2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

VarNet: Exploring Variations for Unsupervised Video Prediction

Beibei Jin, Yu Hu, *Member, IEEE*, Yiming Zeng, Qiankun Tang, Shice Liu, Jing Ye, *Member, IEEE*

***Abstract*— Unsupervised video prediction is a very challenging task due to the complexity and diversity in natural scenes. Prior works directly predicting pixels or optical flows either have the blurring problem or require additional assumptions. We highlight that the crux for video frame prediction lies in precisely capturing the inter-frame variations which encompass the movement of objects and the evolution of the surrounding environment. We then present an unsupervised video prediction framework — Variation Network (VarNet) to directly predict the variations between adjacent frames which are then fused with current frame to generate the future frame. In addition, we propose an adaptively re-weighting mechanism for loss function to offer each pixel a fair weight according to the amplitude of its variation. Extensive experiments for both short-term and long-term video prediction are implemented on two advanced datasets — KTH and KITTI with two evaluating metrics — PSNR and SSIM. For the KTH dataset, the VarNet outperforms the state-of-the-art works up to 11.9% on PSNR and 9.5% on SSIM. As for the KITTI dataset, the performance boosts are up to 55.1% on PSNR and 15.9% on SSIM. Moreover, we verify that the generalization ability of our model excels other state-of-the-art methods by testing on the unseen CalTech Pedestrian dataset after being trained on the KITTI dataset. Source code and video are available at `https://github.com/jinbeibei/VarNet`.**

I. Introduction

Unsupervised video prediction generates the future frames based on previous video sequences without external supervision. Computer systems that can forecast how the scene will unfold would open up new possibilities ranging from domestic service robots that can better interact with humans, autonomous cars that can self-drive more safely in cities to emergency response systems that can timely anticipate sudden accidents. In [1]–[3], video prediction is applied to self-driving cars in order to help autonomous navigation.

Fundamentally, learning such predictive models for natural videos is very challenging because of the complexity and diversity in the scenes. At the early stage, [4]–[6] attempt to make predictions by using high-level semantic information such as human actions and unusual events. Since these approaches rely on predefined semantic information, they can only provide partial descriptions for the future and have limited applications like Atari games. Recently, pixel-level approaches have been proposed. [3], [7]–[10] directly predict the entire frame by hallucinating the pixel values. Due to the complexity and diversity of the scenes, these works usually have the problem of blurring, especially for moving objects and tiny details. In order to reduce blurring, [11]–[15] explicitly model the pixel-wise motion trajectory with optical flow by using a one-stream neural network or a two-stream neural network. Although optical flow is the most commonly explored motion field, it is susceptible to failure in challenging conditions such as occlusion, fast motions, as well as abrupt illumination or nonlinear structural changes.

We point out that the key to video prediction is to accurately capture the inter-frame variations between frames which refer to the change extent of pixels between two adjacent frames and reflect the movement of objects and the evolution of the surrounding environment. Previous works that directly predict the entire frame inherently regard the pixels of a moving object contributing the same as that of the static background, leading to an averaging effect on pixels and therefore blurry predictions. On the other hand, the optical flow based approaches only capture the motion factor in the variation, hence cannot reflect the overall differences between frames. In contrast, we directly predict the inter-frame variations between adjacent frames with a Generative Adversarial Network (GAN) architecture for unsupervised prediction. The main contributions of our work are summarized as follows:

1) To the best of our knowledge, this is the first work to generate inter-frame variations instead of absolute pixel values for unsupervised video prediction. We develop a network model — VarNet to predict the inter-frame variations. Meanwhile, the current frame is directly forwarded from the input to the end of the network with a dedicated connection, which will be fused with the predicted variation map to produce the future frame.
2) An adaptively re-weighting mechanism is imported into the loss function in the purpose of highlighting the contributions of pixels to variations. A pixel with a higher variation will be given a higher weight than a pixel with a lower variation. During training, the loss function is updated by re-weighting pixel-level loss between the prediction and the ground truth.
3) Extensive experiments for long-term and short-term video prediction are implemented on advanced datasets — KTH [16] and KITTI [17], [18] with two evaluating metrics — PSNR and SSIM. For the KTH dataset, our results demonstrate a significant performance boost than the state-of-the-art up to 11.9% on PSNR and 9.5% on SSIM. As for the KITTI dataset, the performance boosts are up

*This work is supported by National Natural Science Foundation of China under Grant No. 61274030, 61532017.

