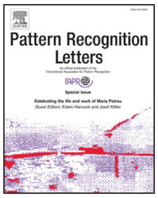
Pattern Recognition Letters 110 (2018) 58–65

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec



# Deep generative video prediction

Tingzhao Yu[a,b,*], Lingfeng Wang[a,c], Huxiang Gu[a], Shiming Xiang[a], Chunhong Pan[a]

[a] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[b] School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 101408, China
[c] Hunan Provincial Key Laboratory of Network Investigational Technology, Hunan Police Academy, Changsha 410138, China

ARTICLE INFO

*Article history:*
Received 25 July 2017
Available online 28 March 2018

*MSC:*
41A05
41A10
65D05
65D17

*Keywords:*
Video prediction
Two stream
Adversarial training
Convlstm

ABSTRACT

Video prediction plays a fundamental role in video analysis and pattern recognition. However, the generated future frames are often blurred, which are not sufficient for further research. To overcome this obstacle, this paper proposes a new deep generative video prediction network under the framework of generative adversarial nets. The network consists of three components: a motion encoder, a frame generator and a frame discriminator. The motion encoder receives multiple frame differences (also known as *Eulerian motion*) as input and outputs a global video motion representation. The frame generator is a pseudo-reverse two-stream network to generate the future frame. The frame discriminator is a discriminative 3D convolution network to determine whether the given frame is derived from the true future frame distribution or not. The frame generator and frame discriminator train jointly in an adversarial manner until a Nash equilibrium. Motivated by theories on color filter array, this paper also designs a novel cross channel color gradient (3CG) loss as a guidance of deblurring. Experiments on two state-of-the-art data sets demonstrate that the proposed network is promising.

## 1. Introduction

Understanding videos is a core problem of pattern recognition and artificial intelligence [35]. It has many applications such as video classification [10], video segmentation [2], video retrieval [6], action recognition [31], crowd analysis [24], event detection [36] and video prediction [18]. Among these applications, video prediction has received growing interests in computer vision and is of great significance for video surveillance [3], video forecasting [32] and autonomous vehicles [13].

As a promising avenue for video understanding, video (or Pixel-level) prediction is of great challenge. This paper addresses the issue of future frame prediction [5,17,20,27,30,33]. Existing methods mainly focus on exploiting the neighbor frame correlation via cross channel or cross frame convolution. For a given video, in order to estimate the discrete joint distribution of the raw pixel values, Video Pixel Network (VPN) [9] constructs a probabilistic model. The model captures the four-dimensional video structure in the temporal dimension of the sequence, in the two spatial dimensions of each frame and in the color channels of a pixel. [20] propose a new spatial-temporal video autoencoder. It consists of a classic spatial image autoencoder and a novel nested temporal autoencoder. At each time step, the network receives a video frame, predicts the optical flow and generates the next frame. Another possible solution for video prediction is generative adversarial networks (GAN) [7], *e.g.* [33] utilize GAN to generate videos from scratch instead of conditioned on the past. Besides, inspired by the concept of predictive coding in neuroscience, [15] propose Predictive Network (PredNet) to predict future frames in a video sequence. Each layer of the network only makes local predictions to subsequent network layers.

This paper proposes a new deep generative architecture for future frame prediction. This work is mainly inspired by Motion and Content Network (MC-net) [32] and Multi Scale Deep Generative network (MSDG) [17]. This paper differs from them in three aspects:

1) Different from MC-net, which is a totally hierarchical network, this paper adopts an adversarial training strategy. This enables the network to automatically generate frames from the original future frame distribution.
2) Different from MSDG, which needs multi-scale RGB information, this paper utilizes single scale frame difference. Using frame difference helps to characterize the motion information more concretely.
3) Different from these two works, this paper employs a new cross channel color gradient loss. This loss function forces the cross

[*] Corresponding author.
*E-mail address:* tingzhao.yu@nlpr.ia.ac.cn (T. Yu).

https://doi.org/10.1016/j.patrec.2018.03.027
0167-8655/© 2018 Elsevier B.V. All rights reserved.

channel color difference to be consistent, thus it can reduce the blur effect.

In fact, the essence of future frame prediction is to minimize the reconstruction error $\|\hat{\mathbf{x}}_{t+1} - \mathbf{x}_{t+1}\|_\alpha^\alpha$ between the true $(t+1)$th future frame $\mathbf{x}_{t+1}$ and the predicted $(t+1)$th future frame $\hat{\mathbf{x}}_{t+1}$, where $\alpha$ is an integer greater or equal to 1. To some extent, it can be dealt with an autoencoder. Srivastavaet al. [30] use a LSTM autoencoder to minimize the reverse video sequence reconstruction error and at the same time present the predict future frame. Walker et al. [34] propose to use Conditional Variational AutoEncoder to depict the uncertain future via optimizing a $\mathcal{KL}$-divergence term $\mathcal{KL}[p(\hat{\mathbf{x}}_{t+1}|\mathbf{x}_{1:t})\|q(\hat{\mathbf{x}}_{t+1}|\mathbf{x}_{1:t})]$, where $p$ represents the predicted distribution and $q$ describes a random distribution.

The contributions of this paper are summarized as follows:

1) A new deep generative network is proposed for video prediction. Within this framework, a generative model generates the future frame of a given video utilizing frame differences. And a discriminative model estimates the probability of a given image being the true future frame. Benefiting from the net architecture, this network can be trained end-to-end via back propagation.
2) A pseudo-reverse two-stream frame generator is proposed for future frame generation. A dynamic stream is designed to encode the motion representation and predict the future motion representation. A static stream is designed to preserve the static content. The future frame is obtained by integrating the outputs of these two streams.
3) A new cross channel color gradient loss is designed to improve the generated frame quality. The motivation is to preserve the consistency among color channels. Theories on color filter array demonstrate that color gradients are consistent for real world images. This can help to highlight the edge areas and reduce blur effect.

Even though the ideas of utilizing difference images Xue et al. [37] and GAN Vondrick et al. [33] have been exploited, the proposed framework is quite different from these works in four aspects.

