EDGE GUIDED GENERATION NETWORK FOR VIDEO PREDICTION

Kai Xu[1], *Guorong Li*[1],*Huijuan Xu*[2],*Weigang Zhang*[3], *Qingming Huang*[1]

[1]School of Computer and Control Engineering, University of Chinese Academy of Sciences, China
[2]Boston University, USA [3]Harbin Institute of Technology, Weihai, China
xukai16@mails.ucas.ac.cn,{liguorong,qmhuang}@ucas.ac.cn, hxu@bu.edu,wgzhang@hit.edu.cn

ABSTRACT

Video prediction is a challenging problem due to the highly complex variation of video appearance and motions. Traditional methods that directly predict pixel values often result in blurring and artifacts. Furthermore, cumulative errors can lead to a sharp drop of prediction quality in long-term prediction. To alleviate the above problems, we propose a novel edge guided video prediction network, which firstly models the dynamic of frame edges and predicts the future frame edges, then generates the future frames under the guidance of the obtained future frame edges. Specifically, our network consists of two modules that are ConvLSTM based edge prediction module and the edge guided frames generation module. The whole network is differentiable and can be trained end-to-end without any supervision effort. Extensive experiments on KTH human action dataset and challenging autonomous driving KITTI dataset demonstrate that our method achieves better results than state-of-the-art methods especially in long-term video predictions.

***Index Terms*—** Video prediction, deep learning, spatial-temporal network, image generation

1. INTRODUCTION

Since video provides more spatial and temporal information than image, it is extensively studied in many tasks such as video caption, video detection and video segmentation. These supervised learning tasks require large number of labeled training data, which limits the utility of deep learning in many tasks where such data are not available. Video frame prediction can be used to learn the effective representation within the video through unsupervised methods. In addition, video prediction can promote many applications, such as activity prediction, event prediction, human-robot interaction and automatic driving. Unfortunately, pixel-level video prediction is very challenging due to the inherent uncertainty of videos and the changes of various factors, such as object motion, deformation, occlusion and background transformation.

Recently, there have been a lot of works on video frames prediction [1–5]. They employ deep network to model the spatial-temporal dependencies and motion rule of pixels in the video, then hallucinate future frame pixels directly. These previous works usually result in blurring and artifacts because of the inherent uncertainty in video. In long-term prediction, these exiting methods take the predicted frames as input recursively for further prediction, so the errors will accumulate and result in a sharp drop of prediction quality.

Compared to pixel, image edge is an important and more robust high-level structure of image. Phillip et al. [6] proposed the Image-to-Image Translation network which can generate diverse corresponding natural images from an edge image. Moreover, through edge detection, only the important structural attributes of the image are retained and the amount of data can be greatly reduced. So, the edge image prediction is simpler and more robust than frame prediction. Therefore, in order to solve the problems mentioned above, we propose an edge guided video prediction network (EVPnet), which first models the motion of the video frame edges and predicts the future edge images, then the corresponding future frame is generated by the guidance of the obtained future edge image. Due to the robustness of edge image prediction, our method can realize the long-term video prediction. Besides, our network can be applied to any video because the edge is a general high-level structure and is easy to obtain.

Specifically, our network consists of two modules, namely the edge prediction module and the edge guided frame generation module. These two modules combine into a whole network that can be trained and tested end-to-end. The edge prediction module consists of a fully convolutional edge encoder-decoder and a nested memory convolutional LSTM cell [7], which models the dynamic of the video frame edges and predicts the future edge images. The second module is the edge guided frame generation network, which generates the corresponding frame from the predicted edge representation. Different from [6], the content of the predicted video frames by our edge guided frame generator should be consistent with the observed video frames. The frame generation module consists of a content encoder and a frame decoder. In addition, we design a two-pathway skip connection that directly connects the intermediate results of content encoder and edge decoder respectively to frame decoder with the purpose of promoting the propagation of low-level information.

