Towards Generative Adversarial Networks with Clear Contour Lines

Rui Qiu[1,a)] Danilo Vasconcellos Vargas[2,b)] Kouichi Sakurai[2,c)]

Abstract: In this paper, we proposed a conception of generative adversarial network for video with an architecture that uses a method of frame difference and includes two different types of generators and two different types of discriminators. We use the architecture based on Temporal GAN. The frame difference method we use aims not only to generate the motion track of moving objects, but also clear contour lines that can be identified easily by human. Such as the outline of fingers when people in the video are waving. Furthermore, we envisage that the proposal we proposed could become the cornerstone of the way of accelerating neural network learning and training process.

Keywords: Frame Difference, Generative Adversarial Network, video generation

1. Introduction

Learning feature representation from a large amount of dataset is a task that many studies aim to solve. A good representation is useful in sundry ways. Also, the study of deep generative model develops rapidly in recent years. Not only image classification and cluster, but also image generation. There also has some notable results in image generation. As the extension of image generation, the importance of video generation goes without saying. Comparing to image generation, it is more difficult to generate video. Although the difference between image and video is just the dimension of time, it is difficult to generate changes which include the order and information. A well generated video should have meaningful changes that make it more reality and indistinguishable with real video. The generated video should have a series of continuous changes, but not sudden and unnatural changes. The difficulty also exists in variations of local part change.

Generative Adversarial Networks[5] provide the possibility of widely changing which shows the potential of training networks to do Zero-sum game. Along with development of Generative Adversarial Networks in recent years, this technology has been used in various area of computer vision. This kind of network can generate images swiftly, and the quality of images is higher and higher. if this kind of network could be studied in-depth, the prospect will be inestimable. However, it is hard to train GANs stably, sometimes the result of generator is exquisite, but usually the result is unrealistic. In image generation area, to deal with the instability of training GANs, some researches have made great advancement. These works will be introduced in section 2.

Lately, some studies challenged to make some progress on video generation by using Generative Adversarial Networks[5], Saito etc. [17] proposed Temporal GAN framework which use the method Singular Value Clipping and include two generators, one generates a series images from latent space, another one uses these images to generate video. Same as TGAN, VGAN[20] also proposed an idea that treat the video clip as a latent space point. Argue with two articles above, somebody proposed the framework MoCoGAN that sampling latent vector from results of recurrent neural network every iteration[19].

Although above methods have made some progress in video generation, the performance of generated videos is still not enough to compare with real videos. Especially the detail of generated motion, which is always ambiguous. In this paper, we put forward an idea that using frame difference to catch the motion which contains trajectory and outline. We boldly assume that by implementing frame difference into GAN, the generated videos could be more meticulous. The contents of this paper will be shown as follows: Section 2 will be related work, Section 3 will be the conception we proposed, Section 4 will be the summary and future work. In this paper, we propose a method which use the difference of each adjacent frame. Based on existing GANs model, we propose a framework that discriminators constitute by two sub networks: video discriminator and frame difference discriminator.

[1] Department of Informatics Graduate School of Information Science and Electrical Engineering, Kyushu University
[2] Department of Informatics Faculty of Information Science and Electrical Engineering, Kyushu University
[a)] qiu.rui.850@s.kyushu-u.ac.jp
[b)] vargas@inf.kyushu-u.ac.jp
[c)] sakurai@inf.kyushu-u.ac.jp

2. Related Work

2.1 VAE, GAN and GAN's improvement

The objective of VAE and GAN is basically consistent that aims to construct a model which can generate targeted data from latent variable. But the way they chose is different.

Variational autoencoder includes encoder and decoder. The function of encoder is mapping features of data distribution to the latent vector, and the decoder makes latent vector as its input, then generates features. It is worth mentioning that the encoder of VAE include two subnetworks, one is used calculating the value of mean, which add Gaussian noise to the result of encoder that make decoder has robustness with noise. Another one is used calculating variance, which could adjust intensity of noise dynamically. On the other side, a well-trained decoder can make the reconstruct process have no noise. In this way, the training process of VAE includes adversarial, only encoder and decoder updated simultaneously.