1) Given a sequence of static frames, Xue et al. [37] take difference images as convolution kernels. However, we take difference images as the input data. Employing difference images as input is essential, because it simplifies the issue of frame prediction to motion representation.
2) For getting better results, Xue et al. [37] need multi-scale frames as input. On the contrary, we only require frames within a single resolution. Thus the proposed network is more efficient.
3) Vondrick et al. [33] utilize GAN for video generation given a random noise. However, we employ GAN for video prediction given a series of known frames. Thus the two problems are quite different from each other.
4) Vondrick et al. [33] apply a two-stream network for both background and foreground generation. Nonetheless, we adopt a pseudo two-stream network, in which only the foreground is predicted and the background is given by the former frame. This in turn makes our network more accurate and more efficient.

## 2. Related work

The main consideration of future frame prediction is to minimize the reconstruction error between the true future frame and the generated future frame. The related works to this paper are video prediction, video synopsis and two-stream networks.

**Video prediction**. Given a short video clip, Ranzato et al. [22] propose a baseline of video prediction based on theories about language model. They construct multiple quantized patch dictionaries and apply a recurrent neural network to classify whether an image patch is the future frame. Yet, the future frame is indeterminate, Xue et al. [37] propose to characterize the future frame in a probabilistic manner. This is implemented via cross convolutional. They regard image and motion as feature maps and convolutional kernels, respectively. A Conditional Variational Autoencoder [12] form loss function, which makes synthesizing many possible future frames possible, is designed to model the probability of the future frame. Specifically, Oh et al. [19] firstly encode the frame level information through CNN and then depict the motion information via LSTM. Deconvolution is applied to decode the predicted frames from the transformed encoding. Nevertheless, the main drawback of these methods is the blur effect caused by the minimization of reconstruction error. Mathieu et al. [17] propose a MSDG network to deal with the inherently blurry predictions. Besides, they design a new image gradient loss function to address the problem of lack of sharpness.

**Video synopsis**. The aim of video synopsis [11,21,28] is to select a sparse subset of video frames that can optimally represent the input video. On the contrary, in this paper, we aim to obtain an optimal motion representation of the input video. Zhang et al. [39] use two LSTMs, one along the time sequence and the other in reverse from the video's end, to select key video frames. To ensure diversity of the selected frames, the network is trained via minimize the cross-entropy loss on annotated ground-truth key frames with an additional loss based on determinantal point process. Mahasseni et al. [16] learn a deep synopsis network to minimize the distance between the training videos and the distribution of their summarizations in an unsupervised way. They utilize a LSTM summarizer to select video frames and a LSTM discriminator to distinguish their similarity.

**Two-stream network**. Motivated by researches on physiology, Simonyan and Zisserman [26] first propose two-stream network for video-based human action recognition. In this paper, we employ a pseudo-reverse two-stream network for future frame generation. Within the framework of two-stream, a spatial network is designed to detect the moving object, and a temporal network is employed for motion recognition [4]. Saito and Matsumoto [23] first implement a two-stream adversarial network, named Temporal Generative Adversarial Network (TGAN), for generating future frames. Different from MSDG, TGAN consists of two generators, a temporal generator and a frame generator. The temporal generator corresponds to motion transformation and the frame generator handles object generation. Besides, Villegas et al. [32] also propose a two-stream network, called MC-net, to decompose the motion and content in videos. The network is built both upon autoencoder and LSTM. Thus it can capture the spatial layout of and temporal dynamics independently.

## 3. Architecture

This section formulates the task of future frame prediction and presents the details of the proposed network. Let $\mathbf{x}_{1:t} \in \mathbb{R}^{t\times w\times h\times c}$ represents the first $t$ frames of a given video $\mathbf{x}$, where $t$, $w$, $h$ and $c$ denote the *temporal length, spatial width, spatial height* and *channel numbers*, respectively. The aim of future frame prediction is to predict the following future frame $\hat{\mathbf{x}}_{t+1}$ conditioned on the given input video frames $\mathbf{x}_{1:t}$. This equals to maximize the conditional distribution $p_\theta(\hat{\mathbf{x}}_{t+1}|\mathbf{x}_{1:t})$ [37], where $p_\theta$ describes the frame distribution.

Nevertheless, a straightforward modeling of $p_\theta(\hat{\mathbf{x}}_{t+1}|\mathbf{x}_{1:t})$ is difficult due to the complex backgrounds in real world videos. Considering the high correlation among neighbor frames, the frame

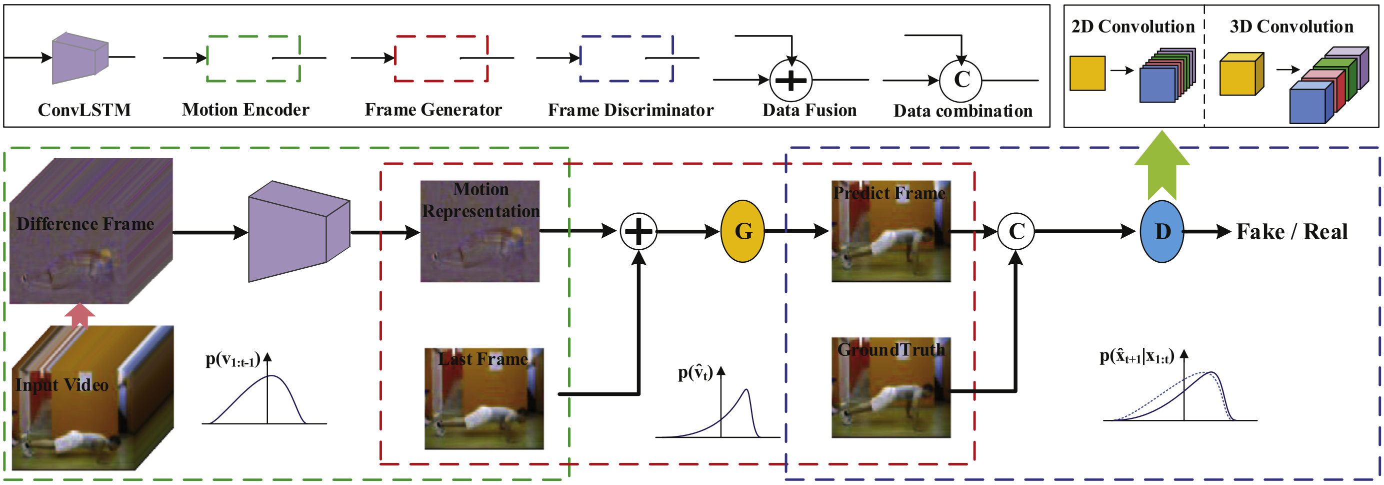


**Fig. 1.** The semantic architecture of deep generative video prediction. The green dash box represents the ***Motion Encoder***, the red dash box denotes the ***Frame Generator*** and the blue dash box indicates the ***Frame discriminator***. The motion encoder first calculates the frame differences and then encode them as a motion representation. The frame generator simply sums the predicted motion representation and the last frame into a future frame. The frame discriminator determines whether the input frame is a real future frame or a fake future frame. Details of this architecture can be found in Section 3. $p(\mathbf{v}_{1:t-1})$ describes the frame difference distribution, $p(\hat{\mathbf{v}}_t)$ depicts the predicted motion distribution and $p(\hat{\mathbf{x}}_{t+1}|\mathbf{x}_{1:t})$ characterizes the predicted future distribution conditioned on the given frames $\mathbf{x}_{1:t}$. The solid curve portrays the true distribution while the dash curve describes the predicted distribution. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