The contributions of this paper are three-fold. 1) We propose a novel video prediction network that decomposes the

978-1-5386-1737-3/18/$31.00 ©2018 IEEE

video prediction task into the edge prediction and the edge guided frame generation. 2) A frame generator with a two-pathway skip connection is proposed, which maps predicted edges to natural frames and meanwhile ensures the generated frames to be consistent with the observed frames sequence. 3) Experiments conducted on KTH human action dataset and more challenging KITTI autonomous driving dataset demonstrate that our approach achieves great improvement in terms of quantitative and qualitative evaluations over state-of-the-art methods in long-term video prediction.

2. RELATED WORK

Long short-term memory (LSTM) network [8] has been successfully applied to temporal sequence related tasks such as language model, so it is natural to employ LSTM to the task of video prediction [1, 2]. Inspired by the language model, Ranzato et al. [1] build a LSTM network to predict future frames by predicting the combination coefficients of image patches on visual word dictionaries. Srivastava et al. [2] adapt a LSTM model and an autoencoder to predict the unseen future frames. Since the traditional LSTM network employs fully connected operation, it is not suitable for processing two-dimensional image data due to the loss of spatial information. Shi et al. [7] combine convolution operation with LSTM obtaining the ConvLSTM network to predict nowcasting. Some works combine ConvLSTM network and optical flow for video prediction [5, 9].

In order to reduce the blurring of the generated image, a multi-scale structure which combines the adversarial loss and the GDL loss is proposed [3]. Motivated by the predictive coding concept in neuroscience literature, W. Lotter et al. [4] develop an architecture that predicts realistic looking frames. For human action video, LSTM is used to predict the key point of human body to generate the corresponding image [10]. However, it can only be used in human action videos and is not able to model the background variations. R. Villegas et al.[11] propose a two-encoder architecture to separately model motion and content then combine the two features to predict the next frame.

3. THE PROPOSED METHOD

The task of video prediction is to observe k video frames $\boldsymbol{x}_{1:k}$ and then output the next T frames $\hat{\boldsymbol{x}}_{k+1:k+T}$, while the ground-truth is represented as $\boldsymbol{x}_{k+1:k+T}$. We propose an edge guided video prediction network (EVPnet), which is an end-to-end differentiable network. The only training data we need are the video frames without any human annotations. By functional division, our EVPnet consists of the edge prediction module and the edge guided frame generation module. In order to improve the texture details and structure of the predicted frames, we design a two-pathway skip connection in the frame generator. Finally, we combine some complementary sub-loss function to generate realistic frames. More details of our model are presented in the following sections.

3.1. Edge Prediction Module

In this section, a detailed description of the edge prediction network is given. The edge prediction network requires a frame sequence $\boldsymbol{x}_{1:k}$ as input and outputs the subsequent T edge images $\hat{\boldsymbol{e}}_{k+1:k+T}$. The edge prediction module is composed of an edge encoder and edge decoder with a ConvLSTM cell embedded between them. The illustration of our edge prediction module is shown in Fig.1(a). The edge encoder and decoder constitute an edge detector to extract the edge image from natural image. The edge encoder extracts the embedding edge features by

$$\boldsymbol{f}^e = En^e(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{x}$ is the input image,$\boldsymbol{f}^e$ is the the embedding edge feature extracted by the edge encoder, $En^e(\cdot)$ is the edge encoder function. The edge decoder generates the edge image from embedding features by

$$\hat{\boldsymbol{e}} = De^e(\boldsymbol{f}^e) \tag{2}$$

where $\hat{\boldsymbol{e}}$ is the detected edge image of input image $\boldsymbol{x}$, $De^e(\cdot)$ is the edge decoder function.

