Synthesizing Dynamic Textures and Sounds by Spatial-Temporal Generative ConvNet

Jianwen Xie, Song-Chun Zhu, Ying Nian Wu
Department of Statistics
University of California, Los Angeles
jianwen@ucla.edu, sczhu@stat.ucla.edu, ywu@stat.ucla.edu

Abstract

Dynamic textures are spatial-temporal processes that exhibit statistical stationarity or stochastic repetitiveness in the temporal dimension. In this paper, we study the problem of modeling and synthesizing dynamic textures using a generative version of the convolution neural network (ConvNet or CNN) that consists of multiple layers of spatial-temporal filters to capture the spatial-temporal patterns in the dynamic textures. We show that such spatial-temporal generative ConvNet can synthesize realistic dynamic textures. We also apply the temporal generative ConvNet to the one-dimensional sound data, and show that the model can synthesize realistic natural and man-made sounds. The videos and sounds can be found at http://www.stat.ucla.edu/~jxie/STGConvNet/STGConvNet.html

1 Introduction

The notion of dynamic textures was first formally defined by Doretto et al. (2003) by extending the notion of textures from the spatial domain to the temporal or dynamic domain. Roughly speaking, dynamic textures refer to any spatial-temporal processes that exhibit statistical stationarity or stochastic repetitiveness in the temporal dimension. The processes may also exhibit statistical stationarity in the spatial domain, although this is unnecessary for the processes to be qualified as dynamic textures. Synthesizing and analyzing dynamic textures has been an interesting research problem. In this paper, we focus on the problem of synthesizing dynamic textures using a generative version of convolution neural network (ConvNet or CNN). We shall also study the problem of synthesizing natural and man-made sounds by treating them as one-dimensional dynamic textures in the temporal domain.

The ConvNet (LeCun et al., 1998; Krizhevsky et al., 2012) has proven to be an immensely successful discriminative learning machine. The convolution operation in the ConvNet is particularly suited for signals such as images, videos and sounds that exhibit translation invariance either in the spatial domain or the temporal domain or both. Recently, researchers have become increasingly interested in the generative aspects of ConvNet, for the purpose of visualizing the knowledge learned by the ConvNet, or synthesizing realistic images, or developing generative models that can be used for unsupervised learning.

In terms of synthesis, various approaches based on the ConvNet have been proposed to synthesize realistic static images (Dosovitskiy et al., 2015; Gregor et al., 2015; Denton et al., 2015; Gatys et al., 2015; Kulkarni et al., 2015; Lu et al., 2016). However, there has not been much work in the literature on synthesizing dynamic textures based on the ConvNet, and this is the focus of the present paper.

Specifically, we propose to synthesize dynamic textures by generalizing the generative ConvNet model recently proposed by Xie et al. (2016). The generative ConvNet can be derived from the discriminative ConvNet by assuming a base category such as Gaussian white noise. It is a random field model or an energy-based model (LeCun et al., 2006; Ngiam et al., 2011) that is in the form of exponential tilting of a reference distribution that corresponds to the base category. Specifically, the reference distribution is taken to be the Gaussian white noise model. The exponential tilting is parametrized by the ConvNet that involves multiple layers of linear filters and rectified linear units (ReLU) (Krizhevsky et al., 2012) , which seeks to capture non-Gaussian patterns or features. The energy function of the model is thus a combination of a ConvNet term that is piecewise linear due to the ReLU non-linearity (Montufar et al., 2014) and the ℓ_2 norm of the signal that comes from the Gaussian white noise. As a result, the energy function is piecewise quadratic, and the generative ConvNet model is piecewise Gaussian. Interestingly, the mean of each Gaussian piece is generated by a top-down process that involves multiple layers of binary activation variables, with the multiple layers of filters of the ConvNet serving as the basis functions for top-down generation. Such an explicit representation is unique among energy-based models (LeCun et al., 2006; Ngiam et al., 2011), and is a result of the fusion between the ReLU piecewise linear structure and the ℓ_2 norm term from the Gaussian white noise reference distribution.

The generative ConvNet can be sampled by the Langevin dynamics. Because of the aforementioned representational structure of the generative ConvNet, the Langevin dynamics is driven by the reconstruction error, i.e., the difference between the current sample and its reconstruction by the mean of the Gaussian piece. The model can be learned by the stochastic gradient algorithm (Younes, 1999). It is an "analysis by synthesis" scheme that seeks to match the synthesized images generated by the Langevin dynamics to the observed training images. Xie et al. (2016) show that the learning algorithm can synthesize realistic spatial image patterns such as textures and objects.

