$ is an n-dimensional vector with each dimension standing for one emotion type. Since each training video clip contains one type of emotion, only one entry of the vector $\boldsymbol{a}(t)$ is non-zero in the training stage. At the test stage, however, we examine the effect of mixing some emotions by allowing non-zeros values in multiple entries of the vector $\boldsymbol{a}(t)$. Note that different types of emotions share the same base encoder e_0 (as well as the decoder d and discriminators D_g, D_l) and differ only by the residual encoders $e_1, e_2, \cdots, e_n$.

Experiments

Given a neutral face image and a target expression(*e.g.*, smile), we generate a video clip to simulate how the face will change towards the target expression. We not only use the public CK+ (Lucey et al. 2010) dataset for model training but also significantly extend it in scale. The new larger-scale dataset is named CK++. To better evaluate the performance of our method, we further collect around raw 150 face images from the Web. We then generate the facial expression videos based on these collected photos and submit them to the AMT.

CK+. The Extended Cohn-Kanade (CK+) dataset (Lucey et al. 2010) is a widely used dataset for facial emotion analysis. It contains 593 videos of 8 different emotion categories (including the neutral category) and 123 subjects. Each video frame is provided with a 68-point facial landmark label. We use three major categories (*i.e.*, "happy", "angry", and "surprised") in this paper.

CK++. Most images in CK+ are in the gray-scale. We augment CK+ by additionally collecting the facial expression videos in the RGB-scale. The videos are collected by a fixed camera from 65 volunteers consisting of 32 males and 33 females. Each volunteer is asked to perform each of the "happy", "angry" and "surprised" expressions for at least twice. We manually remove the redundant frames before the initial neutral state and after the stationary peak state. We also remove the videos that contain severe head movement or blurry faces. There are 214, 167, and 177 video clips for the "happy", "angry", and "surprised" expressions, respectively. On average, each clip has 21 frames. Finally, we use the Dlib

Figure 2: Visualization for the "happy" expression by different methods.

Library (King 2009) to detect 68 landmarks from each of the frames.

Implementation Details

For our encoders, we employ eight-downsampling-layer architectures with the Leaky-ReLu activation function. The decoder mirrors the encoder's architecture by eight upsampling layers and yet the ReLu activation function. Inspired by the U-Net (Ronneberger, Fischer, and Brox 2015), we further add skip connections between intermediate layers of the encoders and the decoder (cf. Figure 1). Both the global and local discriminators are constructed by concatenating 3 convolution layers.

We use 10 video clips from the CK++ dataset for validation and all the others for training. Our network is trained from scratch with all parameters normally initialized. For each training batch, we randomly sample a video clip and then use its first frame as the input image to train the network. All images are resized to 289x289 and randomly cropped to 256x256 before being fed into the network. The Adam optimizer is used in the experiments, with the initial learning rate of 0.0002. The whole training process takes 2100 epochs, where one epoch means a complete pass over the training data. As discussed in § , training the local discriminator requires a mask to crop the local regions of interest. Since mouth is the most expressive region, we crop it out by a convex closure of the landmarks around the mouth. We set the small increment to $\Delta a = 0.1$ for temporal regulation $\mathcal{R}_t$.

As our task of controllable image-to-video translation is new, there is no exactly related method in the literature. Conservatively, we adapt two previous methods to our experiments including Hierarchical Prediction (HP) (Villegas et al. 2017) and Convolution-LSTM (ConvLSTM) (Xingjian et al. 2015). In particular, we make the following changes to HP and ConvLSTM to fit them to our problem: 1) Since both HP and CovLSTM use LSTM to recursively generate video frames, we have to fix the length of the video sequence to be generated. We do so by uniformly sampling 10 frames per video clip. 2) We train a separate model for each target expression. 3) We replace the CovNet in ConvLSTM and the Visual-Structure in HP with U-Net because the results of their default architectures works not well.

