Supplementary Material

Yiwei Lu[1[0000−0001−7872−3186]], Frank Yu[1[0000−0002−5620−8842]], Mahesh Kumar Krishna Reddy[1[0000−0001−5645−4931]], and Yang Wang[1,2[0000−0001−9447−1791]]

[1] University of Manitoba, [2]Huawei Technologies Canada
{luy2,kumark,ywang}@cs.umanitoba.ca

In this document, we give details of the backbone architectures used in the experiments of the paper.

1 r-GAN

This backbone architecture is based on the model in [4]. The model in [4] is built on a conditional GAN architecture with a modified U-Net [7]. Additionally, [4] uses a Flownet [1] to capture temporal information of an image sequence. To build an end-to-end model, we remove the Flownet and instead learn the spatial-temporal feature of an image sequence using a ConvLSTM module. We call our model *r-GAN*. Our proposed model consists of two major parts: a sequential image generator and a discriminator. Fig 1 shows an overview of *r-GAN*.

1.1 Generator:

We apply the same modified U-Net with [4] as the backbone of our generator $\mathcal{G}(\cdot)$. Given an image sequence $I_1, ..., I_t$ (note that we choose $t = 3$ in our case), we pass each image $I_T (T = 1, 2, ..., t)$ to the U-net to generate a prediction $\hat{I}_{T+1}$. A ConvLSTM module then takes $\hat{I}_{T+1}$ and the last hidden state h_T as input and generate the current hidden state h_{T+1}:

$$h_{T+1} = f_{ConvLSTM}(h_T, \hat{I}_{T+1}) \tag{1}$$

The hidden state in the ConvLSTM module is used to remember the previous information of an image sequence.

To learn parameters in this module, we combine the least absolute deviation (L_1 loss) [6], multi-scale structural similarity measurement (L_{ssm} loss) [8] and gradient difference (L_{gdl} loss) [5] to define a loss that measures the quality of the predicted frame:

$$L(\hat{I}_{t+1}, I_{t+1}) = L_1(\hat{I}_{t+1}, I_{t+1}) + L_{ssm}(\hat{I}_{t+1}, I_{t+1}) + L_{gdl}(\hat{I}_{t+1}, I_{t+1}) \tag{2}$$

1.2 Discriminator:

The goal of the discriminator is to differentiate the output of the generator and the ground-truth. Our discriminator in this network targets at classifying I_{T+1} as

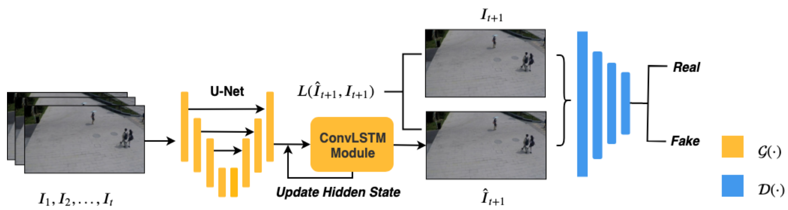

Fig. 1. An overview of our backbone architecture. Our anomaly detection model consists of a Sequential Image Generator $\mathcal{G}(\cdot)$ and a Discriminator $\mathcal{D}(\cdot)$. Given an image sequence $I_1, I_2, ..., I_t$ as the input, $\mathcal{G}(\cdot)$ outputs a prediction $\hat{I}_{t+1}$ of the next frame. A prediction loss is computed between $\hat{I}_{t+1}$ and the actual frame I_{t+1} for parameter updating. $\mathcal{D}(\cdot)$ takes both $\hat{I}_{t+1}$ and I_{t+1} as its input to classify which one is real and which one is fake. These two networks are trained adversarially to obtain a good $\mathcal{G}(\cdot)$ that is able to fool $\mathcal{D}(\cdot)$.

1 and $\hat{I}_{T+1}$ as 0. More specifically, we optimize our discriminator $\mathcal{D}(\cdot)$ according to the objective function below:

$$\begin{aligned} L^D_{adv}(\hat{I}_{t+1}, I_{t+1}) = \; & \frac{1}{2} L_{MSE}(\mathcal{D}(\hat{I}_{t+1}), 0) + \\ & \frac{1}{2} L_{MSE}(\mathcal{D}(I_{t+1}), 1) \end{aligned} \tag{3}$$

where L_{MSE} is the Mean Square Error loss function.