The authors are with the State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing,100190, P.R. China, and are also with the University of Chinese Academy of Sciences, Beijing, 101407, P.R. China {Yu Hu is the corresponding author, phone: 86-10-62600632; fax: 86-10-62600600; e-mail:{jinbeibei, huyu, zengyiming, tangqiankun, liushice, yejing}@ict.ac.cn}

to 55.1% on PSNR and 15.9% on SSIM. Moreover, we verify that the generalization ability of our model is superior to other approaches by testing on the unseen CalTech Pedestrian dataset [19] after models being trained on the KITTI dataset.

II. Related Work

Prior video prediction works can be broadly classified into two categories: semantic-level prediction and pixel-level prediction. Among the semantic-level prediction works, [4] leverages event knowledge from a training database of videos to construct an event prediction for a given static query image. [6] and [5] propose action-conditional encoder-decoder networks to predict future frames in Atari games. Semantic-level approaches require not only extra semantic information but also fully-labeled data for training, which is very costly, so in recent years, pixel-level unsupervised prediction has attracted a lot of attentions.

The pixel-level video prediction works attempt to model the evolution of pixels over time. [7] uses an encoder LSTM to map input sequences into a fixed length representation and then decodes it via a multiple decoder LSTMs. Because of not distinguish moving pixels from static background, the model makes an averaging effect on pixels. [8] mitigates blurry prediction with three techniques including of a multi-scale architecture, a generative adversarial training method, and an image gradient difference loss function. [9] uses a deterministic prediction model to leverage scene or video similarities for predicting the visual appearance of the near future frames. However, [10] finds that there is an intrinsic ambiguity in deterministic prediction and then proposes a probabilistic prediction framework to mitigate this problem. [3] propose a PredNet architecture in which a network layer makes local predictions and forwards deviations from the local predictions to the subsequent layers and finally generates the-next frame prediction. Instead of directly predicting an entire frame, [20] learns the transformations of the past frames with a convolution neural network, and predicts the affine transformation needed for generating the next frame.

To obtain sharp video prediction, some recent works have explicitly modeled the pixel-wise motion trajectory with optical flow by using a one-stream network [11]–[13] or by using a two-stream network [14], [15]. [11] proposes a CNN-based optical flow estimation network NextFlow which is trained by a mixture of synthetic and real videos. Hence compared to previous works that can only perform well on synthetic datasets, the NextFlow can yield favorable results on real-world videos. [12] presents a fully-convolution encoder-decoder network for video frame interpolation (synthesis in-between existing frames) and extrapolation (prediction the subsequent next-frame). [13] introduces a spatio-temporal network which contains CNN-based spatial autoencoders, optical flow modules, and ConvLSTM-based temporal encoders to generate future frames. [14] is a two-stream network in which the inputs are divided into two groups of motion and content, and are encoded by separate pathways to capture motion and content independently. [15] proposes a dual GAN model in which the future-frame prediction and future-optical-flow prediction mutually help each other to synthesize new video frames.

Beside the model design, the loss function which guides the convergence of the model is also of vital importance. Existing loss functions such as mean square loss (MSE) [3], [7], [8], [10], [21] and gradient difference loss (GDL) [8] treat every pixel position equally. However, according to human experience, the pixels in the vicinity of the moving object bear much greater variations than the other locations. Thus, greater variations deserve greater attention during prediction.