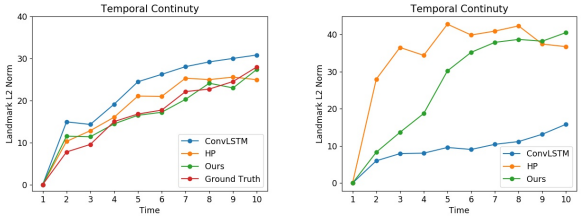

Figure 3: The L2 norm between the landmarks in each frame and the initial frame over time. The left and right sub-figures are on the training and validation samples, respectively.

Evaluations

Visualization. For a fair comparison, we let our method output the same number of frames as HP and ConvLSTM do (e.g., 10) by setting $a = \{0.1, 0.2, \cdots, 1\}$. Figure 2 displays the generated video frames of the "happy" emotion for two

Table 1: Comparisons of AMT results between our method against two baselines.

"Which video looks more realistic?"	Happy	Surprise	Angry	Mean
Prefers ours over ConvLSTM	83.8%	82.2%	83.0%	83.0%
Prefers ours over HP	77.3%	69.7%	67.1%	71.4%
Prefers ours over both baselines	69.9%	63.7 %	61.8%	65.1%

Table 2: AMT results on how many videos generated by our method can fool the workers.

"Is this video real?"	Happy	Surprise	Angry	Mean
Training Videos	64.5%	59.7%	57.3%	60.5%
Testing Videos	49.4%	52.2%	48.3%	49.9%

Table 3: SSIM evaluation results.

Methods	Happy	Surprise	Angry	Mean
ConvLSTM	0.880	0.893	0.898	0.890
HP	0.928	0.912	0.931	0.924
Ours	0.958	0.953	0.963	**0.953**

persons, one seen at training and the other unseen. We can see that both our method and the LSTM-based baseline models perform well on the training image. However, when it comes to the person of the validation set, our model clearly outperforms the baselines in terms of both image quality and temporal continuity.

Analysis on temporal continuity. To evaluate temporal continuity quantitatively, we extract facial landmarks from each generated video frame (*i.e.* a 68x2 dimensional vector) and then compute the L2 distance between the landmarks of each frame and those of the initial one. Figure 3 plots the distances versus the time steps on the same training and validation samples in Figure 2. We find that the curve of our approach aligns well with that of the groundtruth video frames. the face in the image sequence generated by ConvLSTM doesn't seem to move much, as the facial keypoints almost remain in the same place as the generation process goes. As for HP baseline, there is a sudden change between the generate frame and the initial frame, which means the expression generate by HP do not have a good temporal continuity. In fact, in our validation data, we see many cases where the person in the generated video by HP seem to smile a little, go back, and then smile again. We expect to avoid this phenomenon in our task. While for our proposed model, the L2 norm between keypoints in each frame and the initial frame grows steadily and linearly without decreasing, showing the images generated by our method have a good temporal continuity.

AMT results. Following (Villegas et al. 2017), we also conduct user studies to compare the results of different methods. For this purpose, we formulate test data by downloading about 150 face images from the Web without any post-processing. We then generate the facial expression videos of "happy", "surprise", and "angry" by our approach and two baselines. We pair the video clip of the same input image for the same emotion by our method with that by either of the baselines, and then ask an AMT worker to choose which one is more realistic in terms of the temporal continuity, image quality, naturalness of the expressions, etc. For each face image pair, we ask 50 AMT worker to answer the questions.

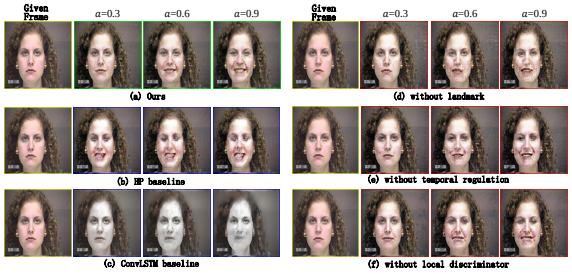

Figure 4: Ablation studies on our method.