VAE uses unsupervised learning, to support generating samples there has a class of VAE models that using label which called Conditional VAE, corresponding to CVAE?, there also has a GAN model called Conditional GAN. CGAN will be introduced in following subsection.

The Generative Adversarial Network include two subnetworks that always compete. One is called generator which generates samples from input noise, another is called discriminator which is trained to distinguish generated data from real data. Both of networks have multilayer, which include convolutional/deconvolutional layers and fully-connected layers. As the first GAN, the structure of GAN model is relatively simple, which both of generator and discriminator use fully connected neural networks. The training process of GANs is shown in Figure1. This structure could deal with image generation problem which is like the hand writing digits MNIST[11] dataset. There already have some extended variation about GANs. Arjovsk[2] proposed Wasserstein GAN which used Earth-Mover distance to solve instability of GAN. LS-GAN[15] was put forward against overfitting problem. This model could generate some good images without using batch normalization. Lecun[21] proposed EBGAN which based on energy. Comparing with GAN, EBGAN shows stability through training process and generates clearer images. Combined the advantages of WGAN and EBGAN, Google[3] proposed BEGAN which made generator and discriminator balance each other.

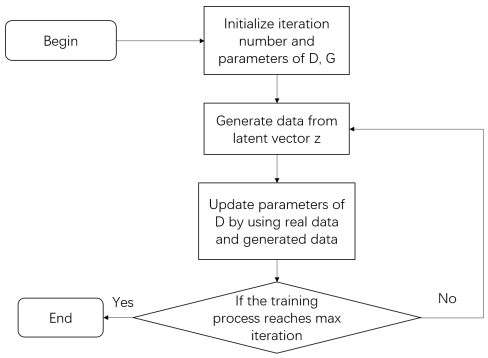

Fig. 1 The training process of Generative Adversarial Networks.

2.2 Image generation with Generative Adversarial Networks

To control the output of GAN, Mirza[13] proposed CGAN[13] which put extra information into the generator to guide training process of generator and discriminator. It is worth mentioning that CGAN first turns unsupervised learning into supervised learning. CGAN can generate images which contain specific attribute. Although the generated images have many defects, the idea of CGAN inspired follow-up research. Different from CGAN, ACGAN[14] also put classification information into discriminator that makes generator generate grab the feature more quickly. InfoGAN[4] proposed a method that make the generating process is controllable, at the same time, the generated result is also interpretable. Pix2pix[7],DiscoGAN[9] and cycle-GAN[22] make a progress that could transfer origin image into another domain. Pix2pix is different from DiscoGAN and cycle-GAN that pix2pix uses supervised learning. The input of Pix2pix generator is image, which is different from other GAN structures. Cycle-GAN can transform photos into oil paintings using unsupervised learning which pix2pix cannot do. DiscoGAN can learn the connect between cross domains without label and image pairing. Such as DiscoGAN can transform handbag images into shoe images. Another noticeable work is provided by NVIDIA[8], which uses progressive growing way to make computer generate the size of 1024×1024 images. This work achieved transition from low resolution to high resolution that can generate HD model smoothly. Because of the progressive growing method, the training process is slower compared to other GAN models.

2.3 Video Generation with Generative Adversarial Networks

Although video generation is not a recently raised topic, using GAN to generate video is still rare to see. Mathiew[12] first focus on video generation. They proved that using GAN could generate clearer frame. However, based on a series of frames, the generator just generates last one frame. To video generation, it is not enough. Apart from this study, Vondrick[20] made a great progress on video generation. This study divides the input into foreground and background two parts, which made model focus on the change of foreground dynamic. The model they proposed called VGAN could generate 32 frames 64x64 pixels video. The content of video includes golf course, beach, train station and newborn. About 20 percent markers could not identify the authenticity. Like VGAN, TGAN[17] also aimed to generate a fixed length video, except generate dy-

namic background. To this end, TGAN proposed a clipping method called singular value clipping and modified the structure of generator. Tulyakov[19] proposed MoCoGAN that could generate same content but different motion video as well as different content but same motion video. MoCoGAN decompose the latent vector which generated by RNN into content part and motion part and modify the scheme that includes image discriminator and video discriminator. The image discriminator discriminates the image sampled from generated video, video discriminator discriminate the video fragment sampled from generated video. The whole framework is based on GAN.