Although the aforementioned works make great progress in video prediction, they still have the blurring problem or have difficulties in coping with fast motion and abruptly changed illumination. In contrast to existing works, we directly predict inter-frame variations to generate future frames and re-weight the pixels according to their contributions to variations in the loss function. The proposed model can generate sharp results even in long-term prediction, and has good generalization ability. Next, we will introduce our model in detail.

III. The VarNet Model

We use generative adversarial networks (GANs) as the top-level architecture of our model. It has been proved in [8], [22] that generative adversarial training [23] can be successfully employed for next-frame prediction. The generator network produces the variation map from the input frame sequence and then combine the predicted variation map with the latest input frame to predict the next frame. For adversarial training, we layout a discriminator network to ensure the prediction to be more realistic.

A. Generator Network

The generator network architecture is shown in Fig. 1. Let $\mathscr{X} = \{X_1, X_2, ..., X_t\}$ represent a video sequence of t input frames and $\Delta\mathscr{X} = \{\Delta X_1, \Delta X_2, ..., \Delta X_{t-1}\}$ represent the $t-1$ variation maps between adjacent input frames. The generator network aims at predicting the variation map ΔX_t between the current X_t and the next X_{t+1} by

$$\Delta X_t = g(\mathscr{X}, c) \tag{1}$$

where c represents the LSTM cell state memorizing the temporal information of input variation maps. Then the future frame X_{t+1} is obtained via element-wise summation between the predicted variation map ΔX_t and current frame X_t. For prediction of multiple frames, the generator takes the prediction $\hat{x}_{t+1}$ and the variation map between the prediction $\hat{x}_{t+1}$ and previous frame x_t as input to generate the next future frame $\hat{x}_{t+2}$, and so forth.

The generator network is an encoder-decoder framework. For the encoder as illustrated at the bottom of Fig. 1, we adopt a Convolutional LSTM architecture. We experiment with two different convolution network architectures: VGG-16 [24] and ResNet-32 [25]. In Fig. 1, we take ResNet-32 as

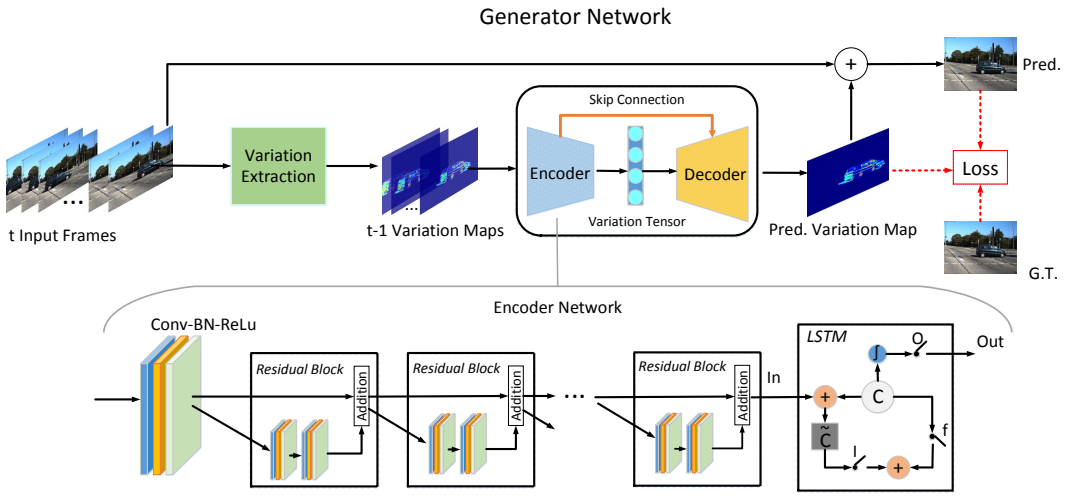

Fig. 1. The pipeline architecture of the VarNet generator network. At the beginning of the pipeline, a variation extraction module produces variation maps from the input sequence. The encoder network is implemented with ResNet-32 as an example, which is followed by a convolution LSTM to explore temporal information of variation maps. The decoder subnetwork converts the variation tensor to the predicted variation map. Skip connections are employed for fusion of the multiple scale information. The predicted variation map is then combined with the last input frame to obtain the prediction of the next frame. As to the long-term prediction, the VarNet takes existing frames and previously predicted frames as input, and then combines the currently predicted variation map with the latest predicted frame to predict the next frame, and so forth.