The ConvLSTM cell embedded between edge encoder and decoder can model the dynamic variation of edge features through observing continuous edge feature sequences$\boldsymbol{f}^e_{1:k}$ by

$$[\boldsymbol{h}_t, \boldsymbol{c}_t] = ConvLSTM(\boldsymbol{f}^e_t, \boldsymbol{h}_{t-1}, \boldsymbol{c}_{t-1}) \tag{3}$$

where $\boldsymbol{c}_t$ is a memory cell that retains temporal dynamics information of the observed sequence until time t. $\boldsymbol{h}_t$ is the output of ConvLSTM cell at time step t, which is also the predicted edge feature tensor at next time step, i.e. $\hat{\boldsymbol{f}}^e_{t+1} = \boldsymbol{h}_t$. After observing the k video sequence frames, ConvLSTM has modeled the dynamic rule of the edge features, then can predict the edge features of the subsequent T frames according to the learned historical rule.

After obtaining the predicted edge features$\hat{\boldsymbol{f}}^e_{k+1:T}$, they are used to generate corresponding video frames (Section 3.2). The edge features are also used to generate the edge image by edge decoder as Equation(2), then the edge loss can be calculated (Section 3.4).

3.2. Edge Guided Frame Generation Module

Edge guided frame generation module aims to generate the corresponding frame using the predicted edge feature. Phillip et al. [6] proposed an image to image network based on Generative Adversarial Networks [12] which can generate the corresponding real image from edge image. In [6] one edge image generates diverse real images with the same edge structure but random content. In order to do this, method in [6] takes an edge image and a random noise vector as inputs to ensure diversity. Different from [6], video prediction requires that the content of the generated video frame is consistent with the observed video sequence. Due to the continuity of the video in time domain, the content variation between adjacent video frames is similar. So, we replace the noise vector with the content feature extracted from the last observed

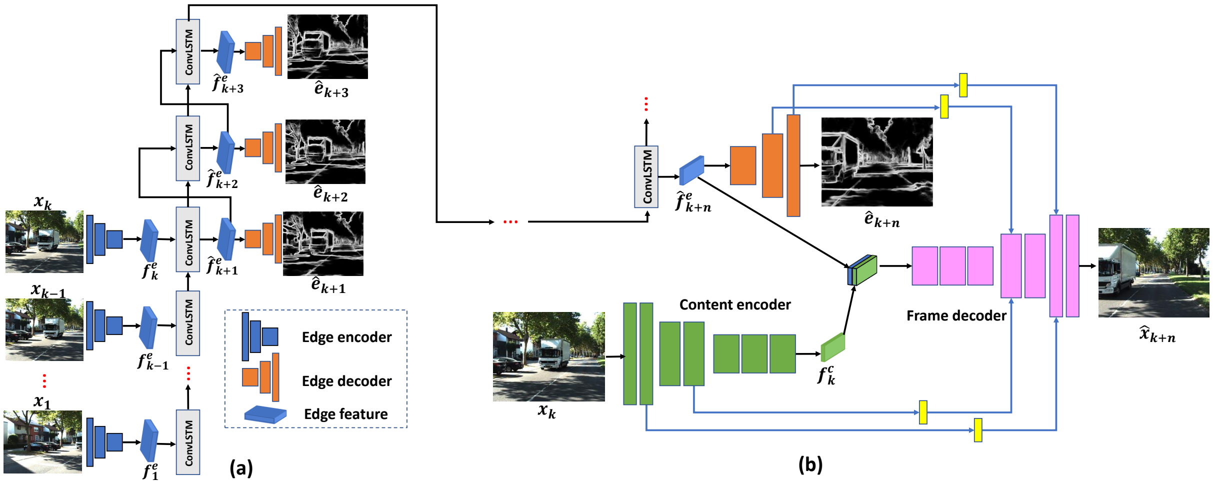

Fig. 1. Illustration of our edge guided video prediction network. (a) The unrolled illustration of our edge prediction module. Edge encoder observes k consecutive video frames and extracts their edge features. ConvLSTM cell models the dynamic of k edge features, then predicts the next T edge features, which are inputted to edge decoder to generate the corresponding predicted edge images. (b) The illustration of our edge guided frame generation module. The content features are extracted by content encoder from the last frame of the observed video sequence $\boldsymbol{x}_k$. Then the content features concatenate with the edge features predicted by edge prediction module at $k+n$ time step. Finally, the concatenated feature maps are fed into the frame decoder and the frame at $k+n$ time step is obtained. The blue arrow represents skip connection, and the yellow rectangle in skip connection is the channel compression layer.