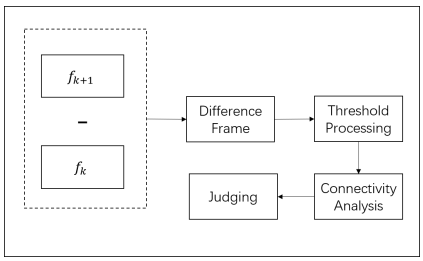

Fig. 2 The processing of temporal difference method.

3. Conception

3.1 Frame difference method

Although frame difference is not a new topic in computer vision, it is still rarely used in GANs video generation. The method of frame difference is usually used by motion detection, which could focus on the moving object by pixel-based difference. If there has no moving object, the changing of continuous frames would be hard to observe, on the other hand, if the object is moving, there will be an obviously changing between two frames. Frame difference method contains two ways which are temporal difference and three-frame differencing. Both of these methods use absolute value of frame difference to judge if an object is moving. First, we want to show temporal difference method. Suppose f_k to be the value of the k^{th} frame, f_{k+1} is the value of $(k+1)^{th}$ frame. We define the frame differential as follows:

$$D_{k+1}(x,y) = |f_{k+1}(x,y) - f_k(x,y)| \tag{1}$$

For each pixel, using binarization to gain the binarized frame, if the grayscale value of a pixel is 0, then the pixel would be a point of background, otherwise the pixel would be a point of moving object. After analyzing connectivity of the binarized frame, we can get the frame which includes complete moving object. Figure2 shows the processing of frame difference method. The equation could be written as follows (T represents for the threshold):

$$F_{d_{k+1}}(x,y) = \begin{cases} 255, D_{k+1}(x,y) \geq T \\ 0, else \end{cases} \tag{2}$$

Next let us see the three-frame difference method. The process of this method is shown in Figure4. Suppose f_{k-1} to be the value of the $(k-1)^{th} frame$, f_k to be the value of the k^{th} frame, f_{k+1} is the value of $(k+1)^{th}$ frame. Then calculate difference between two adjacent frames respectively.

$$\begin{cases} D_{k,k-1}(x,y) = |f_k(x,y) - f_{k-1}(x,y)| \\ D_{k+1,k}(x,y) = |f_{k+1}(x,y) - f_k(x,y)| \end{cases} \tag{3}$$

After obtaining difference between frames, using threshold T to binarize them.

$$\begin{cases} b_{(k,k-1)}(x,y) = \begin{cases} 255, D_{k,k-1}(x,y) \geq T \\ 0, else \end{cases} \\ b_{(k+1,k)}(x,y) = \begin{cases} 255, D_{k+1,k}(x,y) \geq T \\ 0, else \end{cases} \end{cases} \tag{4}$$

Last, let binarized frames do logic "and", the equation is shown as below:

$$F_{d_k}(x,y) = \begin{cases} 255, b_{(k,k-1)}(x,y) \cap b_{(k+1,k)}(x,y) = 1 \\ 0, b_{(k,k-1)}(x,y) \cap b_{(k+1,k)}(x,y) \neq 1 \end{cases} \tag{5}$$