an example. The LSTM layer is laid on top of the last layer of ResNet-32 to memorize the high-level spatio-temporal information into variation tensor recurrently. For decoder part, We adopt the deconvolutional network which is similar to [26]. To obtain the predicted variation map which may be positive or negative, we employ $tanh(\cdot)$ as the last activation function in the decoder network. In the whole generator network, we also employ skip connections for multi-scale information fusion across layers in the encoder.

During training or testing, the $t-1$ extracted variation maps are fed into the encoder one by one. After observing the input sequence frames, the decoder transforms the variation tensor from LSTM into the prediction of variation map between the frame X_t and the frame X_{t+1}. The predicted variation map ΔX_t is then summed with the previous frame X_t to generate the predicted frame $\hat{X}_{t+1}$. The predicted frame can be fed into the generator to predicted the next frame $\hat{X}_{t+2}$, and so forth.

The output of the generator represents the difference between the current frame and the next frame to be predicted. In the variation map, the pixel that has a higher value obviously contributes more to the variation than the pixel that has a lower value, and should be given a higher weight during the training process. Based on the observation, we propose to employ a re-weighting mechanism on the content loss to highlight the pixels' contribution to the inter-frame variation, which will be detailed in the next section.

We assume that the content information between adjacent frames in videos will not change drastically. Hence the previous adjacent frame can offer very rich information for the prediction of the future frame. The model produces the next frame prediction X_{t+1} by adding the frame X_t to the generator's output — the predicted variation map. The addition operation is element-wise and computation efficient. Long-term prediction can be achieved by taking existing frames and previously predicted frames as input, and then combining the currently predicted variation map with the latest predicted frame to predict the next frame.

B. Discriminator Network

The discriminator needs to be able to distinguish real frame sequences from the predicted frame sequences. The architecture of the discriminator network in this work is a deep spatio-temporal convolution neural network similar to [15]. To make use of temporal information, the video sequences concatenated by the input $X_{1:t}$ and the future $X_{t+1:t+T}$ are fed into the discriminator network. Consequently, the generated video sequences are consistent with the input from the dataset.

The generator network and the discriminator network are two independent models and we train them alternately and iteratively. We explain the training method in the next section.

IV. Training

To avoid the instability during training the GAN, we adopt multi-module losses similar to [14]. Please note, we apply an adaptively re-weighting mechanism to the loss function design in order to consider the pixels' contribution to variations. The multi-module losses consist of the content loss and the adversarial learning loss.

Let $\mathscr{X} = \{X_1, X_2, ..., X_t\}$ represent the input sequence of t frames and $\mathscr{Y} = \{X_{t+1}, X_{t+2}, ..., X_{t+T}\}$ be the next sequence of T frames to be predicted. $\hat{\mathscr{Y}} = \{\hat{X}_{t+1}, \hat{X}_{t+2}, ..., \hat{X}_{t+T}\}$ represents the predicted sequence of T frames. Next we will introduce the content loss and the adversarial loss in detail.

A. Content Loss

Adaptively re-weighting content loss (RWL): In the natural scenes, most of the time, only a small percentage of the scene has obvious changes. In other words, a major part of the scene is consistent between adjacent frames. For human beings, we usually pay more attention to the objects which have larger motions or more significant structural changes. Therefore, pixels belonging to the changing foreground or the static background indicate different levels of importance. In our model, the elements in the predicted ΔX represent the pixel variation between adjacent frames. So we can use its absolute value to re-weight the content loss by:

$$L_{RWL}(\mathscr{Y},\hat{\mathscr{Y}}) = \sum_{i=1}^{T} \left|\left|\Delta X_{i-1}\right|(Y_i - \hat{Y}_i)\right|_p^p \tag{2}$$

where p is a hyper-parameter. The content loss guides the prediction of the VarNet to be as close as possible to the ground truth. The training process is promoted by the re-weighting operation to generate a sharp prediction.

