Generating Videos with Scene Dynamics

Carl Vondrick, Hamed Pirsiavash, Antonio Torralba
NIPS 2016
Presented by: Sai Bi
Outline

- Introduction
- Contributions
- Related work
- Algorithms
- Experiments
- Weakness
- Possible extensions
Introduction

✧ Understanding object motions and scene dynamics.
✧ Modeling how scenes transform is necessary for many video tasks.
✧ The creation of such a model is challenging and complex.
✧ Labeling data is expensive and ambiguous.
Contributions

- Propose an approach to generate tiny videos that have fairly dynamics.
- Leant he model directly form large amount of unlabeled data.
- Extend generative adversarial networks (GAN) to videos.
- Introduce two-stream generative model.
Related work

- Generative adversarial networks
  - Generator: takes noise as input and generate samples
  - Discriminator: distinguish between samples from both generator and training data
  - Applications: image synthesis, image super-resolution
Related work

- Predict the future in video
  - Generative model conditioned on the past frames.
  - *Unsupervised Learning for Physical Interaction through Video Prediction*, Finn et al., NIPS 2016
Related work

- Action recognition from videos
  - *Learning Spatiotemporal Features with 3D Convolutional Networks*, Tran et al., ICCV 2015
  - *Two-Stream Convolutional Networks for Action Recognition in Videos*, Simonyan et al., NIPS 2014
Algorithms

- Generative models for videos
  - Generative networks $G$: produces the videos.
  - Discriminator $D$: distinguish between real videos and fake videos.
Generator

- Network design principles
  - Invariant to translations in both space and time.
  - Stationary camera, only objects move.
- Two different architectures
  - One-stream generator
  - Two-stream generator
One-stream generator

- One-stream architecture
  - Input: 100 dimension random noise
  - Output: 64×64 videos for 32 frames
  - Architecture: DC-GAN
One-stream generator

- **DC-GAN**
  - Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
  - Use BatchNorm in both the generator and the discriminator.
  - Remove fully connected hidden layers for deeper architectures.
  - Use ReLU activation in generator for all layers except for the output, which uses Tanh.
  - Use LeakyReLU activation in the discriminator for all layers.

- **Extend to videos:**
  - First layer: $2 \times 4 \times 4$ (time x width x height), no padding
  - Other layers: $4 \times 4 \times 4$, 1 zero padding
Two-stream generator

- Multiple-branch generator
  - Foreground branch
  - Foreground mask branch
  - Background branch
 Discriminator

- Classify realistic scenes from synthetically generated scenes.
- Recognize realistic motion between frames.
- Network architecture: reverse the foreground stream in the generator.
- Loss function: cross entropy loss
Implementations

- Alternate between optimizing generator and discriminator.
- Trained from scratch.
- Adam optimizer, learning rate = 0.0002, momentum = 0.5.
- Latent code: 100 dims, sampled from a normal distribution.
- Batch size: 64.
- Initialize all weights with zero mean Gaussian noise with standard deviation of 0.01.
- Normalize all videos to be in range of \([-1, 1]\)
Training data

- Two million videos from Flickr
- Unfiltered unlabeled videos
  - Over 5000 hours.
  - For representation learning.
- Filtered unlabeled videos
  - For video generation.
  - Use Places2 model to classify the videos by scene category.
  - Experiment with four scene categories: golf course, baby, beaches, train station.
Training data

◇ Video stabilization
  ◇ Reduce the effect of camera shake.
  ◇ Extract SIFT keypoints
  ◇ Use RANSAC to estimate a homography (rotation, translation, scale) between adjacent frames.
  ◇ Warp frames to minimize background motion.
  ◇ If the homography has too large re-projection error, remove the segment from training data.
Video generation results
Video generation results

- One-stream and two-stream generator were evaluated.
- For each scene category, a generator is trained.
- The generated scenes tend to be fairly shapre, and the motion patterns are generally correct.
- Failure mode
  - Objects lack resolution, details missing.
Video generation results: beach
Video generation results: golf
Video generation results: train station
Video generation results: baby
Video generation result

- **Baseline: variational auto-encoder (VAE)**
  - Encoder: similar to the discriminator network, producing 100 dimension code
  - Decoder: follows the two-stream generator network
  - Feed examples through the encoder
  - Fit a Gaussian Mixture Model (GMM) with 256 components over the 100 dimensional hidden space.
  - To generative a novel video, sample from GMM, feed the sample through the decoder.
Quantitative evaluations

- Ask workers on Amazon Mechanical Turk to compare generated videos.
- 13,000 opinions across 150 unique workers.