In this article, we generalize the spatial generative ConvNet by adding the temporal dimension, so that the resulting ConvNet consists of multiple layers of spatial-temporal filters that seek to capture spatial-temporal patterns at various spatial and temporal scales in the dynamic textures. These spatial-temporal filters are convolutional in the temporal domain, reflecting the statistical stationarity or stochastic repetitiveness of the dynamic textures in the temporal domain. If the dynamic textures also exhibit spatial stationarity, we also make the spatial-temporal filters convolutional in the spatial domain. Otherwise, the top-layer spatial temporal filters will be fully connected in the spatial domain at the top layer. We show that the learning algorithm for training the spatial-temporal generative ConvNet can synthesize realistic dynamic textures. We also apply a simple temporal generative ConvNet to the one-dimensional sound data, and we show that the model can generate realistic natural and man-made sounds.

2 Related work

Our work is a generalization of the generative ConvNet model of Xie et al. (2016) by adding the temporal dimension. See also Dai et al. (2015). They did not work on stationary dynamic patterns such as those in the video and sound data.

Dynamic textures were originally modeled by Doretto et al. (2003) using a vector auto-regressive model coupled with frame-wise dimension reduction by single value decomposition. Their model is a linear model with Gaussian innovations. In contrast, the spatial-temporal generative ConvNet is a non-linear and non-Gaussian model and is expected to be more flexible for capturing complex spatial-temporal patterns in dynamic textures with multiple layers of non-linear spatial-temporal filters.

The spatial-temporal discriminative ConvNet was used by Ji et al. (2013) for analyzing video data. We are not aware of prior work on generative version of spatial-temporal ConvNet.

For temporal data, a popular model is the recurrent neural network (Williams & Zipser, 1989; Hochreiter & Schmidhuber, 1997). It is a causal model and it requires a starting frame. In contrast, our model is non-causal, and does not require a starting frame. Compared to the recurrent network, our model is more convenient and direct in capturing temporal patterns at multiple time scales.

3 Spatial-temporal generative ConvNet for stationary dynamic patterns

3.1 Spatial-temporal filters

To fix notation, let $\mathbf{I}(x, t)$ be an image sequence of a video defined on the square (or rectangular) image domain $\mathcal{D}$ and the time domain $\mathcal{T} = (1, ..., T)$, where $x = (x_1, x_2)$ indexes the coordinates of pixels, and t indexes the frames in the video sequence. We can treat $\mathbf{I}(x, t)$ as a three dimensional function defined on $\mathcal{D} \times \mathcal{T}$. For a spatial-temporal filter F, we let $F * \mathbf{I}$ denote the filtered image sequence or feature map, and let $[F * \mathbf{I}](x, t)$ denote the filter response or feature at pixel x and time t.

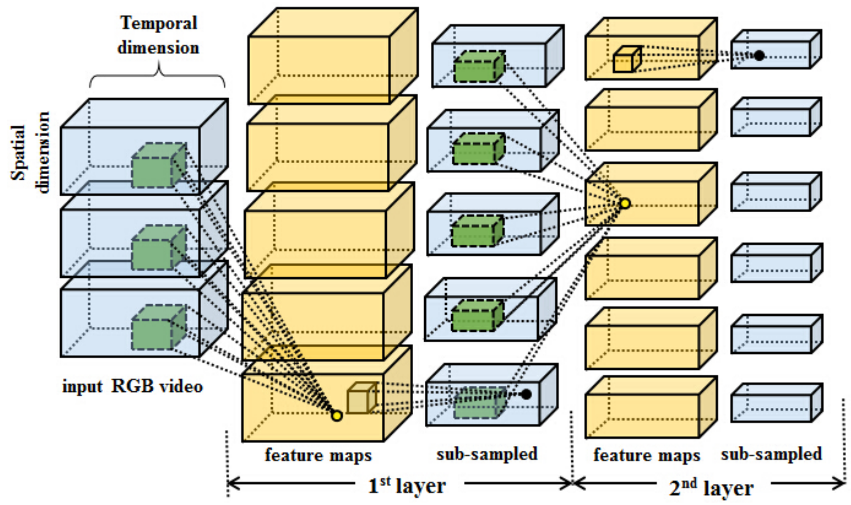


Figure 1: Spatial-temporal generative ConvNet consists of multiple layers of spatial-temporal filtering and sub-sampling operations for bottom-up feature extraction, resulting in multiple layers of feature maps (in yellow) and their sub-sampled versions (in blue). This figure illustrates an example of 2 layer network, where green cubes represent spatial-temporal filters. No pooling layers are used, but spatially fully connected layers are allowed.