video frame. The edge guided video frame generator makes not only the structure of the generated frame be consistent with the predicted edge, but also the content be similar to the last observed video frame. The frame generator, as shown in Fig.1(b), consists of content encoder, frame decoder and the two-pathway skip connection. The content encoder extracts embedding content features from the last observed frame by

$$\boldsymbol{f}^c_k = En^c(\boldsymbol{x}_k) \tag{4}$$

where $\boldsymbol{x}_k$ is the last frame of the observed video sequence, $\boldsymbol{f}^c_k$ is the encoding content feature. $En^c(\cdot)$ is the content encoder function. Then the content feature tensor concatenates with the edge feature tensor as input to the frame decoder. Finally the frame decoder generates the predicted frame by

$$\hat{\boldsymbol{x}}_{k+n} = De^f([\boldsymbol{f}^c_k, \hat{\boldsymbol{f}}^e_{k+n}]) \tag{5}$$

where $\hat{\boldsymbol{x}}_{k+n}$ is the predicted frame at $k+n$ time step, $[\cdot]$ is the concatenation operation. $De^f(\cdot)$ is the frame decoder function. Note that the n is from 1 to T, so the predicted subsequent T frames $\hat{\boldsymbol{x}}_{k+1:k+T}$ can be obtained.

3.3. Two-Pathway Skip Connection

In deep network, shallow layers extract low-level visual features while deep layers extract high-level advanced semantic features. Low-level details are important for image generation tasks. When using encoder-decoder structure to generate images, a lot of information will be lost due to the existence of max pooling layer. So we design a two-pathway skip connection which can retain more details of the generated images by transferring the low-level features directly to frame decoder. In particular, the two-pathway skip connection connects the intermediate results of content encoder and edge decoder respectively to frame decoder as shown in Fig.1(b). From content encoder skip connection, more detail features of content can be transmitted to frame decoder, and the edge structure of predicted frames can be emphasized from edge decoder skip connection. The concatenation operation is used to combine the decoder feature maps and skip feature maps. In order to avoid propagating useless information and increasing the amount of calculation, we add a channel compression layer in the skip connection, which is implemented by the convolutional layer with 1×1 kernel. As a result, the output channel number is compressed fewer than the input, which means only the channels with decision and effective information are transmitted. The compression ratio of edge skip connection ρ_e and content skip connection ρ_c are hyper-parameters.

3.4. Loss Function

In this section, the loss function used to train our network is introduced in detail. In order to produce realistic image quality, we combine several complementary sub-loss functions, namely edge loss, l_p norm loss, gradient difference loss and the adversarial loss. We combine all the sub-loss functions to form the total loss function as follows

$$L = \lambda_e L_e + \lambda_i (L_p + L_{gdl}) + \lambda_{gan} L_{gan} \tag{6}$$

where λ_e, λ_i and λ_{gan} are hyper-parameters that control the effect of each sub-loss during optimization.