Gradient difference loss: Moreover, we directly penalize the differences of image gradient predictions [8] to further sharpen the prediction. The GDL loss function in [8] is given by :

$$\begin{aligned} L_{GDL}(X,\hat{X}) = &\sum_{i,j} \left|\left|X^{i,j} - X^{i-1,j}\right| - \left|\hat{X}^{i,j} - \hat{X}^{i-1,j}\right|\right|^{\alpha} + \\ &\sum_{i,j} \left|\left|X^{i,j-1} - Y^{i,j}\right| - \left|\hat{X}^{i,j-1} - \hat{Y}^{i,j}\right|\right|^{\alpha} \end{aligned} \tag{3}$$

where α is a hyper-parameter greater or equal to 1, i and j represent pixel coordinates in frames. The GDL loss keeps the forecast in line with the ground truth in terms of gradient.

In summary, the whole content loss is combined by:

$$L_{content} = L_{RWL} + L_{GDL} \tag{4}$$

B. Adversarial Learning Loss

Training the discriminator D: To train the discriminator D, we need to freeze the weights of the generator G and perform the SGD optimization on the network. The goal is to classify the input sequence $\{\mathscr{X},\mathscr{Y}\}$ into real sample and $\{\mathscr{X},\hat{\mathscr{Y}}\}$ into fake sample. Thus the loss function of D and the binary cross-entropy loss L_{bce} are given by:

$$L_D = L_{bce}(D(\{\mathscr{X},\mathscr{Y}\}),1) + L_{bce}(D(\{\mathscr{X},\hat{\mathscr{Y}}\}),0) \tag{5}$$

$$L_{bce}(Y,\hat{Y}) = -\hat{Y}log(Y) + (1-\hat{Y})log(1-Y) \tag{6}$$

where Y and $\hat{Y}$ take values in $\{0,1\}$.

Training the generator G: Similarly, to train the generator G, we need to freeze the weights of the discriminator D and perform the SGD optimization on the network. The goal is to make the discriminator be not able to distinguish the real frame sequences from the generated frame sequences. The loss for the generator is defined by:

$$L_G = L_{bce}(D(\{\mathscr{X},\hat{\mathscr{Y}}\}),1) \tag{7}$$

We set thresholds for the update of the generator and the discriminator. If the discriminator's loss is lower than a threshold, we will stop the update of the discriminator. And if the discriminator's loss is higher than a threshold, we will stop the update of the generator. In other cases, the updates of the generator and the discriminator are iteratively.

C. Total Loss

The total loss of the VarNet model is defined by:

$$L = \lambda_{content} L_{content} + \lambda_G L_G \tag{8}$$

where $\lambda_{content}$ and λ_G are two hyper-parameters to balance the weight of each sub-loss.

V. Experiment

We evaluate the VarNet on three challenging datasets: the KTH [16], the KITTI [17], [18] and the CalTech Pedestrian [19] datasets. The metrics we use to measure the results are the Peak Signal to Noise Ratio (PSNR) and the Structural Similarity (SSIM) [27]. The higher values of the both metrics indicate better results. For all experiments, we set $p = 2$ in Eq. (2) and $\alpha = 1$ in Eq. (3), respectively. The experiments are conducted on the public Tensorflow platform with version 1.2 on a single NVIDIA GeForce GTX 1080.