3.2 VAE and GAN

Variational Autoencoder[10] was proposed by Kingma et al., the purpose of VAE is to generate data from high-dimensional features which come from data samples $X_1, \cdots, X_n$. Figure5 shows the structure of VAE. The distribution could be written as follows:

$$p(Z) = \sum_X p(Z \mid X)\, p(X) \tag{6}$$

To avoid there has no noise, VAE makes all $p(Z \mid X)$ near to normal distribution. This action promised the generation ability of model. If all of $p(Z \mid X)$ is close to normal distribution $N(0, I)$, Equation6 could be written as follows:

$$p(Z) = \sum_X N(0,I)p(X) = N(0,I)\sum_X p(X) = N(0,I) \tag{7}$$

To reach the $N(0, I)$, the means and variance need to be close to 0 as much as possible. Then the loss function will be like equation as follows:

$$\mathfrak{L}_{\mu,\sigma^2} = \mathfrak{L}_\mu + \mathfrak{L}_{\sigma^2} \tag{8}$$

$$\mathfrak{L}_\mu = \frac{1}{2}\sum_{i=1}^{d} \mu_{(i)}^2 = \frac{1}{2}|f_1(X)|^2 \tag{9}$$

$$\mathfrak{L}_{\sigma^2} = \frac{1}{2}\sum_{i=1}^{2} (\sigma_{(i)}^2 - \log \sigma_{(i)}^2 - 1) \tag{10}$$

At last we can derive that Kullback-Leibler divergence is shown as:

$$KL(N(\mu,\sigma^2)\|N(0,1)) = \frac{1}{2}(-\log\sigma^2 + \mu^2 + \sigma^2 - 1) \tag{11}$$

Generative Adversarial Networks was first put forward by Goodfellow[5], the initial structure of GANs includes two subnetworks, one is a generator and another one is a discriminator. The purpose of generator is to generate data (such as image etc.) which are lifelike. On the other hand, the purpose of discriminator is to differentiate data

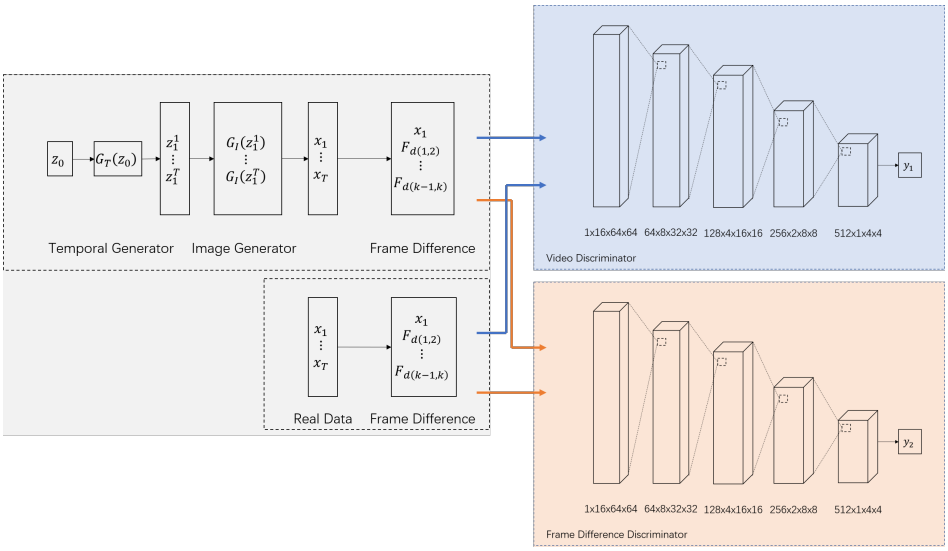

Fig. 3 Structure of our proposal. Left side is Temporal Generator G_T and Image Generator G_I, right side is Video Discriminator D_V and Frame Difference Discriminator D_F. Where z_0 represents for the vector of latent space. The vector $[x_1, F_{d(1,2),\cdots,F_{d(k-1,k)}}]$ represents for frame difference.

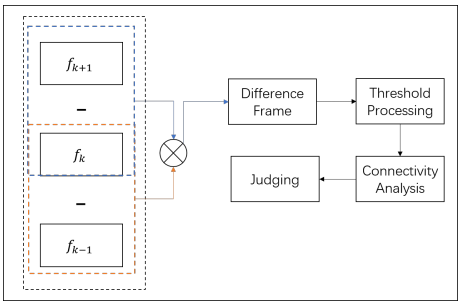

Fig. 4 The processing of three-frame difference method.