The spatial-temporal ConvNet is a composition of multiple layers of linear filtering and ReLU non-linearity, as expressed by the following recursive formula:

$$[F_k^{(l)} * \mathbf{I}](x,t) = h\left(\sum_{i=1}^{N_{l-1}} \sum_{(y,s)\in\mathcal{S}_l} w_{i,y,s}^{(l,k)} [F_i^{(l-1)} * \mathbf{I}](x+y, t+s) + b_{l,k}\right), \tag{1}$$

where $l \in \{1, 2, ..., L\}$ indexes the layers. $\{F_k^{(l)}, k = 1, ..., N_l\}$ are the filters at layer l, and $\{F_i^{(l-1)}, i = 1, ..., N_{l-1}\}$ are the filters at layer $l-1$. k and i are used to index filters at layers l and $l-1$ respectively, and N_l and N_{l-1} are the numbers of filters at layers l and $l-1$ respectively. The filters are locally supported, so the range of (y, s) is within a local support $\mathcal{S}_l$ (such as a $7 \times 7 \times 3$ box of image sequence). The weight parameters $(w_{i,y,s}^{(l,k)}, (y,s) \in \mathcal{S}_l, i = 1, ..., N_{l-1})$ defines a linear filter that operates on $(F_i^{(l-1)} * \mathbf{I}, i = 1, ..., N_{l-1})$. The linear filtering operation is followed by ReLU $h(r) = \max(0, r)$. At the bottom layer, $[F_k^{(0)} * \mathbf{I}](x,t) = \mathbf{I}_k(x,t)$, where $k \in \{\mathrm{R}, \mathrm{G}, \mathrm{B}\}$ indexes the three color channels. Sub-sampling may be implemented so that in $[F_k^{(l)} * \mathbf{I}](x,t)$, $x \in \mathcal{D}_l \subset \mathcal{D}$, and $t \in \mathcal{T}_l \subset \mathcal{T}$. See Figure 1 for an illustration.

The spatial-temporal filters at multiple layers are expected to capture the spatial-temporal patterns at multiple scales in the dynamic textures. It is possible that the top-layer filters are fully connected in the spatial domain (i.e., the filtered images are 1×1 in the spatial domain) if the dynamic textures do not exhibit spatial stationarity.

3.2 Spatial-temporal generative ConvNet

The spatial-temporal generative ConvNet is a random field model defined on the image sequence $\mathbf{I} = (\mathbf{I}(x,t), x \in \mathcal{D}, t \in \mathcal{T})$. It is in the form of exponential tilting of a reference distribution $q(\mathbf{I})$:

$$p(\mathbf{I}; w) = \frac{1}{Z(w)} \exp\left[f(\mathbf{I}; w)\right] q(\mathbf{I}), \tag{2}$$

where the scoring function $f(\mathbf{I}; w)$ is

$$f(\mathbf{I}; w) = \sum_{k=1}^{K} \sum_{x\in\mathcal{D}_L} \sum_{t\in\mathcal{T}_L} [F_k^{(L)} * \mathbf{I}](x,t), \tag{3}$$

where w consists of all the weight and bias terms that define the filters $(F_k^{(L)}, k = 1, ..., K = N_L)$ at layer L, and q is the Gaussian white noise model, i.e.,

$$q(\mathbf{I}) = \frac{1}{(2\pi\sigma^2)^{|\mathcal{D}\times\mathcal{T}|/2}} \exp\left[-\frac{1}{2\sigma^2}||\mathbf{I}||^2\right], \tag{4}$$

where $|\mathcal{D}\times\mathcal{T}|$ counts the number of pixels in the domain $\mathcal{D}\times\mathcal{T}$. Without loss of generality, we shall assume $\sigma^2 = 1$.

The scoring function $f(\mathbf{I}; w)$ in (3) tilts the Gaussian reference distribution into a non-Gaussian model. In fact, the purpose of $f(\mathbf{I}; w)$ is to identify the non-Gaussian spatial-temporal features or patterns. In the definition of $f(\mathbf{I}; w)$ in (3), we sum over the filter responses at the top layer L over all the filters, positions and times. The spatial and temporal pooling reflects the fact that we assume the model is stationary in spatial and temporal domains. If the dynamic texture is non-stationary in the spatial domain, then the top layer filters $F_k^{(L)}$ are fully connected in the spatial domain, i.e., $\mathcal{D}_L$ is 1×1.

A simple but consequential property of the ReLU non-linearity is that $h(r) = \max(0, r) = 1(r > 0)r$, where $1()$ is the indicator function, so that $1(r > 0) = 1$ if $r > 0$ and 0 otherwise. As a result, the scoring function $f(\mathbf{I}; w)$ is piecewise linear (Montufar et al., 2014), and each linear piece is defined by the multiple layers of binary activation variables $\delta_{k,x,t}^{(l)}(\mathbf{I}; w) = 1\left([F_k^{(l)} * \mathbf{I}](x, t) > 0\right)$, which tells us whether a local spatial-temporal pattern represented by the k-th filter at layer l, $F_k^{(l)}$, is detected at position x and time t. Let $\delta(\mathbf{I}; w) = \left(\delta_{k,x,t}^{(l)}(\mathbf{I}; w), \forall l, k, x, t\right)$ be the activation pattern of $\mathbf{I}$. Then $\delta(\mathbf{I}; w)$ divides the image space into a large number of pieces according to the value of $\delta(\mathbf{I}; w)$. On each piece of image space with fixed $\delta(\mathbf{I}; w)$, the scoring function $f(\mathbf{I}; w)$ is linear, i.e.,