A. PSNR and SSIM on the KTH and KITTI datasets

Experimental settings: The KTH dataset contains a total of 2391 video samples taken by 25 people in 6 different scenarios. It has 6 types of human actions: walking, jogging, running, boxing, hand waving and hand clapping. All sequences were taken over homogeneous backgrounds but were different from each other in scale, clothe and lighting. Following the previous work [14], we use person 1-16 for training and person 17-25 for testing. We resize frames into 128*128 pixels and normalize them to the range [0, 1]. We set $\lambda_{content} = 1$ and $\lambda_G = 0.02$ for training. The KITTI dataset is currently the world's largest autonomous driving benchmark. We use the raw data which was captured by a car-mounted camera. We pre-process the data into 61 recording sessions. Frames are center-cropped and down-sampled to 240*320 pixels. We set $\lambda_{content} = 1$ and $\lambda_G = 0.0001$ for training. For short-term prediction, we train the network to predict 10 future frames on KTH and 4 future frames on KITTI by observing 10 frames; for long-term prediction, we train the network to predict 40 future frames by observing 10 frames.

Results: Fig. 2 shows the comparison for long-term prediction results of the state-of-the-art method MCNet [14] and the VarNet. On both datasets, all VarNet variants demonstrate a significant performance boost than the MCNet. Specifically, the VarNet(ResNet)+RWL outperforms the MCNet by up to 11.9% on PSNR and 9.5% on SSIM for the KTH dataset, and by up to 55.1% on PSNR and 15.9% on SSIM for the more complex KITTI dataset. According to our testing, the consumption of time for the model to predict one frame is about 0.02 seconds on the KTH dataset and 0.05 seconds on the KITTI dataset.

From Fig. 2, we can see the re-weighting loss mechanism obviously has positive effects, especially for long-term

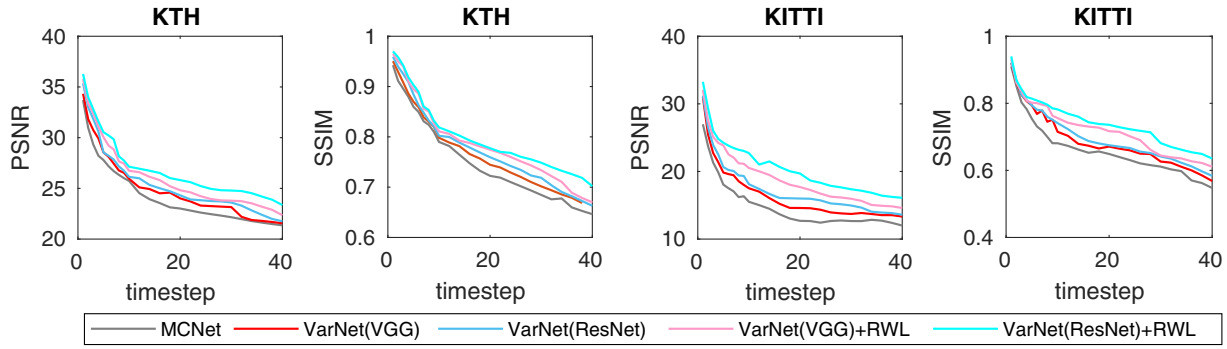

Fig. 2. Comparison of MCNet and VarNet variants. "RWL" represents the VarNet adopts the adaptively re-weighting loss mechanism during training. Left two columns: evaluation on KTH dataset. Right two columns: evaluation on KITTI dataset.

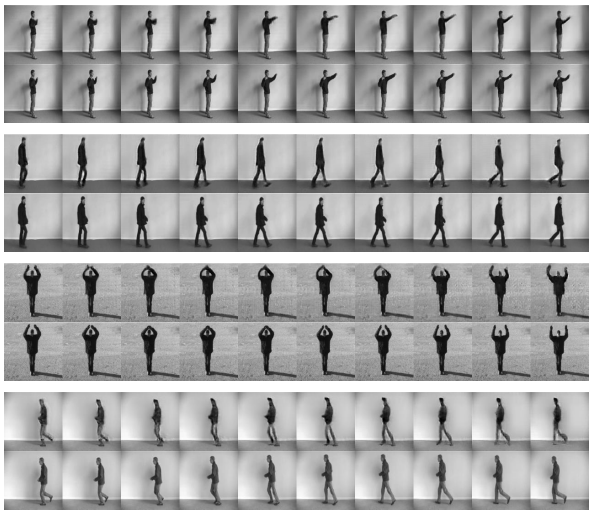

Fig. 3. 10-step short-term prediction on the KTH dataset. In each group, the first row indicates the predicted results and the second row indicates the ground truth.