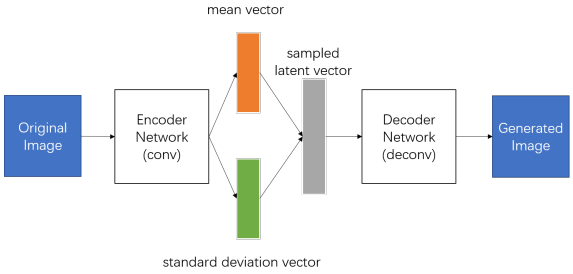

Fig. 5 The structure of VAE, which includes encoder that calculate mean and variance from input image. The function of decoder is to generate image from latent vector.

whether generated by generator. The whole network looks like that two subnetworks are playing Zero-sum game. Assume input data as $z \in \mathbb{R}^K$, the function of generator G is to generate data $x \in \mathbb{R}^M$ that like real data from latent space. The function of discriminator D is to discriminate data whether is from real dataset or generator. The output of discriminator is the probability of the input is real. Discriminator network could be expressed as:$\mathbb{R}^M \rightarrow [0, 1]$. The training process of GAN needs to solve the problem that search for the discriminator parameters which could maximize accuracy of classification, and search for the generator parameters which could deceive discriminator as far as possible. Although the training process of generator and discriminator is synchronous, the time of updating parameters is not. Ideally, when the parameters of discriminator are updated, the parameters of generator are fixed.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data(x)}}[\log D(x)] + \mathbb{E}_{x\sim p_{z(G)}}[\log(1 - D(G(z)))] \quad (12)$$

Where $p_{data(x)}$ represents for the distribution of real data from dataset, and $p_{z(G)}$ represents for the distribution of data generated by generator. Let $\boldsymbol{p}$ be the result probability of discriminator given. When $\boldsymbol{p}_{data}(x) = \boldsymbol{p}_g(x)$, which the probability of real data equal to the probability of generated data. Discriminator gives all of input data sampled from x prediction 0.5. However, the discriminator always cannot be trained to be optimal. Actually, the parameters of generator are updated with the parameter of discriminator at the same time. Besides, for the reason of various aspects[16][18][1],the training of GAN often is unstable. Some of reasons are shown below.

- Mode collapse: Despite the diversity of inputs, the generator always generates similar samples.
- Diminished gradient: The discriminator is trained too optimal that generator could learning nothing.
- Non-convergence: The parameters of model are difficult to converge.

The training process of GAN introduced Kullback-Leibler divergence and Jensen-Shannon divergence to measure the similarity. But this method could cause above problems. To solve these problems, on the foundation of GAN, Arjovsky et al. proposed WGAN[2] which uses approximation of the Wasserstein distance (Earth-Mover distance) as an alternative cost function. The alternative cost function which could avoid vanishing gradients problem is k-Lipschitz continuous, and could be applied by weight clip-

ping the discriminator parameters. The follow up research[6] also recommended it is better than clip parameters that penalize the norm of discriminator gradients with respect to data samples in training process. The training of WGAN could be written as follows:

$$\min_G \max_D \mathbb{E}_{x\sim p_{data}}[D(x)] - \mathbb{E}_{z\sim p_G}[D(G(z))] \tag{13}$$

VGAN[20] chose to divide frame sequences into two streams, one includes the moving object, the other includes background. VGAN manages generator by combining foreground and background as follows:

$$G_2(z) = m(z) \odot f(z) + (1 - m(z)) \odot b(z) \tag{14}$$

Where $G_2(z)$ represents for two-streams architecture generator, $m(z)$ represents for a spatio-temporal mask.$f(z)$ and $b(z)$ represent for foreground and background model respectively. Then, Equation (12) could be rewritten as follows:

$$\begin{aligned}\min_G \max_D \mathbb{E}_{x\sim p_x(x)}[\log D(x)] + \\ \mathbb{E}_{x_0\sim p_{x_0}(x_0)}[\log(1 - D(G(x_0)))] + \\ \mathbb{E}_{x_0\sim p_{x0}(x_0)}\left[\lambda\|x_0 - G^0(x_0)\|_2^2\right]\end{aligned} \tag{15}$$