1.3 Anomaly Detection

Given an input sequence of frames $I_1, ..., I_t$ during testing, we use our model to predict the next frame $\hat{I}_{t+1}$ in the future. This predicted future frame $\hat{I}_{t+1}$ is compared with the ground-truth future frame I_{t+1} by calculating $L(\hat{I}_{t+1}, I_{t+1})$ (see Eq. 2). Same as [4], after calculating the overall spatial loss of each testing video, we normalize the losses to get a score $S(t)$ in the range of $[0, 1]$ for each frame in the video by:

$$S(t) = \frac{L(\hat{I}_{t+1}, I_{t+1}) - \min L(\hat{I}_{t+1}, I_{t+1})}{\max L(\hat{I}_{t+1}, I_{t+1}) - \min L(\hat{I}_{t+1}, I_{t+1})} \tag{4}$$

We then use $S(t)$ as the score indicating how likely a particular frame is an anomaly. Note that all of our variants share the same evaluation metrics.

2 r-GAN*

A possible variant of r-GAN is applying the ConvLSTM module in the latent space of an autoencoder. We call this variant *r-GAN**. The discriminator of this module is identical to that of *r-GAN*, the only difference lies in the generator. The generator uses an autoencoder as its backbone network. In our implementation, our autoencoder shares the same structure with the U-net in [4], but without the skip connections. To capture the temporal information of the sequence, we apply a ConvLSTM module to process the latent variables. Taking $I_T \in \mathbb{R}^{H \times W \times 3}$ as the input image at time T, the encoder generates a latent feature $\varphi(I_T) \in \mathbb{R}^{H' \times W' \times F}$. Here we set $H' \times W' \times F = 16 \times 16 \times 32$. We use this latent feature to generate the current hidden state h_T at time T using the ConvLSTM module:

$$h_T = f_{ConvLSTM}(\varphi(I_T), h_{T-1}) \tag{5}$$

Note that h_T and $\varphi(I_T)$ share the same dimension. By recursively updating the hidden state, the output of the ConvLSTM module is h_{t+1}. The decoder simply upsamples h_{t+1} and predict the next frame $\hat{I}_{t+1}$.

3 r-VAE

Variational autoencoder (VAE) [3] has been shown to be effective in reconstructing complex distributions. Given an input image I_T, VAE applies an encoder (also known as inference model) $q_\theta(z|I_T)$ to generate the latent variable z that captures the variation in I_T. It uses a decoder $p_\phi(\hat{I}_{T+1}|z)$ to predict the next frame given the latent variable. The inference model represents the approximate posterior using the mean μ and variance σ^2 calculated by a neural network $q_\theta(z|I_T) \sim \mathcal{N}(\mu, \sigma^2)$, where μ and σ^2 are outputs of neural networks that take I_T as the input. In our implementation, we use VGG16 as backbone architecture. A prior $p(z)$ is chosen to be a simple Gaussian distribution. Similar to *r-GAN*, the prediction $\hat{I}_{T+1}$ is then passed to a ConvLSTM module to remember temporal information:

$$h_{T+1} = f_{ConvLSTM}(h_T, \hat{I}_{T+1}) \tag{6}$$

With the constraints of distribution on latent variables, the complete objective function can be described as below:

$$L(I_{1:t}|\theta, \phi) = \sum_{1}^{T}(-KL(q_\theta(z|I_T)||p(z)) + \mathbb{E}_{q_\theta(z|I_T)}[logp_\phi(\hat{I}_{T+1}|z)]) \tag{7}$$

where $KL(q_\theta(z|I_T)||p(z))$ is the Kullback-Leibler divergence [2] between the prior and the posterior.

References

1. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: ICCV (2015)
2. Hershey, J.R., Olsen, P.A.: Approximating the kullback leibler divergence between gaussian mixture models. In: ICASSP (2007)

3. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
4. Liu, W., Luo, W., Lian, D., Gao, S.: Future frame prediction for anomaly detection–a new baseline. In: CVPR (2018)
5. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. In: ICLR (2016)
6. Pollard, D.: Asymptotics for least absolute deviation regression estimators. Econometric Theory (1991)
7. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-assisted Intervention (2015)
8. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers (2003)