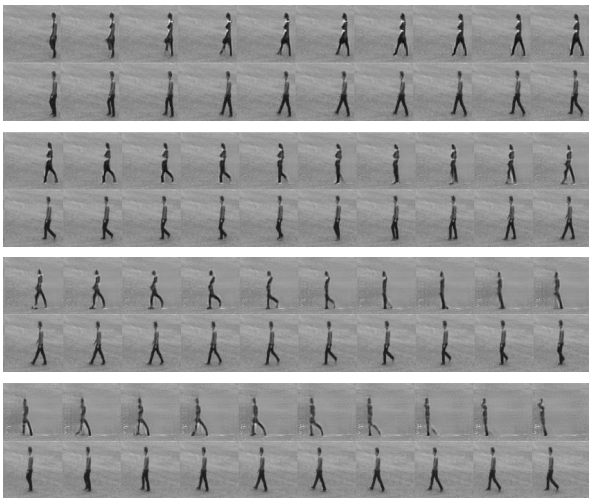

Fig. 4. 40-step long-term prediction on the KTH dataset. In each group, the first row indicates the predicted results and the second row indicates the ground truth.

prediction. The improvement of the long-term prediction is significantly higher than that of the short-term prediction, which is consistent with our observation that the pixels with larger contribution to inter-frame variations deserve more attentions than the pixels with less contribution.

B. Result visualization on the KTH and the KITTI datasets

For the sake of clarity, some short-term and long-term predictions of the VarNet(ResNet)+RWL are visualized.

Fig. 3 shows the 10-step short-term prediction. We can see the predicted frames are sharp and the tiny details are accurate. Please note that the model can predict the switch of the actions accurately — the hands of the man will separate after they are closed, as shown in the third group.

Fig. 4 shows the 40-step long-term prediction. We can see that the model predicts the change of gait accurately including the bending of the legs and the movement of the body.

From Fig. 5, we can see that the model can generate the correct gaits of the walkers and predict the moving vehicles very clearly. As shown in the second group, the network is able to correctly handle the challenging situation that the vehicle is obscured by the traffic light pole.

From Fig. 6, we can see that our model can predict the change of lanes on the road and the movement of the vehicle. For the first 20-step predictions, the cars in the scene can be predicted well. However, as time goes on, the cars (especially the red one) become blur. We think the main reason is that the object red car is so small that it takes up only a few pixels in the picture, which impacts the model's judgment of inter-frame variations.

C. Generalization ability

Experiment settings: To verify the generalization ability of the model, we test our model on the unseen CalTech Pedestrian dataset after training it on the KITTI dataset. The Caltech Pedestrian Dataset consists of approximately 10 hours of videos taken from a vehicle driving through regular traffic in an urban environment. We use the testing data from set 06 to set 10. The model is trained based on ten frames observation to predict the next frame.

Fig. 5. 4-step short-term prediction on KITTI. In each group, the first row indicates the predicted result and the second row indicates the ground truth.

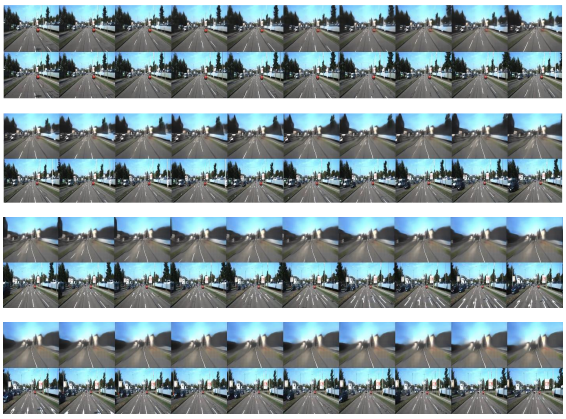

Fig. 6. 40-step long-term prediction on KITTI. In each group, the first row indicates the predicted result and the second row indicates ground truth.