The edge prediction module observes k frames, then predicts the edge images of next T frames. We hope that the predicted edge is accurate, so we employ the edge loss as follows

$$L_e = \frac{1}{T}\sum_{t=k+1}^{k+T} \|\boldsymbol{e}_t - \hat{\boldsymbol{e}}_t\|_2^2 \tag{7}$$

where $\boldsymbol{e}_t$ is the ground-truth edge image of image $\boldsymbol{x}_t$, $\hat{\boldsymbol{e}}_t$ is the edge image predicted by the edge prediction module. In addition, the generated video frame is expected to be similar to the ground-truth in the pixel space, so the l_p norm loss between the predicted frame and the ground-truth is adopted as follows

$$L_p = \frac{1}{T}\sum_{t=k+1}^{k+T} \|\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t\|_p^p \tag{8}$$

where $\boldsymbol{x}_t$ is the ground truth frame at time step t, $\hat{\boldsymbol{x}}_t$ is the predicted frame by our edge guided frame generator. However, using only l_p loss likely results in blurring. In order to increase the sharpness of the image, we use the image gradient difference loss (GDL) proposed in [3]

$$\begin{aligned} L_{gdl} =& \frac{1}{T}\sum_{t=k+1}^{k+T}\sum_{i,j}^{h,w} \left|(|\boldsymbol{x}_t^{i,j} - \boldsymbol{x}_t^{i-1,j}| - |\hat{\boldsymbol{x}}_t^{i,j} - \hat{\boldsymbol{x}}_t^{i-1,j}|)\right|^\gamma \\ &+ \left|(|\boldsymbol{x}_t^{i,j-1} - \boldsymbol{x}_t^{i,j}| - |\hat{\boldsymbol{x}}_t^{i,j-1} - \hat{\boldsymbol{x}}_t^{i,j}|)\right|^\gamma, \end{aligned} \tag{9}$$

where i and j are the pixel coordinates, h and w are the height and width of image, γ is the hyper-parameters of the GDL loss.

In order to make the generated image realistic, the adversarial loss is introduced. Generative adversarial networks are introduced by [12], where images are generated from random noise using the generator network G and the discriminator network D trained simultaneously. In our method, the proposed EVPnet is considered as the generator G which generates the future T video frames. The discriminator network D is trained to predict the probability that the last T frames of the sequence are generated by G. The two networks are simultaneously trained so that G learns to generate frames that are hard to be classified by D, while D learns to discriminate the frames generated by G. The generator loss is defined as follows

$$L_{gan} = -log\ D([\ \boldsymbol{x}_{1:k},\ G(\boldsymbol{x}_{1:k})\ ]) \tag{10}$$

where $\boldsymbol{x}_{1:k}$ is the input images, $G(\boldsymbol{x}_{1:k}) = \hat{\boldsymbol{x}}_{k+1:k+T}$ is predicted future images, and $D(\cdot)$ is the discriminator in adversarial training. The discriminative loss in adversarial training is defined by

$$L_{disc} = -log\ D([\boldsymbol{x}_{1:k}, \boldsymbol{x}_{k+1:k+T}]) - log\ (1 - D([\boldsymbol{x}_{1:k}, G(\boldsymbol{x}_{1:k})])) \tag{11}$$

where $\boldsymbol{x}_{k+1:k+T}$ is the ground-truth future images.

4. EXPERIMENT RESULTS

In this section, video prediction experiments are implemented in two real-world datasets including KTH human action dataset [13] and more challenging autonomous driving KITTI dataset [14]. We use the HED [15] algorithm to generate edge ground-truth of the video frames, while the HED algorithm can be replaced with other edge detector. Intuitively, our EVPnet could obtain better results if using more accurate edge detector to generate the edge ground-truth. All the video frames and edge images are normalized to $[-1, 1]$. The peak signal to noise ratio (PSNR), structural similarity (SSIM) [16] and CPBD [17] are adopted as the quantitative evaluations. CPBD is a perceptual-based no-reference objective image sharpness metric. We compare our EVPnet with the baseline method ConvLSTM [7] and state-of-the-art methods MCnet [11] and Prednet [4]. In all the experiment, we set $\rho_e = \rho_c = \frac{1}{2}$, $\lambda_e = 1$. For fair comparison, we set $\lambda_i = 1$, $p = 2$, $\gamma = 1$, which are the same as [11]. We implement our network using the publicly available Pytorch framework and one TITAN X GPU. The network is trained by the Adam optimization algorithm with the learning rate 0.0001.