$$f(\mathbf{I}; w) = a_{w,\delta(\mathbf{I};w)} + \langle \mathbf{I}, B_{w,\delta(\mathbf{I};w)} \rangle, \tag{5}$$

where both a and B are defined by $\delta(\mathbf{I}; w)$ and w. In fact, $B = \partial f(\mathbf{I}; w)/\partial \mathbf{I}$, and can be computed by back-propagation, with $h'(r) = 1(r > 0)$. The back-propagation process defines a top-down deconvolution process (Zeiler et al., 2011), where the filters at multiple layers becomes the basis functions at those layers, and the activation variables at different layers in $\delta(\mathbf{I}; w)$ become the coefficients of the basis functions in the top-down deconvolution.

$p(\mathbf{I}; w)$ in (2) is an energy-based model (LeCun et al., 2006; Ngiam et al., 2011), whose energy function is a combination of the ℓ_2 norm $||\mathbf{I}||^2$ that comes from the reference distribution $q(\mathbf{I})$ and the piecewise linear scoring function $f(\mathbf{I}; w)$, i.e.,

$$\begin{aligned}\mathcal{E}(\mathbf{I}; w) &= -f(\mathbf{I}; w) + \frac{1}{2}||\mathbf{I}||^2 \\ &= \frac{1}{2}||\mathbf{I}||^2 - \left(a_{w,\delta(\mathbf{I};w)} + \langle \mathbf{I}, B_{w,\delta(\mathbf{I};w)} \rangle\right) \\ &= \frac{1}{2}\left[\mathbf{I} - B_{w,\delta(\mathbf{I};w)}\right]^2 + \text{const},\end{aligned} \tag{6}$$

where $\text{const} = -a_{w,\delta(\mathbf{I};w)} - ||B_{w,\delta(\mathbf{I};w)}||^2/2$, which is constant on the piece of image space with fixed $\delta(\mathbf{I}; w)$.

Since $\mathcal{E}(\mathbf{I}; w)$ is a piecewise quadratic function, $p(\mathbf{I}; w)$ is piecewise Gaussian. On the piece of image space $\{\mathbf{I} : \delta(\mathbf{I}; w) = \delta\}$, where δ is a fixed value of $\delta(\mathbf{I}; w)$, $p(\mathbf{I}; w)$ is $\mathrm{N}(B_{w,\delta}, \mathbf{1})$ truncated to $\{\mathbf{I} : \delta(\mathbf{I}; w) = \delta\}$, where we use $\mathbf{1}$ to denote the identity matrix. If the mean of this Gaussian piece, $B_{w,\delta}$, is within $\{\mathbf{I} : \delta(\mathbf{I}; w) = \delta\}$, then $B_{w,\delta}$ is also a local mode, and this local mode $\mathbf{I}$ satisfies a hierarchical auto-encoder, with a bottom-up encoding process $\delta = \delta(\mathbf{I}; w)$, and a top-down decoding process $\mathbf{I} = B_{w,\delta}$. In general, for an image sequence $\mathbf{I}$, $B_{w,\delta(\mathbf{I};w)}$ can be considered a reconstruction of $\mathbf{I}$, and this reconstruction is exact if $\mathbf{I}$ is a local mode of $\mathcal{E}(\mathbf{I}; w)$.

3.3 Sampling and learning algorithms

One can sample from $p(\mathbf{I}; w)$ of model (2) by the Langevin dynamics:

$$\mathbf{I}_{\tau+1} = \mathbf{I}_\tau - \frac{\epsilon^2}{2}\left[\mathbf{I}_\tau - \mathbf{B}_{w,\delta(\mathbf{I}_\tau;w)}\right] + \epsilon Z_\tau, \tag{7}$$

where τ indexes the time steps, ϵ is the step size, and $Z_\tau \sim \mathrm{N}(0, \mathbf{1})$. The dynamics is driven by the reconstruction error $\mathbf{I} - \mathbf{B}_{w,\delta(\mathbf{I};w)}$.

The learning of w from training image sequences $\{\mathbf{I}_m, m = 1, ..., M\}$ can be accomplished by the maximum likelihood. Let $L(w) = \sum_{m=1}^{M} \log p(\mathbf{I}; w)/M$, with $p(\mathbf{I}; w)$ defined in (2),

$$\frac{\partial L(w)}{\partial w} = \frac{1}{M} \sum_{m=1}^{M} \frac{\partial}{\partial w} f(\mathbf{I}_m; w) - \mathrm{E}_w \left[\frac{\partial}{\partial w} f(\mathbf{I}; w) \right]. \tag{8}$$

The expectation can be approximated by the Monte Carlo samples (Younes, 1999) produced by the Langevin dynamics. See Algorithm 1 for a description of the learning and sampling algorithm. The algorithm keeps synthesizing image sequences from the current model, and updating the model parameters in order to match the synthesized image sequences to the observed image sequences.