TABLE I

PERFORMANCE (PSNR AND SSIM) OF VIDEO PREDICTION ON CALTECH AFTER TRAINED ON KITTI DATASET.

Method	PSNR	SSIM
CopyLast	20.996	0.762
CNN-LSTM Enc.-Dec. [13]	24.353	0.865
BeyondMSE [8]	24.867	0.881
PredNet [3]	25.044	0.884
Dual Motion GAN [15]	26.179	0.899
MCNet [14]	26.98	0.885
VarNet(VGG)+RWL	**27.802**	**0.903**
VarNet(ResNet)+RWL	**27.985**	**0.912**

Results: We compare our approach to the Copy-Last-Frame baseline and the other state-of-the-art works. The results are averaged over the test videos. As illustrated in Table I, our model outperforms all the other methods. These results indicate that this model achieves good generalization ability across different datasets.

VI. CONCLUSION

In this paper, we propose the VarNet network to directly explore the inter-frame variations for unsupervised video prediction. We further propose an adaptively re-weighting mechanism in loss function design to leverage the contributions of pixels to the variations. Extensive experiments demonstrate our model outperforms the state-of-the-arts. In the future work, we will verify the effectiveness of the VarNet on more datasets.

REFERENCES

[1] E. Santana and G. Hotz, "Learning a driving simulator," *arXiv preprint arXiv:1608.01230*, 2016.

[2] B. De Brabandere, X. Jia, T. Tuytelaars, and L. Van Gool, "Dynamic filter networks," *arXiv preprint arXiv:1605.09673*, 2016.

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

[4] J. Yuen and A. Torralba, "A data-driven approach for event prediction," *ECCV*, 2010.

[5] E. Wang, A. Kosson, and T. Mu, "Deep action conditional neural network for frame prediction in atari games," Technical Report, Stanford University, Tech. Rep., 2017.

[6] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh, "Action-conditional video prediction using deep networks in atari games," *NIPS*, 2015.

[7] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised learning of video representations using lstms," *ICML*, 2015.

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

[9] C. Vondrick, H. Pirsiavash, and A. Torralba, "Anticipating visual representations from unlabeled video," *CVPR*, 2016.

[10] J. Walker, C. Doersch, A. Gupta, and M. Hebert, "An uncertain future: Forecasting from static images using variational autoencoders," *ECCV*, 2016.

[11] N. Sedaghat, M. Zolfaghari, and T. Brox, "Hybrid learning of optical flow and next frame prediction to boost optical flow in the wild," *arXiv preprint arXiv:1612.03777*, 2016.

[12] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," *ICCV*, 2017.

[13] C. Lu, M. Hirsch, and B. Schölkopf, "Flexible spatio-temporal networks for video prediction," *CVPR*, 2017.

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

[15] X. Liang, L. Lee, W. Dai, and E. P. Xing, "Dual motion gan for future-flow embedded video prediction," *ICCV*, 2017.

[16] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," *ICPR*, 2004.

[17] R. Urtasun, P. Lenz, and A. Geiger, "Are we ready for autonomous driving? the kitti vision benchmark suite," *CVPR*, 2012.

[18] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *IJRR*, 2013.

[19] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: a benchmark," *CVPR*, 2009.

[20] J. V. Amersfoort, A. Kannan, M. Ranzato, A. Szlam, T. Du, and S. Chintala, "Transformation-based models of video sequences," *arXiv preprint arXiv:1701.08435*, 2017.

[21] C. Vondrick and A. Torralba, "Generating the future with adversarial transformers," *CVPR*, 2017.

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

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

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CVPR*, 2016.

[26] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," *ICCV*, 2011.

[27] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing (TIP)*, 2004.