4.1. Architectures

The content encoder of EVPnet is built with the same architecture as VGG16 [18] up to the third pooling layer. The frame decoder is a mirrored architecture of the content encoder where we replace the max-pooling layer with bilinear interpolation up-sampling layer. The edge encoder is also similar to content encoder, except that we replace its consecutive convolutions group with single convolution in each layer. And the edge decoder is also a mirrored architecture of the content encoder. A Hyperbolic tangent is added at the end of both the edge decoder and frame decoder to ensure that the output values are between -1 and 1. The ConvLSTM cell employs 3×3 convolutions with 256 channels. The channel compression layer in skip connection are single 1×1 convolution layer. The discriminative network D are composed of 4 consecutive 5×5 convolutions (64,128,256 and 512 in each layer) followed by a fully connected layer and a sigmoid function. For the baseline ConvLSTM, we use the same architecture as the content encoder, ConvLSTM cell and frame decoder.

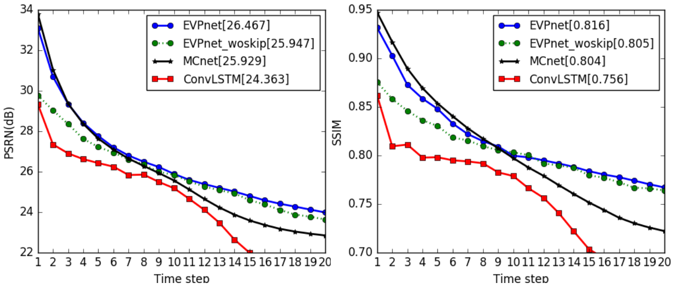

Fig. 2. The quantitative comparisons among our EVPnet and the variation without skip connection, MCnet and the ConvLSTM on KTH dataset. The figure in legend is the average results over 20 time steps.

4.2. KTH Dataset

The KTH human action dataset [13] contains 6 categories action of 25 persons: running, jogging, walking, boxing, handclapping and hand-waving. We train and test our network using the same data and protocol as in [11] (i.e. person 1-17 for training and 18-25 for testing). All frames are resized to 128×128 pixels. In the experiment, all methods observe 10 frames and predict 10 frames into the future when training

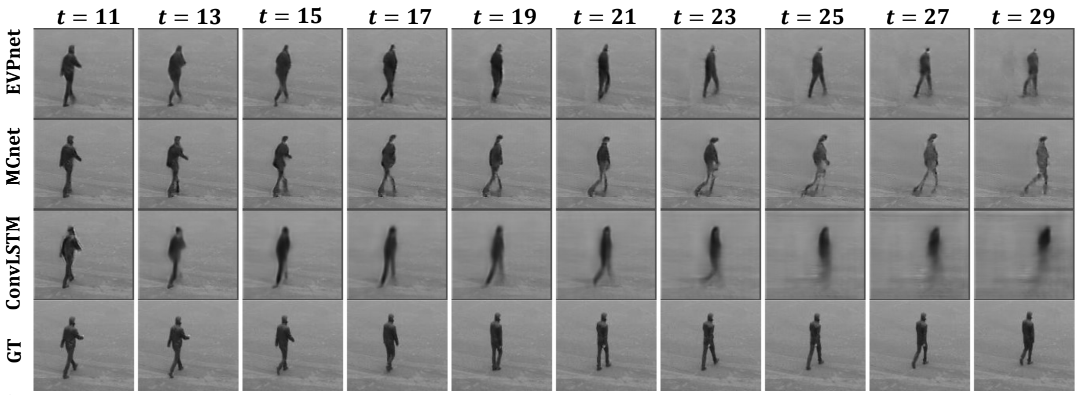

Fig. 3. The qualitative results of EVPnet, MCnet, ConvLSTM and the ground-truth on the KTH dataset.

while predict 20 frames into future when testing. We use the hyper-parameter $\lambda_{gan} = 0.02$ for KTH dataset.