difference $\hat{\mathbf{v}}_t = \hat{\mathbf{x}}_{t+1} - \mathbf{x}_t$ is employed as a substitute based on the frame difference set $\{\mathbf{v}_i : \mathbf{x}_{i+1} - \mathbf{x}_i, i = 1, \dots, t-1\}$. Using frame differences makes the network more robust to noises, *e.g.* background variations. This has also been exploited in [37]. Then the task of future frame prediction can be reformulated as maximize a surrogate distribution $p_\theta(\hat{\mathbf{v}}_t|\mathbf{v}_{1:t-1})$. And the future frame can be reasonably defined as

$$\hat{\mathbf{x}}_{t+1} = \mathbf{x}_t + \hat{\mathbf{v}}_t. \tag{1}$$

The proposed network comprises three components: a **Motion Encoder**, a **Frame Generator** and a **Frame Discriminator** (*see* Fig. 1. for detailed information).

### 3.1. Motion encoder

The motion encoder, denoted **E**, is designed to capture the temporal dynamics $\mathbf{v}_{1:t-1}$ of the given video frames and generate a future motion representation $\hat{\mathbf{v}}_t$. Thus the predicted motion representation is defined by

$$\hat{\mathbf{v}}_t = f^{enc}(\mathbf{v}_{1:t-1}, \mathbf{c}_{1:t-1}), \tag{2}$$

where $\mathbf{c}_{1:t-1}$ records the hidden sequential states in temporal domain. In practice, Long Short Term Memory (LSTM) [8] and Convolution 3D (C3D) [31] can be implemented as the encoder $f^{enc}$ to depict visual dynamics. Though being effective for sequential data, traditional LSTM cannot take spatial correlation into consideration. And C3D is limited to describe the temporal correlation via 3D convolution among neighbor frames (*e.g.* 16 frames). Then a four layer ConvLSTM [25] is employed as the motion encoder. ConvLSTM is an intimate integration of spatial convolution and temporal evolution for its high representation ability both in spatial and temporal domain.

### 3.2. Frame generator

The frame generator, denoted **G**, takes the future motion representation $\hat{\mathbf{v}}_t$ and the last video frame $\mathbf{x}_t$ as input and generates a future frame $\hat{\mathbf{x}}_{t+1}$. This can be formulated as

$$\hat{\mathbf{x}}_{t+1} = \widehat{f^{gen}}(\hat{\mathbf{v}}_t, \mathbf{x}_t). \tag{3}$$

This formulation equals to Eq. (1). In practice, the motion encoder can be regarded as a part of the frame generator. And Eq. (3) can be rewritten as

$$\hat{\mathbf{x}}_{t+1} = \widehat{f^{gen}}(f^{enc}(\mathbf{v}_{1:t-1}, \mathbf{c}_{1:t-1}), \mathbf{x}_t) = f^{gen}(\mathbf{v}_{1:t-1}, \mathbf{x}_t). \tag{4}$$

The frame generator is a pseudo-reverse two-stream network. Two-stream network has been exploited in video action recognition [26]. Here pseudo means that the network is not directly the reverse of the original two-stream network. And reverse means the network takes video representation as input and outputs a video frame, which is the reverse form of the original two-stream network. In this network, the reverse dynamic stream takes the motion representation as input. This stream corresponds to high-frequency dynamic estimation. The reverse static stream takes the last video frame as input. And this stream corresponds to low-frequency content maintenance. Multiple future frames can be recursively generated via simply replacing the last frame $\mathbf{x}_t$ by the newly generated frame $\hat{\mathbf{x}}_{t+1}$ and replacing the motion representation $\hat{\mathbf{v}}_t$ by $\hat{\mathbf{v}}_{t+1}$. Or it can be generated with multiple output motion representations $\hat{\mathbf{v}}_{t:t+\tau}$ and a single last frame $\hat{\mathbf{x}}_t$ by repeat $\hat{\mathbf{x}}_t$ into $\tau$ times, where $\tau$ is the predicted temporal duration.

### 3.3. Frame discriminator

The frame discriminator, denoted **D**, is desired to estimate the probability of a given image being the true future frame. This can be formulated as

$$p = f^{dis}(\mathbf{x}_{t+1}, \hat{\mathbf{x}}_{t+1}). \tag{5}$$

Typical 2D convolution network is capable of discriminating a video frame from a noise input [7]. However, 2D convolution cannot distinguish multiple stacked images from a sequence of successive frames, thus a discriminative 3D convolution network is employed as the frame discriminator.

## 4. Training

Training such a network needs careful consideration about the training loss. As is illustrated before, the basic formulation of future frame prediction is to minimize a distance

$$\mathcal{L}^R(\hat{\mathbf{x}}_{t+1}, \mathbf{x}_{t+1}) = \|\hat{\mathbf{x}}_{t+1} - \mathbf{x}_{t+1}\|_\alpha^\alpha, \tag{6}$$

between the ground truth $(t+1)$th future frame $\mathbf{x}_{t+1}$ and the predicted $(t+1)$th frame $\hat{\mathbf{x}}_{t+1}$. Here $\alpha \geq 0$ is used for controlling the smoothness of the reconstruction error $\mathcal{L}^R(\hat{\mathbf{x}}_{t+1}, \mathbf{x}_{t+1})$.