Algorithm 1 Learning and sampling algorithm

```
Require:
    (1) training image sequences $\{\mathbf{I}_m, m = 1, ..., M\}$
    (2) number of synthesized image sequences $\tilde{M}$
    (3) number of Langevin steps $L$
    (4) number of learning iterations $T$
Ensure:
    (1) estimated parameters $w$
    (2) synthesized image sequences $\{\tilde{\mathbf{I}}_m, m = 1, ..., \tilde{M}\}$

1: Let $t \leftarrow 0$, initialize $w^{(0)} \leftarrow 0$.
2: Initialize $\tilde{\mathbf{I}}_m \leftarrow 0$, for $m = 1, ..., \tilde{M}$.
3: repeat
4:     For each $m$, run $L$ steps of Langevin dynamics to update $\tilde{\mathbf{I}}_m$, i.e., starting from the current $\tilde{\mathbf{I}}_m$,
       each step follows equation (7).
5:     Calculate $H^{\rm obs} = \sum_{m=1}^{M} \frac{\partial}{\partial w} f(\mathbf{I}_m; w^{(t)})/M$, and $H^{\rm syn} = \sum_{m=1}^{\tilde{M}} \frac{\partial}{\partial w} f(\tilde{\mathbf{I}}_m; w^{(t)})/\tilde{M}$.
6:     Update $w^{(t+1)} \leftarrow w^{(t)} + \eta(H^{\rm obs} - H^{\rm syn})$, with step size $\eta$.
7:     Let $t \leftarrow t + 1$
8: until $t = T$
```

4 Experiments

We learn the spatial-temporal generative ConvNet from video clips collected from DynTex++ dataset of Ghanem & Ahuja (2010) and the Internet. We also learn the temporal generative ConvNet from sound data collected from the Internet.

The code in the experiments is based on the MatConvNet of Vedaldi & Lenc (2014). We show the synthesis results by displaying the frames in the video sequences and the waveforms of the sounds. The reader can go to `http://www.stat.ucla.edu/~jxie/STGConvNet/STGConvNet.html` to watch the videos and listen to the sounds.

4.1 Experiment 1: Generating dynamic textures with both spatial and temporal stationarity

We first learn the model from dynamic textures that are stationary in both spatial and temporal domains. We use spatial-temporal filters that are convolutional in both spatial and temporal domains. The first layer has 60 $15 \times 15 \times 15$ filters with sub-sampling size of 3 pixels and frames. The second layer has 50 $25 \times 25 \times 5$ filters with sub-sampling size of 5. The third layer has 25 $3 \times 3 \times 2$ filters with sub-sampling size of 2. Figure 2 displays 2 results. For each category, the first row displays some frames of the observed sequence, while the second row shows the corresponding frames of the synthesized sequence generated by the learning algorithm.

We use the layer-by-layer learning scheme. Starting from the first layer, we sequentially add the layers one by one. Each time we learn the model and generate the synthesized image sequence