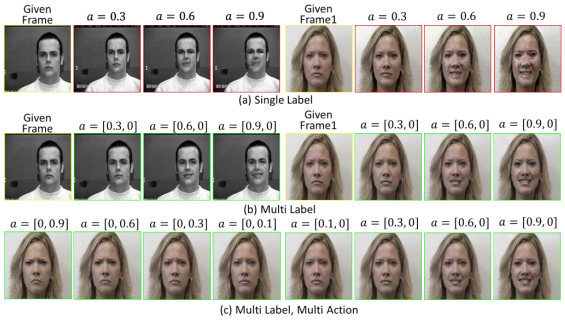

Figure 5: Evaluation on multiple-label models. (a) Videos generated by single-label models; (b) Videos generated by multi-label models; (c) Transferring the "angry" expression to the "happy" one by controlling the action variate in the multi-label model.

After the user study, a preference rate is calculated for each user and the final results are averaged over the 50 scores. The rows above the last in Table 1 show that the users prefer our results to either of the baselines' to a large margin. We also ask users to choose the most realistic clip from three, respectively generated by our model and the two baselines. As shown in the last row of Table 1, our results are again selected significantly more often than the other two. In addition, we perform a more challenging test by mixing the simulated videos by our method with real videos, and then asking an AMT worker to judge if the displayed video is real or not. As reported in Table 2, the results are encouraging, as nearly 50% of our generated videos from the test faces are labeled as real by AMT workers.

Quantitive Results We also quantitively evaluate our method and the baseline appraoches. We computed the SSIM scores for the generated videos of the test faces in CK++. The results are shown in Table 3. There is a notable margin between ours and the competing baselines. We have

also conducted experiments with MocoGAN(Tulyakov et al. 2017) and GANimation appraoch(Pumarola et al. 2018) on the "Happy" emotion. We train these two methods using the CK++ training videos under the same experiment settings as ours. MoCoGAN gives rise to blurry images and the SSIM score drops from 0.958 (ours) to 0.853 (MoCoGAN). This could attribute to the training inefficiency of MoCoGan and its LSTM-based architecture. For GANimation, we first use OpenFace to extract the action units for each image and then train the model with the faces and their associated action units. Unlike the real videos or those generated by other methods, the expressions generated by GANimation change very subtly over time. GANimation gives rise to a SSIM score of 0.930, whereas ours is 0.958. We conjecture that two causes make GANimation less effective than ours. First, in its original paper, GANimation requires a detector to localize and crop the faces. Both training and testing are run on these cropped faces. While in our experiments, we do not need face cropping and our method still delivers promising results. Second, compared to GANimation, our model additionally takes the temporal smooth into account (Eq.(5)), which makes the generated videos more smooth temporally. We will include the above results in the main text.

Ablation Studies. We have run some ablation studies to examine some key components of our approach, as illustrated in Figure 4. We implement several variants of our method without the local discriminator, without predicting the landmarks, and without the temporal continuity regularization. **I.** People easily focus on mouths when they first see a video. So a local discriminator on the mouth would make the video seem more realistic to audience. Without local discriminator, Figure 4 (f) easily involves blurring artifact compared to the original model. **II.** Landmark prediction gives a higher level regulation, which can enable our model to have the ability to generate facial feature in the right place, avoiding generating multiple features in the same image, therefore avoid blurring artifact and make the generate image more clear and reasonable. **III.** We can see from the example in Figure 4 (e), temporal regulation not only force the movement perform continually, avoid sudden change, but also have the effect to make the generated image more clear.

Controlling Action Variable. One of the most interesting part in our approach is that we can control the lengths of the videos by the action variable a. We provide demos on controlling the action variable in the supplementary materials.

Jointly model multiple types of expressions. We present in § that our model is applicable for learning different types of emotions simultaneously. As a result, we may mix different emotions by providing more than one non-zero entries to the vector $\boldsymbol{a}(t)$. We first show that it gives rise to better results to simultaneously model different types of expressions in one neural network than learning a separate model for each. To show this, we further formulate a training set by using the faces of the "happy" and "angry" emotions. Each training person has either emotion but not both. For example, the two persons in Figure 5 have only the "angry" expression. Figure 5 (a) and (b) demonstrate that the generated "happy" videos by the jointly modeling the two emotions is more realistic than the models of individual emotions. We conjecture that it is due to the strong correlation between emotions that enables information sharing between the residual encoders.