### 4.1. Reconstruction loss

A simple reconstruction loss is $l_2$ loss, which is defined by $\|\hat{\mathbf{x}}_{t+1} - \mathbf{x}_{t+1}\|_2^2$. However, simply using $l_2$ loss leads to motion blur, for that the $l_2$ loss is more likely to restrict the generated frame to be the average of multiple predictions [17]. This network implements $l_1$ loss as the reconstruction loss

$$\mathcal{L}^R(\mathbf{x}_{t+1}, \hat{\mathbf{x}}_{t+1}) = \|\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1}\|_1^1. \tag{7}$$

### 4.2. Adversarial loss

Except for reconstruction loss, this network also adopts an adversarial training strategy. The adversarial strategy has two strengths: (1) restricts the predicted frame as real world images and (2) highlights the edge details.

**Training discriminator D**. Let $\{(\mathbf{v}^n_{1:t-1}, \mathbf{x}^n_t), \mathbf{x}^n_{t+1}\}$, $n = 1, \cdots, N$ be a set of $N$ training tuples. The discriminator $\mathbf{D}$ is designed to minimize the misjudgement error

$$\begin{aligned}\mathcal{L}^D\{(\mathbf{v}_{1:t-1}, \mathbf{x}_t), \mathbf{x}_{t+1}\} = {} & \mathcal{L}_b(\mathbf{D}((\mathbf{v}_{1:t-1}, \mathbf{x}_t), f^{gen}(\mathbf{v}_{1:t-1}, \mathbf{x}_t)), 0) \\ & + \mathcal{L}_b(\mathbf{D}((\mathbf{v}_{1:t-1}, \mathbf{x}_t), \mathbf{x}_{t+1}), 1).\end{aligned} \tag{8}$$

Specifically, $\mathbf{D}$ is desired to determine the generated frame $f^{gen}(\mathbf{v}_{1:t-1}, \mathbf{x}_t)$ to be 0 (the first term) and the true frame $\mathbf{x}_t$ to be 1 (the second term). $\mathcal{L}_b$ represents the binary cross-entropy loss defined by

$$\mathcal{L}_b(Y, \hat{Y}) = -\sum \hat{Y}\log(Y) - (1-\hat{Y})\log(1-Y). \tag{9}$$

In experiments, $\mathbf{D}(X, Y) \in [0, 1]$ estimates the possibility of $Y$ being the true future frame as Eq. (5).

**Training generator G**. The generator $\mathbf{G}$ is designed to generate future frames that can mislead the discriminator $\mathbf{D}$. As a result, the generative model $\mathbf{G}$ is formulated to maximize the possibility of misjudgment.

$$\mathcal{L}^G\{\mathbf{x}_{t+1}, \hat{\mathbf{x}}_{t+1}\} = \mathcal{L}_b(\mathbf{D}((\mathbf{v}_{1:t-1}, \mathbf{x}_t), f^{gen}(\mathbf{v}_{1:t-1}, \mathbf{x}_t)), 1). \tag{10}$$

In contrast to $\mathbf{D}$, $\mathbf{G}$ is designed to mislead $\mathbf{D}$ to assign a label 1 to the generated frame $f^{gen}(\mathbf{v}_{1:t-1}, \mathbf{x}_t)$.

However, minimizing Eq. (10) alone can lead to instability [17]. Actually, $\mathbf{G}$ can always generate frames that confuses $\mathbf{D}$ while without being close to $\mathbf{x}_{t+1}$. In turn, $\mathbf{D}$ will learn to discriminate these samples, leading $\mathbf{G}$ to generate other confusing samples [17]. A possible solution is to minimize the weighted loss $\lambda_R\mathcal{L}^R(\mathbf{x}_{t+1}, \hat{\mathbf{x}}_{t+1}) + \lambda_G\mathcal{L}^G(\mathbf{x}_{t+1}, \hat{\mathbf{x}}_{t+1})$. $\lambda_R$ and $\lambda_G$ make a tradeoff between the two loss terms.

### 4.3. Cross channel color gradient loss

In order to make the generated frame more sharpening, a novel cross channel color gradient (3CG) loss is proposed. The motivation behind the design of 3CG loss is to achieve high consistence among color channels.

In general, according to theories related to Color Filter Array, for a given image, the color difference between two channels *e.g.*, between red channel and green channel or between green channel and blue channel, should be consistent within a small region [38]. Otherwise, the artifacts (*see* Fig. 5.(c)) will be introduced. The color difference is defined as the diversity of two channels within a spatial position.

The core of 3CG loss is to minimize the cross channel difference between two channels. Then the 3CG loss can be defined as

$$\begin{aligned}\mathcal{L}^C\{\mathbf{x}_{t+1}, \hat{\mathbf{x}}_{t+1}\} = {} & \sum_{i,j}\left\| |\mathbf{x}^{i,j,g}_{t+1} - \mathbf{x}^{i,j,r}_{t+1}| - |\hat{\mathbf{x}}^{i,j,g}_{t+1} - \hat{\mathbf{x}}^{i,j,r}_{t+1}| \right\|^{\alpha} \\ & + \sum_{i,j}\left\| |\mathbf{x}^{i,j,g}_{t+1} - \mathbf{x}^{i,j,b}_{t+1}| - |\hat{\mathbf{x}}^{i,j,g}_{t+1} - \hat{\mathbf{x}}^{i,j,b}_{t+1}| \right\|^{\alpha},\end{aligned} \tag{11}$$

where the superscript *i, j* represent a spatial position and *r, g, b* represent the color channels. The proposed 3CG loss is quite different from Gradient Difference Loss (GDL) [17], which is defined by

$$\begin{aligned}\mathcal{L}^L\{\mathbf{x}_{t+1}, \hat{\mathbf{x}}_{t+1}\} = {} & \sum_{i,j}\left\| |\mathbf{x}^{i,j}_{t+1} - \mathbf{x}^{i-1,j}_{t+1}| - |\hat{\mathbf{x}}^{i,j}_{t+1} - \hat{\mathbf{x}}^{i-1,j}_{t+1}| \right\|^{\alpha} \\ & + \sum_{i,j}\left\| |\mathbf{x}^{i,j}_{t+1} - \mathbf{x}^{i,j-1}_{t+1}| - |\hat{\mathbf{x}}^{i,j}_{t+1} - \hat{\mathbf{x}}^{i,j-1}_{t+1}| \right\|^{\alpha}.\end{aligned} \tag{12}$$

GDL ensures the homogeneity among neighbor positions within a single channel, while 3CG loss ensures the homogeneity among channels within a spatial position. Without the constraint of neighbor positions, using 3CG loss is more likely to generate a frame with distinct edges. In experiments, $\alpha$ is set to be 1. Nevertheless, 3CG loss is also quite different from l1 loss, for that l1 loss also minimize the summation of difference within a channel (*e.g.*, red, green or blue), while 3CG loss minimizes the summation of difference between channels (*e.g.*, between red and green or between green and blue). Section 5 demonstrates the effectiveness of the proposed 3CG loss.