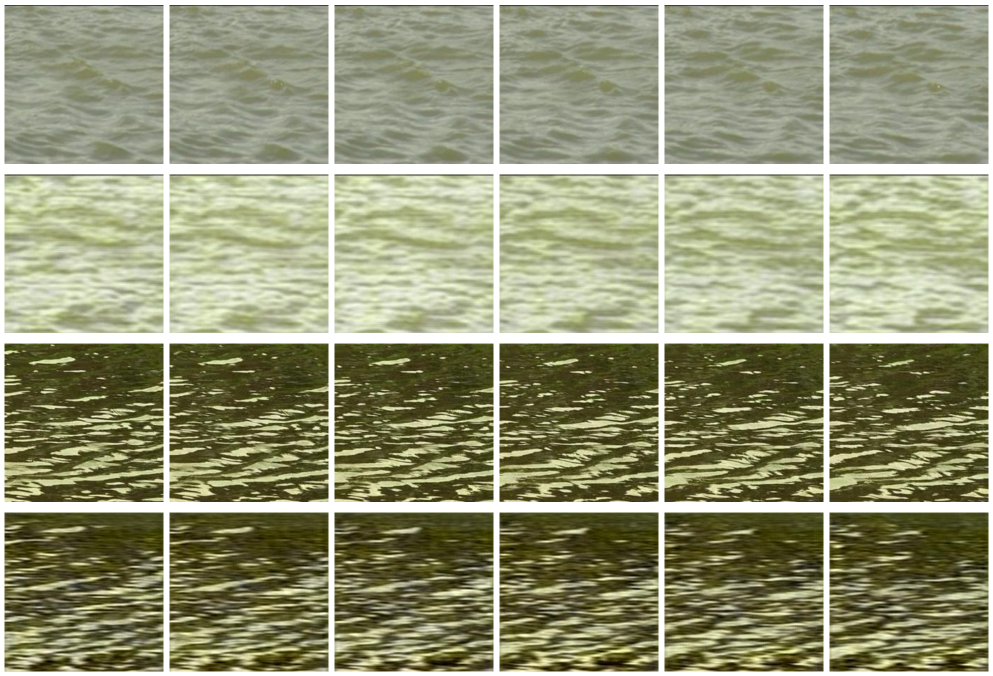


Figure 2: Synthesizing dynamic textures. For each category, the first row displays the frames of the observed sequence, and the second row displays the corresponding frames of the synthesized sequence generated by the learning algorithm. (1) ocean. (2) river.

using Algorithm 1. While learning the new layer of filters, we refine the lower layers of filters with back-propagation.

Due to the limitation of the GPU memory and for the sake of computational efficiency, we use $\tilde{M} = 1$ chain for Langevin sampling. Even with a single chain, the statistics can still be estimated because of stationarity in the temporal and spatial domains. The number of Langevin iterations between every two consecutive updates of parameters $L = 10$. The number of learning iterations $T = 1000$. The observed sequences are prepared to be of size $224 \times 224 \times 40$. The range of intensities is [0, 255]. Mean subtraction is used as pre-processing.

4.2 Experiment 2: Generating dynamic textures with only temporal stationarity

Many dynamic textures have structured background and objects that are not stationary in the spatial domain. In this case, the network used in Experiment 1 may fail. However, we can modify the network in Experiment 1 by using filters that are fully connected in the spatial domain at the second layer. Specifically, the first layer has 120 $7 \times 7 \times 7$ filters with sub-sampling size of 3 pixels and frames. The second layer is a spatially fully connected layer, which contains 30 filters that are fully connected in the spatial domain but convolutional in the temporal domain. The temporal size of the filters is 4 frames with sub-sampling size of 2 frames in the temporal dimension. Due to the spatial fully connectivity at the second layer, the spatial domain of the feature maps at the third layer is reduced to 1×1. The third layer has 5 $1 \times 1 \times 2$ filters with sub-sampling size of 1 in the temporal dimension.

We use end-to-end learning scheme to learn the above 3-layer spatial-temporal generative ConvNet for dynamic textures. At each iteration, the 3 layers of filters are updated with 3 different layer-specific learning rates. The learning rate in the higher layer is much less than that in the lower layer to avoid the issue of large gradients.

We learn a spatial-temporal generative ConvNet for each category from one or two training videos. Figure 3 displays the results. For each category, the first row shows 6 frames of the observed sequence ($224 \times 224 \times 50$), and the second row shows the corresponding frames of the synthesized sequence