Another interesting application for the joint model is that it can transfer between two different emotions by using proper values of the action variable, as illustrated in Figure 5 (c). More results are included in the supplementary materials.

Conclusion

In this paper, we study image-to-video translation with a special focus on the facial expression videos. We propose a user-controllable approach so as to generate video clips of various lengths and different target expressions from a single face image. Both the lengths and types of the expressions can be controlled by users. To this end, we design a novel neural network architecture that can incorporate the user input and also propose several improvements to the adversarial training method for the neural networks. Experiments and user studies verify the effectiveness of our approach. It would be interesting to investigate the image-to-video translation in domains other than the facial expressions in the future work. In addition, we will explore the potential of progressive training (Karras et al. 2017) for generating higher-definition video clips from a single input image.

References

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2018. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence* 40(4):834–848.

Finn, C.; Goodfellow, I.; and Levine, S. 2016. Unsupervised learning for physical interaction through video prediction. In *Advances in neural information processing systems*, 64–72.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Goroshin, R.; Mathieu, M. F.; and LeCun, Y. 2015. Learning to linearize under uncertainty. In *Advances in Neural Information Processing Systems*, 1234–1242.

Hao, Z.; Huang, X.; and Belongie, S. 2018. Controllable video generation with sparse trajectories.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2016. Image-to-image translation with conditional adversarial networks. arxiv (2016). *arXiv preprint arXiv:1611.07004*.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. *arXiv preprint*.

Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

King, D. E. 2009. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research* 10:1755–1758.

Li, C., and Wand, M. 2016. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2479–2486.

Liu, Z.; Yeh, R.; Tang, X.; Liu, Y.; and Agarwala, A. 2017. Video frame synthesis using deep voxel flow. In *International Conference on Computer Vision (ICCV)*, volume 2.

Liu, M.-Y.; Breuel, T.; and Kautz, J. 2017. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, 700–708.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440.

Lotter, W.; Kreiman, G.; and Cox, D. 2016. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*.

Lu, Y.; Tai, Y.-W.; and Tang, C.-K. 2017. Conditional cyclegan for attribute guided face image generation. *arXiv preprint arXiv:1705.09966*.

Lucey, P.; Cohn, J. F.; Kanade, T.; Saragih, J.; Ambadar, Z.; and Matthews, I. 2010. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, 94–101. IEEE.

Mathieu, M.; Couprie, C.; and LeCun, Y. 2015. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*.

Oh, J.; Guo, X.; Lee, H.; Lewis, R. L.; and Singh, S. 2015. Action-conditional video prediction using deep networks in atari games. In *Advances in Neural Information Processing Systems*, 2863–2871.

Pumarola, A.; Agudo, A.; Martinez, A.; Sanfeliu, A.; and Moreno-Noguer, F. 2018. Ganimation: Anatomically-aware facial animation from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Ranzato, M.; Szlam, A.; Bruna, J.; Mathieu, M.; Collobert, R.; and Chopra, S. 2014. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241. Springer.

Shen, W., and Liu, R. 2017. Learning residual images for face attribute manipulation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1225–1233. IEEE.

Srivastava, N.; Mansimov, E.; and Salakhudinov, R. 2015. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, 843–852.

Sutskever, I.; Hinton, G. E.; and Taylor, G. W. 2009. The recurrent temporal restricted boltzmann machine. In *Advances in Neural Information Processing Systems*, 1601–1608.

Tulyakov, S.; Liu, M.-Y.; Yang, X.; and Kautz, J. 2017. Mocogan: Decomposing motion and content for video generation. *arXiv preprint arXiv:1707.04993*.

Villegas, R.; Yang, J.; Zou, Y.; Sohn, S.; Lin, X.; and Lee, H. 2017. Learning to generate long-term future via hierarchical prediction. *arXiv preprint arXiv:1704.05831*.

Vondrick, C., and Torralba, A. 2017. Generating the future with adversarial transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Vondrick, C.; Pirsiavash, H.; and Torralba, A. 2016. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, 613–621.

Xingjian, S.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, 802–810.

Xue, T.; Wu, J.; Bouman, K.; and Freeman, B. 2016. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances in Neural Information Processing Systems*, 91–99.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*.