### 4.4. Combining losses

In experiments, the training loss is the weighted combination of the three loss terms. It can be formulated as

$$\begin{aligned}& \mathcal{L}\{\mathbf{x}_{t+1}, \hat{\mathbf{x}}_{t+1}\} \\ & = \lambda_R\mathcal{L}^R(\mathbf{x}_{t+1}, \hat{\mathbf{x}}_{t+1}) + \lambda_G\mathcal{L}^G(\mathbf{x}_{t+1}, \hat{\mathbf{x}}_{t+1}) + \lambda_C\mathcal{L}^C(\hat{\mathbf{x}}_{t+1}, \mathbf{x}_{t+1}),\end{aligned} \tag{13}$$

where $\lambda_R$, $\lambda_G$, and $\lambda_C$ are the corresponding weights of each loss term. $\lambda_R$, $\lambda_G$, and $\lambda_C$ are experimentally set to be 1, $1e^1$ and $1e^{-1}$, according to their influence on the final results. The training process is summarized in Algorithm 1 .

**Algorithm 1:** Training of deep generative network.

```
Input: Training videos $\mathbf{x}^n_{1:t}$, ground-truth future frame $\mathbf{x}^n_{t+1}$
Output: Model Parameters $W_D, W_G$
Initialization: Loss weight $\lambda_R, \lambda_G, \lambda_C$; learning rate $\rho_D, \rho_G$;
while not converge do
    Update the discriminator D;
    $W_D = W_D - \rho_D \sum \frac{\partial \mathcal{L}_D}{\partial W_D}$ ;
    Update the generator G;
    $W_G = W_G - \rho_G \sum \left\{\lambda_R \frac{\partial \mathcal{L}_R}{\partial W_D} + \lambda_G \frac{\partial \mathcal{L}_G}{\partial W_D} + \lambda_C \frac{\partial \mathcal{L}_C}{\partial W_D}\right\}$ ;
end
```

## 5. Evaluation

As a sanity check, this section demonstrates the effectiveness of the proposed network considering single frame prediction. Both quantitative and qualitative results are provided. The code and pre-trained models are released at https://github.com/Tsingzao/DeepGenerativeVideoPrediction for reproduction.

**Table 1**
Network parameters.

| | Motion encoder | Frame discriminator |
|---|---|---|
| Kernel number | 32, 64, 128, c | 64, 128, 256 |
| Kernel size | 3 × 3 | 3 × 3 × 3 |

### 5.1. Datasets

Except the *Moving Mnist*[1] dataset, other two representative video prediction datasets, *i.e., UCF-101*[2] and *Ms. Pac-Man,*[3] are employed to demonstrate the effectiveness of the proposed method. For moving mnist, each video clip consists of 20 frames. There are two digits moving inside a 64 × 64 patch within each frame. The main advantage of this dataset is that researchers can generate training data as much as possible using simulation techniques. UCF-101 [29] consists of 101 action classes for video-based human action recognition. However, it is a challenging task of predicting videos with multiple actions. Actually, when it comes to multiple action classes, researchers often train one network for one action class. Ms. Pac-Man dataset contains 516 game videos for training and 50 game videos for testing. Each video game is comprised of at least 600 frames.

Different from the works that only generates a small patch of a frame, this paper devotes to predicting the entire frame of a given video clip. For simplicity, UCF-101 frames are resized into 64 × 64 in order to fit the size of the network, and Ms. Pac-Man are divided into 64 × 64 patches. Videos are split into multiple five-frame video clips. Thus each video clip equals to a training tuple of four-frame differences plus one last frame.

### 5.2. Implementation details

Table 1 presents the detailed network parameters. The motion encoder consists of four ConvLSTM layers. And the last kernel number corresponds to the channel number of the video frame (i.e. 1 for moving mnist and 3 for both UCF-101 and Ms. Pac-Man). The frame generator simply sums up the predicted motion representation and the last video frame. The frame discriminator contains three 3D convolution layers and a fully connected layer. All of the convolution kernels are of length 3 in each dimension. The network is implemented using *Keras* [1]. *RMSprop* with learning rate 0.001, is used to train the network.

### 5.3. Convergence analysis

One main obstacle of adversarial training is it requires to find a Nash equilibrium [7]. That is to say, this training strategy is unstable and sometimes even cannot converge. Thus the network convergence is first analyzed via quantities of experiments. Figs. 2. and 3 present the training loss curve both on moving mnist and UCF-101. The two figures demonstrate that the generative loss (green line) is unstable during each iteration, but the global loss curve tends to converge after about 3000 (for moving mnist) or 7000 (for UCF-101) iterations. For that the generative loss depicts the similarity between the generated future frame and the ground-truth future frame, the decline of the generative loss indicates that the generated future frame becomes similar to the ground-truth. Note that the loss curve of UCF-101 converges much slower than moving mnist. This can be explained that UCF-101 is a dataset composed of real world videos. It has complex background variations and unpredictable extra motions. This inevitably increases the difficulty of frame prediction.