In this equation, x_0 represents for the first frame, G^0 is the first frame of generated video, λ is a hyperparameter. Based on WGAN, Saito et al. proposed Temporal

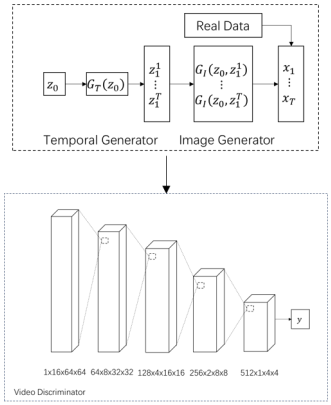

Fig. 6 The structure of Temporal GAN which includes Temporal GeneratorG_T that generate a series of latent variables$[z_1^1, \cdots, z_1^T]$, Image Generator G_I usesz_0 and $[z_1^1, \cdots, z_1^T]$ generate consequent frames, and Video Discriminator distinguishes the input data whether is generated or not.

GAN. Because WGAN needs discriminator to satisfy the condition of k-Lipschitz, the training process of WGAN has sensitivity to a hyperparameter which is related to k, TGAN also has this problem. To deal with this, they put forward the method called Singular Value Clipping. Assume in the k-Lipschitz condition, $k = 1$, then the discriminator must satisfy 1-Lipschitz condition, here comes the method of Singular Value Clipping. In linear layer, SVC replaces all the singular values which are larger than 1 with 1, then reconstruct the parameters with these changed values. In convolution layer, the weight parameter matrix $\widetilde{\boldsymbol{W}}$ also was clipped by SVC. TGAN[17] increase one generator $G_0$to generate $T\,(T > 0)$ frames data from latent space:$\mathbb{R}^{K_0} \to \mathbb{R}^{T\times K_1}$, where $z_0 \in \mathbb{R}^{K_0}$ and generated data is $\left[z_1^1, \cdots, z_1^T\right]$. And then use G_1 to generate whole video:$\mathbb{R}^{K_0} \times \mathbb{R}^{K_1} \to \mathbb{R}^M$,the generated video is represented as $\left[G_1\left(z_0, z_1^1\right), \cdots G_1\left(z_0, z_1^T\right)\right]$. Then Equation (13) could be rewritten as follows:

$$\begin{aligned}\min_{G_0,G_1} \max_D \mathbb{E}_{[x^1,\cdots x^T]\sim p_{data}}\left[D\left(\left[x^1, \cdots, x^T\right]\right)\right] - \\ \mathbb{E}_{z_0\sim p_{G_0}}\left[D\left(\left[G_1\left(z_0, z_1^1\right), \cdots, G_1\left(z_0, z_1^T\right)\right]\right)\right]\end{aligned} \tag{16}$$

One of the applications of TGAN is to generate the intermediate frame between adjacent frames, and if the labels of videos were given, TGAN could generate videos with higher quality. Last, let us see MoCoGAN[19]. Differ-

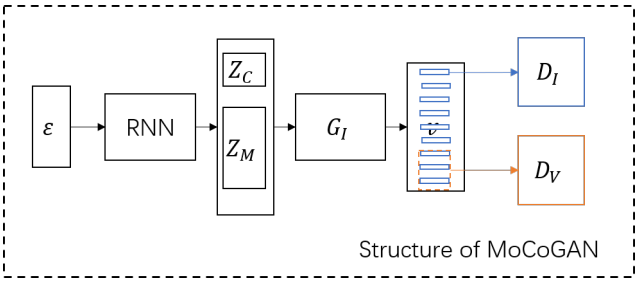

Fig. 7 The structure of MoCoGAN which includes RNN and Image Generator, Image Discriminator and Video Discriminator. The function of RNN is to generate latent vector $z = [z_C, z_M]$, C represents for the content part and M represents for the motion part. After generating images, two discriminators choose 1 frame and T continuous frames respectively to discriminate if the data is from dataset.