<table>
<thead>
<tr>
<th>“Which video is more realistic?”</th>
<th>Percentage of Trials</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Golf</td>
</tr>
<tr>
<td>Random Preference</td>
<td>50</td>
</tr>
<tr>
<td>Prefer VGAN Two Stream over Autoencoder</td>
<td>88</td>
</tr>
<tr>
<td>Prefer VGAN One Stream over Autoencoder</td>
<td>85</td>
</tr>
<tr>
<td>Prefer VGAN Two Stream over VGAN One Stream</td>
<td>55</td>
</tr>
<tr>
<td>Prefer VGAN Two Stream over Real</td>
<td>21</td>
</tr>
<tr>
<td>Prefer VGAN One Stream over Real</td>
<td>17</td>
</tr>
<tr>
<td>Prefer Autoencoder over Real</td>
<td>4</td>
</tr>
</tbody>
</table>
Video representation learning

- Using the proposed model to learn unsupervised representations for videos.
  - Train two-stream model on 5000 hours of unfiltered, unlabeled videos
  - Find-tune the discriminator on action recognition using a small set of labeled videos.
  - Replace the last layer with a K-way softmax classifier.
  - Add dropout to the penultimate layer to reduce overfitting.
Action classification

- Evaluation performance on classifying actions on UCF101
- Initialize the network with the weights learnt from the generative adversarial network.
- Fine-tune the network for action classification.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chance</td>
<td>0.9%</td>
</tr>
<tr>
<td>STIP Features [35]</td>
<td>43.9%</td>
</tr>
<tr>
<td>Temporal Coherence [10]</td>
<td>45.4%</td>
</tr>
<tr>
<td>Shuffle and Learn [24]</td>
<td>50.2%</td>
</tr>
<tr>
<td>VGAN + Random Init</td>
<td>36.7%</td>
</tr>
<tr>
<td>VGAN + Logistic Reg</td>
<td>49.3%</td>
</tr>
<tr>
<td>VGAN + Fine Tune</td>
<td>52.1%</td>
</tr>
<tr>
<td>ImageNet Supervision [46]</td>
<td>91.4%</td>
</tr>
</tbody>
</table>

(a) Accuracy with Unsupervised Methods

(b) Performance vs # Data

(c) Relative Gain vs # Data
Visualizing representations

(a) hidden unit that fires on “person”

(b) hidden unit that fires on “train tracks”
Future generation

- Predict the future of a static image
  - Given a static image, extrapolate a video of possible consequent frames.
Future generation results

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image 1" /></td>
<td><img src="image2.png" alt="Image 2" /></td>
</tr>
<tr>
<td><img src="image3.png" alt="Image 3" /></td>
<td><img src="image4.png" alt="Image 4" /></td>
</tr>
<tr>
<td><img src="image5.png" alt="Image 5" /></td>
<td><img src="image6.png" alt="Image 6" /></td>
</tr>
<tr>
<td><img src="image7.png" alt="Image 7" /></td>
<td><img src="image8.png" alt="Image 8" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image9.png" alt="Image 9" /></td>
<td><img src="image10.png" alt="Image 10" /></td>
</tr>
<tr>
<td><img src="image11.png" alt="Image 11" /></td>
<td><img src="image12.png" alt="Image 12" /></td>
</tr>
<tr>
<td><img src="image13.png" alt="Image 13" /></td>
<td><img src="image14.png" alt="Image 14" /></td>
</tr>
<tr>
<td><img src="image15.png" alt="Image 15" /></td>
<td><img src="image16.png" alt="Image 16" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image17.png" alt="Image 17" /></td>
<td><img src="image18.png" alt="Image 18" /></td>
</tr>
<tr>
<td><img src="image19.png" alt="Image 19" /></td>
<td><img src="image20.png" alt="Image 20" /></td>
</tr>
<tr>
<td><img src="image21.png" alt="Image 21" /></td>
<td><img src="image22.png" alt="Image 22" /></td>
</tr>
<tr>
<td><img src="image23.png" alt="Image 23" /></td>
<td><img src="image24.png" alt="Image 24" /></td>
</tr>
</tbody>
</table>
Weakness

✧ The generations are usually distinguishable from real videos.
✧ Generated video are of low resolution and short length.
✧ Lack of robust evaluation of generative models.
✧ A generator is trained for each category of scenes, which greatly limits the usage of such a model.
✧ The future extrapolations do not match the given image very well.
Extensions

- Increase resolution of generated videos:
  - Progressive Growing of GANs for Improved Quality, Stability, and Variation, Karras et al., ICLR 2018
Extensions

✧ Flow based future prediction

✧ *Unsupervised Learning of Long-Term Motion Dynamics for Videos*, Luo et al., CVPR 2017
Extensions

- Separate generator for temporal and spatial
- *Temporal Generative Adversarial Nets with Singular Value Clipping, Saito et al., ICCV 2017*
Thank you