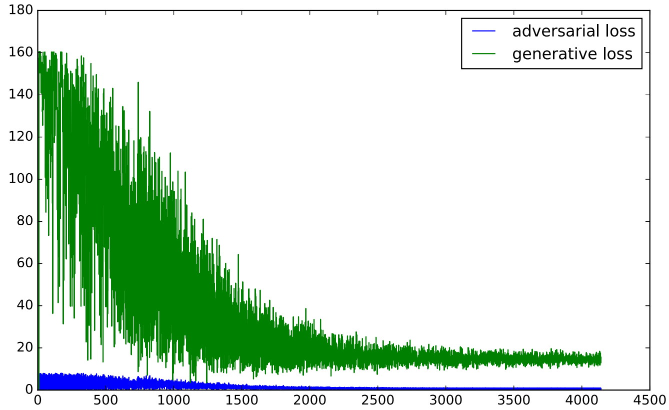


**Fig. 2.** The training loss curve of the proposed network on moving mnist.

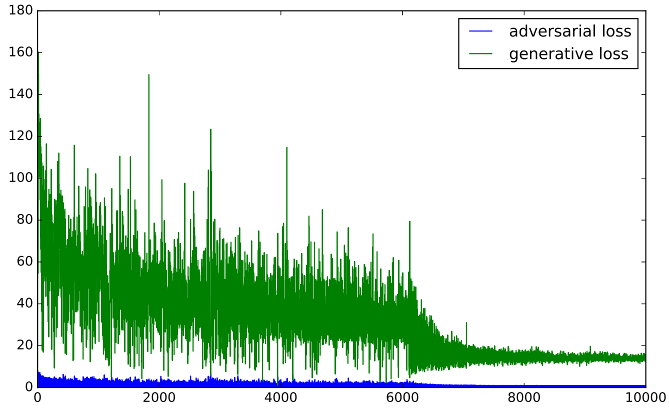


**Fig. 3.** The training loss curve of the proposed network on UCF-101.

**Table 2**
PSNR and SSEQ results for single frame prediction (UCF-101). PSNR depicts the ratio of signal to noise and SSEQ describes the spatial-spectral entropy. A larger PSNR and a smaller SSEQ guarantee a better frame quality.

| Method | $l_2$ | $l_1$ | Adv+$l_1$ | 3CG+Adv+$l_1$ |
|---|---|---|---|---|
| PSNR | 27.6 | 27.7 | 28.5 | 28.6 |
| SSEQ | 53.6 | 53.4 | 42.6 | 39.2 |

### 5.4. Quantitative evaluation

To evaluate the effectiveness of the three loss terms, *i.e.* $l_1$ loss, *GAN* loss and *3CG* loss, both the full-reference Peak Signal to Noise Ratio (PSNR) and no-reference Spatial-Spectral Entropy-based Quality (SSEQ) index [14] are employed. Given the ground-truth future frame $\mathbf{x}_{t+1}$ and the predicted frame $\hat{\mathbf{x}}_{t+1}$, PSNR is defined as

$$PSNR(\mathbf{x}_{t+1}, \hat{\mathbf{x}}_{t+1}) = 10\log_{10} \frac{\max^2 \hat{\mathbf{x}}_{t+1}}{\frac{1}{N}\sum_{i=0}^{N}(\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1})^2}. \tag{14}$$

SSEQ is a SVM-based method that learns to predict image quality scores from local entropy feature vector. Table 2 presents the results of each loss term. The **baseline** is set to be the $l_2$ loss. Table 2 indicates that (**a**) Using $l_1$ loss as reconstruction error performs slightly better than the baseline $l_2$ loss. This is because $l_2$ loss is more sensitive to noises than $l_1$ loss. In real world videos,

[1] http://www.cs.toronto.edu/~nitish/unsupervised_video.
[2] http://crcv.ucf.edu/data/UCF101.php.
[3] https://drive.google.com/open?id=0Byf787GZQ7KvV25xMWpWbV9LdUU.

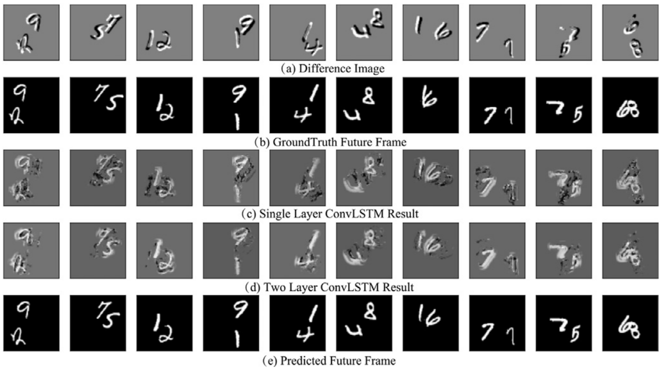

(a) Difference Image
(b) GroundTruth Future Frame
(c) Single Layer ConvLSTM Result
(d) Two Layer ConvLSTM Result
(e) Predicted Future Frame

**Fig. 4.** Examples of results on moving mnist dataset. The first row (a) presents the difference images, the second row (b) gives the ground-truth future frame, the third row (c) demonstrates the results of single-layer convlstm, the fourth row (d) shows the results of two-layer convlstm and the last row (e) is the result of the proposed architecture.

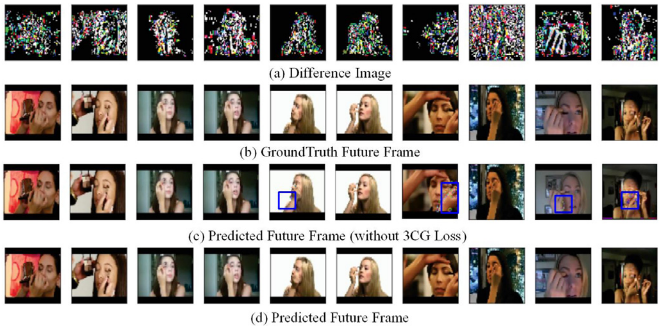

(a) Difference Image
(b) GroundTruth Future Frame
(c) Predicted Future Frame (without 3CG Loss)
(d) Predicted Future Frame

**Fig. 5.** Examples of results on UCF-101 (*ApplyEyeMakeup*). The first row (a) is the difference image, the second row is the ground-truth future frame, the third row (c) is the results without cross channel loss, and the last row (d) presents the integrated results.

*e.g.* UCF-101, there are complex backgrounds and irrelevant motions. (**b**) Integrating the adversarial loss with $l_1$ loss significantly boosts the performance. The adversarial loss can be regarded as a compensation of the $l_1$ loss. $l_1$ loss generates motion blur while the adversarial loss highlights the details. (**c**) A combination of the $l_1$ loss, adversarial loss and 3CG loss outperforms other single loss terms. 3CG loss forces the generated frame performs as real images.

### 5.5. Qualitative evaluation

Figs. 4 and 5 present several qualitative results. Benefiting from the reverse static stream, the network can thoroughly depict the basic frame content. Nevertheless, this mechanism sometimes prevents the reverse dynamic stream to capture the motion properly (*see* Fig. 4 column 2 and column 7). Nevertheless, it outperforms multi-layer AE-ConvLSTM (Fig. 4 row (c) and row (d)), especially in edge areas.