ent from other GAN models, the latent vectors$[z_1, \cdots, z_K]$ are produced by recurrent neural network. MoCoGAN chose one-layer GRU network as the RNN. Because of video includes not only images, but also temporal information, the generated video should include motion part and content part. Each latent vector includes same content part and different motion part, which could be expressed as:$[[z_C, z_M^1], \cdots, [z_C, z_M^K]]$. After generating by G_I, these latent vectors become images which are regarded as a frame from video clips. The whole frames form a K-length video $\widetilde{\boldsymbol{v}} = [\widetilde{\boldsymbol{x}}^1, \cdots, \widetilde{\boldsymbol{x}}^K]$. Then generated video and real video would be sampled randomly by S_1 and S_T. S_1 chose one frame from whole video, S_T chose consequent T frame as a video fragment. The chosen samples S_1 and S_T are token as input of discriminators D_I and D_V respectively. Figure7 shows the structure of MoCoGAN. This model uses the output of RNN as input of generator, after generator generating the video, images and video are input in different discriminator respectively. Therefore, Equation (12) was rewritten as follows:

$$\max_{G_1,R_M} \min_{D_I,D_V} F_V(D_I, D_V, G_I, R_M) \tag{17}$$

The function $F_V(D_I, D_V, G_I, R_M)$ is

$$\mathbb{E}_{\boldsymbol{v}}[-\log D_I(S_1(\boldsymbol{v}))] + \mathbb{E}_{\widetilde{\boldsymbol{v}}}[-\log(1 - D_I(S_1(\widetilde{\boldsymbol{v}})))] + \mathbb{E}_{\boldsymbol{v}}[-\log D_V(S_T(\boldsymbol{v}))] + \mathbb{E}_{\widetilde{\boldsymbol{v}}}[-\log(1 - D_V(S_T(\widetilde{\boldsymbol{v}})))] \quad (18)$$

Where $\mathbb{E}_{\boldsymbol{v}}$ and $\mathbb{E}_{\widetilde{\boldsymbol{v}}}$ are short hands for $\mathbb{E}_{\boldsymbol{v}\sim p_v}$ and $\mathbb{E}_{\widetilde{\boldsymbol{v}}\sim p_{\widetilde{v}}}$ respectively. S_1 means the function takes a video clip and outputs a random frame. S_T outputs T consecutive frames.

3.3 New structure of GAN

Table 1 Network configuration of generators

Temporal Generator	Image Generator
$z_0 \in \mathbb{R}^{1\times 100}$	$z_1^t \in \mathbb{R}^{100}$
$deconv(1, 512, 0, 1)$	$linear(256 \cdot 4 \cdot 4)$
$deconv(4, 256, 1, 2)$	$deconv(1, 256, 0, 1)$
$deconv(4, 128, 1, 2)$	$deconv(4, 128, 1, 2)$
$deconv(4, 128, 1, 2)$	$deconv(4, 64, 1, 2)$
$deconv(4, 100, 1, 2)$	$deconv(4, 32, 1, 2)$
$tanh$	$deconv(3, 1, 1, 1) + tanh$

[1] The second row represents the input. "linear"means linear layer, "deconv"means deconvolutional layer. The parameters in the deconvolution layer are expressed as ($kernelsize$, $outputchannels$, $padding$, $strides$).

Table 2 Network configuration of discriminators

Video Discriminator	Frame Difference Discriminator
$G_I(z_1^1), \cdots, G_I(z_1^T)$	$F_d \in \mathbb{R}^{100\times T}$
$conv(4, 64, 1, 2)$	$conv(4, 64, 1, 2)$
$conv(4, 128, 1, 2)$	$conv(4, 128, 1, 2)$
$conv(4, 256, 1, 2)$	$conv(4, 256, 1, 2)$
$conv(4, 512, 1, 2)$	$conv(4, 512, 1, 2)$
$2Dconv(4, 1, 0, 1)$	$2Dconv(4, 1, 0, 1)$

[1] The second row represents the input. "conv" means 3D convolutional layer except last layer. The parameters in the convolution layer are expressed as ($kernelsize$, $outputchannels$, $padding$, $strides$).