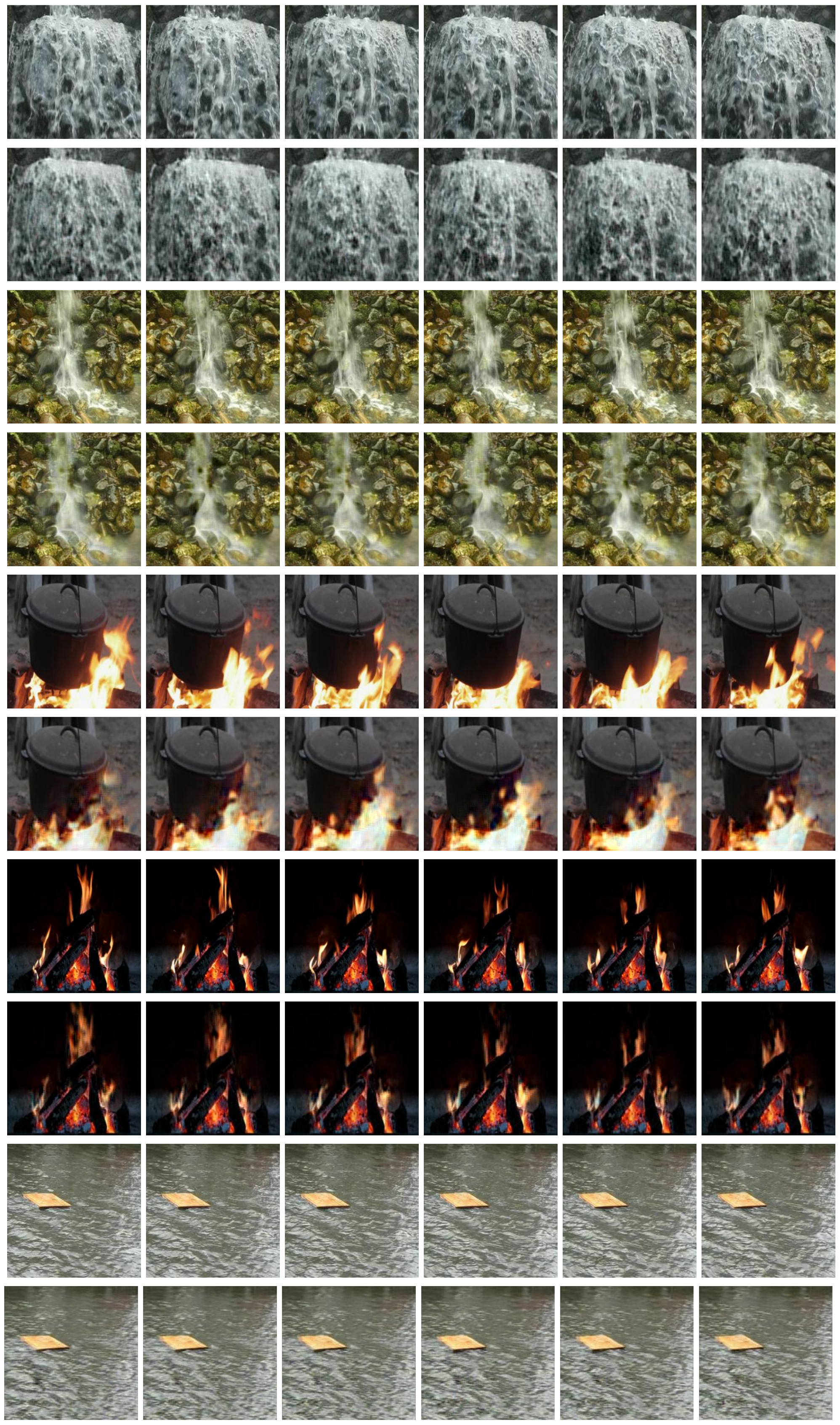
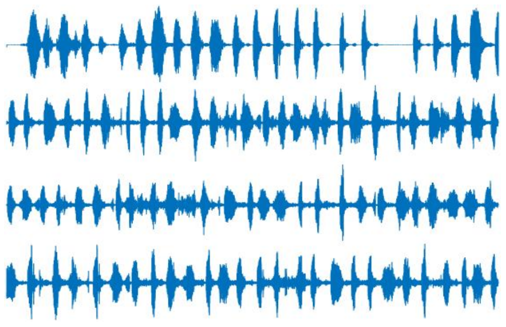
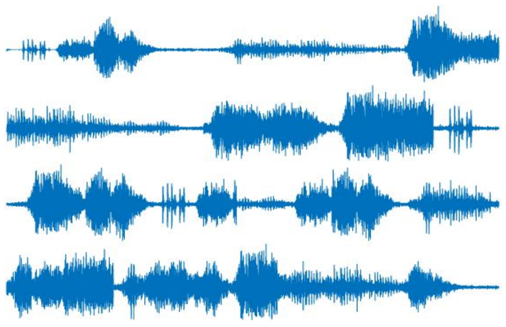
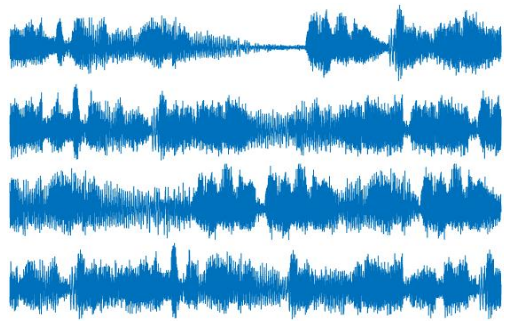


Figure 3: Synthesizing dynamic textures. For each category, the first row displays the frames of the observed sequence, and the second row displays the corresponding frames of the synthesized sequence generated by the learning algorithm. (1) spring water. (2) waterfall. (3) burning fire heating a pot. (4) camp fire at night. (5) floating wood in the river.


(a) chicken


(b) cougar


(c) jazz

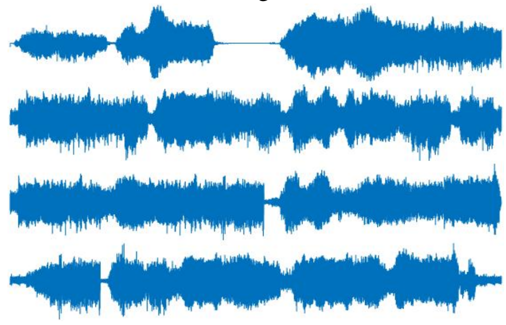

(d) sheep

Figure 4: Synthesizing sounds. For each category, the first row displays the observed sound signal, and the rest display the synthesized sound signals generated by the learning algorithm.

generated by the learning algorithm. We use the same set of parameters for all the categories without tuning.