Fig. 5(c) and (d) also demonstrate the effectiveness of the proposed 3CG loss. Without 3CG loss, the hard requirement of adversarial loss can introduce artifacts at the edge areas of the generated frame. 3CG loss can help to eliminate these effects. A possible explanation is the 3CG loss ensures the color difference among RGB channels to be homogeneous. Yet the neighbor frames are sometimes nearly the same and the network is more likely to generate a future frame with high similarity to the last input frame.

**Table 3**
Parameter scale and frame per-pixel error on moving mnist.

| Architecture | # Para | *err* |
|---|---|---|
| [20]-v1 | 0.11M | 0.095 |
| [20]-v2 | 33.62M | 0.065 |
| [20]-v3 | 1.26M | 0.064 |
| [20]-v4 | 1.04M | **0.044** |
| [25]-ConvLSTM | 0.04M | 0.083 |
| [30]-AE-LSTM | 0.12M | 0.067 |
| Proposed | 1.56M | **0.041** |

### 5.6. Comparison with state-of-the-art algorithms

For better comparison with other algorithms, this section re-implements several state-of-the-art algorithms, including AE-LSTM[4] [30], ConvLSTM [25], AE-ConvLSTM[5] [20], MSDG[6] [17], and PredNet[7] [15].

Table 3 presents a comparison of the proposed network with other state-of-the-art networks considering the number of parameters (denoted as # Para) and pixel error (abbreviated as *err*) on moving mnist dataset. The methods in comparison includes AE-LSTM [25], ConvLSTM [30], AE-ConvLSTM (v1) [20] and its variations, *e.g.*, AE-fcLSTM (v2), AE-ConvLSTM (v3) and AE-ConvLSTM-flow (v4).

According to Table 3 using optical flow ([20]-v4) reduces the pixel error significantly. This is because optical flow can be regarded as a representation of motions, with an extra help of this representation, the network can better illustrate the future motion. The proposed method also preserve motion representation using frame differences. Networks with fully connected layers (v2) require large amount of parameters (about 30 times of the proposed network). The proposed network is a generative model which can be divided into two parts, i.e., the generator **G** and the discriminator **D**. The parameters of **G** and **D** are 0.76M and 0.80M, respectively. In fact, both **G** and **D** are employed for training, but during testing, only the generator **G** is employed. That is to say, our network gets an error of 0.041 with 0.76M parameters, therefore, our method outperforms others in testing error and at the same time with less parameters.

Due to the fact that frames from the Ms. Pac-Man dataset have no motion in the majority of pixels, there is no significant difference among the methods in comparison considering PSNR[8]. Fig. 6 presents the qualitative results and SSEQ on Ms. Pac-Man dataset. For better demonstrating the effectiveness of describing motions, in this section, the selected frame sequences are with an temporal interval of three frames. The methods in comparison including AE-ConvLSTM [20], MSDG [17] and PredNet [15]. AE-ConvLSTM is capable of basic content preserving, however, it cannot depict the motions well. Thus the generated frames of AE-ConvLSTM are nearly static backgrounds. MSDG and PredNet can better deal with motion prediction, while the only disadvantage is that they may be blurred due to the effect of $l_2$ loss. The proposed method outperforms other methods both qualitatively and quantitatively.

[4] https://github.com/emansim/unsupervised-videos.
[5] https://github.com/viorik/ConvLSTM.
[6] https://github.com/coupriec/VideoPredictionICLR2016.
[7] https://github.com/coxlab/prednet.
[8] https://github.com/dyelax/Adversarial_Video_Generation.

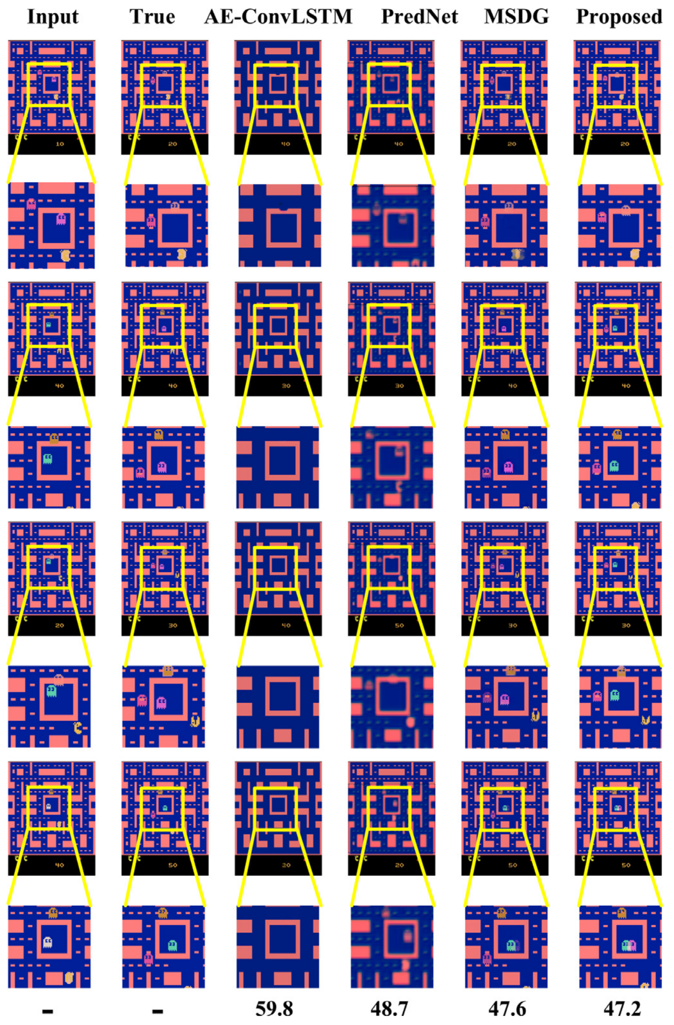


**Fig. 6.** A qualitative comparison with state-of-the-art methods, *i.e.*, AE-ConvLSTM [20], MSDG [17], PredNet [15] and the proposed method (column 3 to column 6). The first column is the input frame and the second column is the groundtrue future frame. The last row is the average SSEQ index.