Fig.2 shows the quantitative evaluation results of the state-of-the-art method MCnet [11], the baseline ConvLSTM [7], our EVPnet and its variation without skip connection. Our algorithm achieves the best average results of the predicted 20 frames in terms of both PSNR and SSIM metrics. Compared with the MCnet, our method obtains almost the same effect for the first few predicted frames but our approach achieves superior performance as time goes by. For the first few frames, MCnet gains a weak advantage than our EVPnet in quantitative evaluation because the margin between two adjacent frames is smaller than the margin from the edge to the corresponding natural image. But the image quality of the MCnet rapidly declines due to the error accumulation. Our edge guided video prediction method is more robust, so it can achieve obviously better results in long-term prediction. Compared with the no skip connection version, applying skip connection can greatly improve the quality of the first few frames. No matter with or without skip connection, our EVPnet is extremely beyond the baseline ConvLSTM.

Fig.3 presents the qualitative results of our EVPnet, MCnet and ConvLSTM on the KTH dataset. It can be seen that the frames generated by ConvLSTM quickly become blurry over time while the frames especially the leg part generated by MCnet are severely distorted. Our EVPnet achieves the most realistic and sharp prediction results over long-term time steps.

4.3. KITTI Dataset

In this section, we test the EVPnet on a more challenging KITTI autonomous driving dataset [14]. These videos contain rich temporal dynamics, including both self-motion of the vehicle and the motion of other objects in the scene, and complex background. We use videos from City, Residential and Road categories (except 2017_9_30:18,28 and 2017_10_03:27,34 four videos, because these four videos are extremely larger than others) in our experiment. We split all the 57 videos into two parts, first 12 videos used for testing and last 45 videos used for training. Frames are center-cropped and downsampled to 128×160 pixels. For KITTI dataset, all methods also observe 10 frames and predict 10 frames into the future when training while predict 20 frames into future when testing. We use the hyper-parameter $\lambda_{gan} = 0.001$ for KITTI dataset.

Table 1. The quantitative results on KITTI dataset

Method	PSNR	SSIM	Sharpness(CPBD)
ConvLSTM[7]	16.158	0.420	0.617
Prednet[4]	17.953	0.540	0.269
EVPnet	16.342	0.465	0.544

Fig.4 shows the qualitative comparison results among our EVPnet and the state-of-the-art method Prednet[4] and ConvLSTM[7]. The proposed EVPnet predicts the most realistic and sharpest frames over time steps. However, the Prednet and ConvLSTM suffer from the accumulative errors because they use the predicted frame as input recursively, thus rapidly resulting in blurring and artifacts over time. The objective comparison results of the KITTI dataset are shown in Table 1. The PSNR and SSIM of Prednet are higher than our EVPnet, while the sharpness(CPBD) index of Prednet drops by 50% compared with our EVPnet. This is because Prednet only uses the L1 loss function, which minimizes the average error of pixels but leads to visually unacceptable blur as shown in Fig4. While, our EVPnet is committed to generate visually realistic video frames, so we apply adversarial loss and GDL loss. Our EVPnet exceeds the ConvLSTM on PSNR and SSIM metric, because the frames generated by ConvLSTM have a lot of noise and artifacts. However, the noise also leads to higher sharpness index than our EVPnet. From above results, it can be seen that single objective image evaluation is not completely consistent with human visual perceptions as described in [19].