4.3 Experiment 3: Generating sounds with temporal stationarity

We learn a single layer temporal generative ConvNet for sound patterns. This layer has 1000 1×2000 filters with sub-sampling size of 500. $\tilde{M} = 3$ parallel chains for Langevin sampling are used. The number of Langevin iterations $L = 10$. The number of learning iterations $T = 2000$. We collect some sound data from the Internet, which include animal sounds, natural sounds, and musical sounds. Each observed sound signal is a 5 second long clip, and is read into a 1×60000 dimensional vector with the sampling rate of 11025 Hertz, and the values are normalized into the range of $[-1.0, 1.0]$. Mean subtraction is used before training. Figure 4 shows 4 results by displaying the waveforms (1D plots) of the observed sound signals and the synthesized sound signals. For each category, the first plot is the observed sound, and the rest are 3 synthesized sounds generated by the learned model.

5 Conclusion

This paper shows that the spatial-temporal generative ConvNet can synthesize realistic dynamic textures and sounds that exhibit statistical stationarity in the temporal domain. In our future work, we shall scale up our method to learn from bigger training datasets. We shall apply our model to tasks such as recondition of dynamic textures. We shall also combine the video and sound signals into a single generative ConvNet model.

Webpage for videos and sounds

http://www.stat.ucla.edu/~jxie/STGConvNet/STGConvNet.html contains the observed and synthesized videos and sounds.

Acknowledgement

The work is supported by NSF DMS 1310391, ONR MURI N00014-10-1-0933 and DARPA SIMPLEX N66001-15-C-4035.

References

Dai, Jifeng, Lu, Yang, and Wu, Ying Nian. Generative modeling of convolutional neural networks. In *ICLR*, 2015.

Denton, E., Chintala, S., Szlam, A., and Fergus, R. Deep generative image models using a laplacian pyramid of adversarial networks. *ArXiv e-prints*, 2015.

Doretto, Gianfranco, Chiuso, Alessandro, Wu, Ying Nian, and Soatto, Stefano. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003.

Dosovitskiy, E, Springenberg, J. T., and Brox, T. Learning to generate chairs with convolutional neural networks. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Gatys, L. A., Ecker, A. S., and Bethge, M. A neural algorithm of artistic style. *ArXiv e-prints*, 2015.

Ghanem, Bernard and Ahuja, Narendra. Maximum margin distance learning for dynamic texture recognition. In *Computer Vision–ECCV 2010*, pp. 223–236. Springer, 2010.

Gregor, Karol, Danihelka, Ivo, Graves, Alex, Rezende, Danilo Jimenez, and Wierstra, Daan. DRAW: A recurrent neural network for image generation. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pp. 1462–1471, 2015.

Hochreiter, Sepp and Schmidhuber, Jürgen. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Ji, Shuiwang, Xu, Wei, Yang, Ming, and Yu, Kai. 3d convolutional neural networks for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):221–231, 2013.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *NIPS*, pp. 1097–1105, 2012.

Kulkarni, T. D., Whitney, W., Kohli, P., and Tenenbaum, J. B. Deep Convolutional Inverse Graphics Network. *ArXiv e-prints*, 2015.

LeCun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

LeCun, Yann, Chopra, Sumit, Hadsell, Raia, Ranzato, M, and Huang, F. A tutorial on energy-based learning. *Predicting structured data*, 1:0, 2006.

Lu, Yang, Zhu, Song-chun, and Wu, Ying Nian. Learning frame models using cnn filters. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Montufar, Guido F, Pascanu, Razvan, Cho, Kyunghyun, and Bengio, Yoshua. On the number of linear regions of deep neural networks. In *NIPS*, pp. 2924–2932, 2014.

Ngiam, Jiquan, Chen, Zhenghao, Koh, Pang Wei, and Ng, Andrew Y. Learning deep energy models. In *ICML*, 2011.

Vedaldi, A. and Lenc, K. Matconvnet – convolutional neural networks for matlab. *CoRR*, abs/1412.4564, 2014.

Williams, Ronald J and Zipser, David. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

Xie, Jianwen, Lu, Yang, Zhu, Song-Chun, and Wu, Ying Nian. A theory of generative convnet. In *ICML*, 2016.

Younes, Laurent. On the convergence of markovian stochastic algorithms with rapidly decreasing ergodicity rates. *Stochastics: An International Journal of Probability and Stochastic Processes*, 65(3-4):177–228, 1999.

Zeiler, Matthew D, Taylor, Graham W, and Fergus, Rob. Adaptive deconvolutional networks for mid and high level feature learning. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2018–2025. IEEE, 2011.