## 6. Conclusion

This letter proposes a new deep generative video prediction network for future frame prediction. Within this model, a new pseudo-reverse two-stream network is designed as the frame generator and a shallow 3D convolution network is designed as the frame discriminator. Besides, a novel cross channel color gradient loss is proposed to improve the frame quality. Both quantitative and quantitative experiments on two state-of-the-art datasets illustrate the effectiveness of the new network.

In future work, a more effective motion encoder will be exploited to demonstrate multiple frames prediction. Besides, considering long temporal range, a probabilistic motion generator is under consideration.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant Number 61403376, 61573352, 91646207 and 91438105); the Beijing Natural Science Foundation (Grant Number 4162064); and the Open Research Fund of Hunan Provincial Key Laboratory of Network Investigational Technology (Grant No. 2015HNWLFZ055).

## References

[1] F. Chollet, Keras, 2015, (https://github.com/fchollet/keras).
[2] K. Desouza, A. Albuquerquearaújo, Z. Patrocínio, S. Guimarães, Graph-based hierarchical video segmentation based on a simple dissimilarity measure, Pattern Recognit. Lett. 47 (2014) 85–92.
[3] I. Elafi, M. Jedra, N. Zahid, Unsupervised detection and tracking of moving objects for video surveillance applications, Pattern Recognit. Lett. 84 (2016) 70–77.
[4] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1933–1941.
[5] C. Finn, I. Goodfellow, S. Levine, Unsupervised learning for physical interaction through video prediction, in: Advances in Neural Information Processing Systems, 2016, pp. 64–72.
[6] X. Gao, X. Li, J. Feng, D. Tao, Shot-based video retrieval with optical flow tensor and hmms, Pattern Recognit. Lett. 30 (2) (2009) 140–147.
[7] I. Goodfellow, J. Pougetabadie, M. Mirza, B. Xu, D. Wardefarley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.
[8] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[9] N. Kalchbrenner, A. Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, K. Kavukcuoglu, Video pixel networks, arXiv:1610.00527 (2016).
[10] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. FeiFei, Large-scale video classification with convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.
[11] A. Khosla, R. Hamid, C. Lin, N. Sundaresan, Large-scale video summarization using web-image priors, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2698–2705.
[12] D. Kingma, M. Welling, Auto-encoding variational bayes, arXiv:1312.6114 (2013).
[13] T. Litman, Autonomous vehicle implementation predictions, Victoria Transp. Policy Instit. 28 (2014).
[14] L. Liu, B. Liu, H. Huang, A.C. Bovik, No-reference image quality assessment based on spatial and spectral entropies, Signal Process. Image Commun. 29 (8) (2014) 856–863.
[15] W. Lotter, G. Kreiman, D. Cox, Deep predictive coding networks for video prediction and unsupervised learning, arXiv preprint arXiv:1605.08104 (2016).
[16] B. Mahasseni, M. Lam, S. Todorovic, Unsupervised video summarization with adversarial lstm networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[17] M. Mathieu, C. Couprie, Y. LeCun, Deep multi-scale video prediction beyond mean square error, arXiv:1511.05440 (2016).
[18] N. Neverova, P. Luc, C. Couprie, J. Verbeek, Y. LeCun, Predicting deeper into the future of semantic segmentation, arXiv:1703.07684 (2017).
[19] J. Oh, X. Guo, H. Lee, R. Lewis, S. Singh, Action-conditional video prediction using deep networks in atari games, in: Advances in Neural Information Processing Systems, 2015, pp. 2863–2871.
[20] V. Patraucean, A. Handa, R. Cipolla, Spatio-temporal video autoencoder with differentiable memory, arXiv:1511.06309 (2016).
[21] Y. Pritch, A. Ravacha, S. Peleg, Nonchronological video synopsis and indexing, IEEE Trans. Pattern Anal. Mach. Intell. 30 (11) (2008) 1971–1984.
[22] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, S. Chopra, Video (language) modeling: a baseline for generative models of natural videos, arXiv:1412.6604 (2014).
[23] M. Saito, E. Matsumoto, Temporal generative adversarial nets, arXiv:1611.06624 (2016).
[24] J. Shao, C. Loy, K. Kang, X. Wang, Slicing convolutional neural network for crowd video understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5620–5628.
[25] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, W. Woo, Convolutional lstm network: a machine learning approach for precipitation nowcasting, in: Advances in Neural Information Processing Systems, 2015, pp. 802–810.
[26] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Advances in Neural Information Processing Systems, 2014, pp. 568–576.
[27] W. Softky, Unsupervised pixel-prediction, in: Advances in Neural Information Processing Systems, 1996, pp. 809–815.
[28] Y. Song, J. Vallmitjana, A. Stent, A. Jaimes, Tvsum: Summarizing web videos using titles, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5179–5187.
[29] K. Soomro, A. Zamir, M. Shah, Ucf101: a dataset of 101 human actions classes from videos in the wild, arXiv:1212.0402 (2012).
[30] N. Srivastava, E. Mansimov, R. Salakhutdinov, Unsupervised learning of video representations using lstms., in: Proceedings of the International Conference on Machine Learning, 2015, pp. 843–852.
[31] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4489–4497.
[32] R. Villegas, J. Yang, S. Hong, X. Lin, H. Lee, Decomposing motion and content for natural video sequence prediction, arXiv:1706.08033 (2017).
[33] C. Vondrick, H. Pirsiavash, A. Torralba, Generating videos with scene dynamics, in: Advances In Neural Information Processing Systems, 2016, pp. 613–621.
[34] J. Walker, C. Doersch, A. Gupta, M. Hebert, An uncertain future: forecasting

from static images using variational autoencoders, in: European Conference on Computer Vision, 2016, pp. 835–851.

[35] X. Wang, Intelligent multi-camera video surveillance: a review, Pattern Recognit. Lett. 34 (1) (2013) 3–19.

[36] Z. Xu, Y. Yang, A. Hauptmann, A discriminative cnn video representation for event detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1798–1807.

[37] T. Xue, J. Wu, K. Bouman, B. Freeman, Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks, in: Advances in Neural Information Processing Systems, 2016, pp. 91–99.

[38] T. Yu, W. Hu, W. Xue, W. Zhang, Colour image demosaicking via joint intra and inter channel information, Electron Lett. 52 (8) (2016) 605–606.

[39] K. Zhang, W. Chao, F. Sha, Video summarization with long short-term memory, in: European Conference on Computer Vision, 2016, pp. 766–782.