5. CONCLUSIONS

We propose an edge guided generation network for future frames prediction in natural video sequences. The proposed method decomposes the video prediction task into two steps, the edge prediction and the edge guided frame generation. Our method can be applied on any scene and resolution videos. Experiments conducted on KTH human action dataset and KITTI autonomous driving dataset show that our ap-

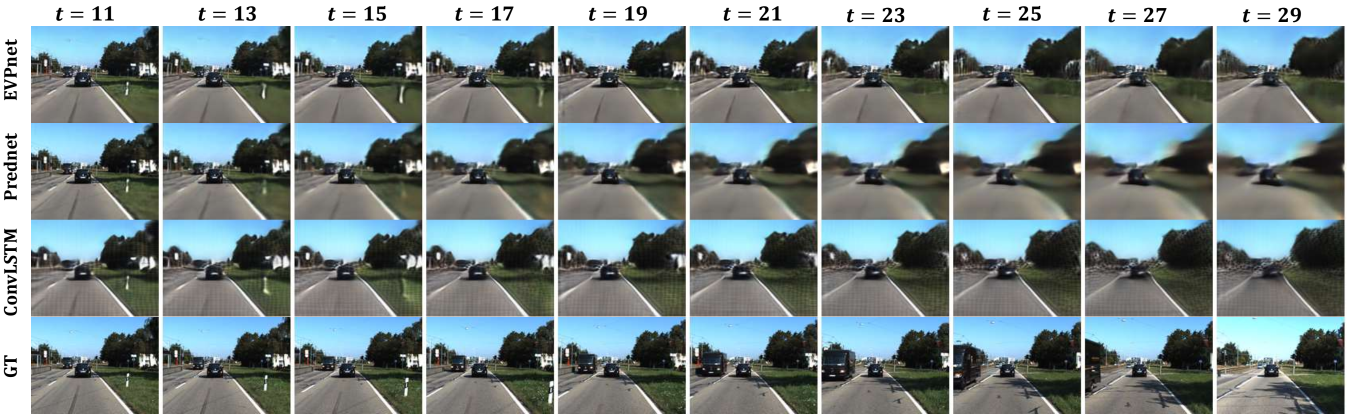

Fig. 4. The qualitative results of EVPnet, Prednet, ConvLSTM and the ground-truth on the KITTI dataset.

proach outperforms the existing state-of-the-art methods in long-term video prediction.

6. ACKNOWLEDGMENT

This work was supported in part by National Natural Science Foundation of China: 61772494, 61332016, 61620106009, U1636214, 61650202 and 61672497, in part by Youth Innovation Promotion Association CAS, in part by Key Research Program of Frontier Sciences, CAS: QYZDJ-SSW-SYS013, in part by Shandong Provincial Natural Science Foundation, China: ZR2017MF001.

References

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

[2] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *ICML*, 2015, pp. 843–852.

[3] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," in *ICLR*, 2016.

[4] W. Lotter, G. Kreiman, and D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," in *ICLR*, 2017.

[5] V. Patraucean, A. Handa, and R. Cipolla, "Spatio-temporal video autoencoder with differentiable memory," in *ICLRW*, 2016.

[6] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017.

[7] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *NIPS*, 2015, pp. 802–810.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] C. Lu, M. Hirsch, and B. Schölkopf, "Flexible spatio-temporal networks for video prediction," in *CVPR*, 2017, pp. 6523–6531.

[10] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, "Learning to generate long-term future via hierarchical prediction," in *ICML*, 2017.

[11] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, "Decomposing motion and content for natural video sequence prediction," in *ICLR*, 2017.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014, pp. 2672–2680.

[13] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *ICPR*, 2004, pp. 32–36.

[14] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[15] S. Xie and Z. Tu, "Holistically-nested edge detection," in *ICCV*, 2015, pp. 1395–1403.

[16] Z. Wang, A. C. Bovik, and H. R. et al Sheikh, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[17] N. D. Narvekar and L. J. Karam, "A no-reference image blur metric based on the cumulative probability of blur detection (cpbd)," *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2678–2683, 2011.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2014.

[19] W. Lin and C. J. Kuo, "Perceptual visual quality metrics: A survey," *Journal of Visual Communication and Image Representation*, vol. 22, no. 4, pp. 297–312, 2011.