The new GAN framework we proposed which is based on TGAN[17] and MoCoGAN[19]. The structure of this new structure GAN includes two generators: image generator and temporal generator. And the discriminator subnetwork also includes two discriminators: frame difference discriminator and video discriminator. Table 1 ,2shows the setting of network. In new GAN framework, the length of generated video is fixed as $T(T > 0)$. The purpose of temporal generator $G_T$is to generate latent variables which expressed as $\boldsymbol{z}_1 = [z_1^1, \cdots, z_1^T]$ from the origin latent variable z_0. Then we use the generated latent variables $\boldsymbol{z}_1$ as the input of image generator. The reason why we not use (z_0, z_1^k) as the input of image generator is that image generator do not need temporal information. The temporal information is used as the input of frame difference discriminator, which will give the criticism to temporal generator after updating. The represent of generated video is written as $[G_I(z_1^1), \cdots, G_I(z_1^T)]$. Above is the introduction of generators, two discriminators both of video discriminator D_V and frame difference discriminator $D_F$play critic role to criticize G_T and G_I. Both of D_F and D_V based on a spatio-temporal 3D convolutional layers model. D_V is trained to determine if a video is from real video dataset or not. D_F is trained to determine if frame difference is from generated video or not. The frame difference could show whether the frame includes moving object. And if the moving object is included in the frame, the frame difference could show the detail of each motion. Therefore, the generated video could be naturally with details. Frame difference of whole video is expressed as F_d, which comes from the result of G_I. After generating frames from latent vector z_0, using the binarized frame as the input of Frame Difference Discriminator. Although D_V is enough for trainging two generators theoretically, it should be more quickly by increasing a discriminator.

Based on the equation13, which is using Earth-Mover Distance, the network works with the following function, which is rewritten as:$F(D_F, D_V, G_I, G_T)$.

$$F(D_F, D_V, G_I, G_T) = \mathbb{E}_{\boldsymbol{v}}[-D_F(F_d(\boldsymbol{x}))] - \mathbb{E}_{\widetilde{\boldsymbol{x}}}[(1 - D_F(F_d(\widetilde{\boldsymbol{x}})))] + \mathbb{E}_{\boldsymbol{v}}[-D_V(G_I(\boldsymbol{x}))] - \mathbb{E}_{\widetilde{\boldsymbol{v}}}[(1 - D_V(G_I(\widetilde{\boldsymbol{x}})))] \quad (19)$$

Where $\mathbb{E}_{\boldsymbol{v}}$ means expectation of generated video distribution, $\mathbb{E}_{\widetilde{\boldsymbol{v}}}$ means expectation of video from real dataset distribution.

4. Summary and future work

In this paper, we proposed a new structure GAN which can generate a fixed length video, and applied frame difference method into Generative Adversarial Networks. We constructed discriminators that can distinguish whether the video or the frame difference are generated or not. We assume that applying frame difference method could increase the accuracy of video generation.

Future work will be do more experiments to verify the effective of the model. Basing on experiment, it is necessary to adjust parameters and network configurations. And it is also needed to deal with the length of generated video. In this paper we proposed the idea that could generate fixed length video, the follow-up work will be generating videos which is not fixed. Because of the idea is based on WGAN, we also consider the method to deal with sensitivity to hyperparameter of our model.

Acknowledgments This work was supported by JST, ACT-I Grant Number JP-50166, Japan.

References

[1] Arjovsky, M. and Bottou, L.: Towards principled methods for training generative adversarial networks, arXiv preprint arXiv:1701.04862 (2017).
[2] Arjovsky, M., Chintala, S. and Bottou, L.: Wasserstein generative adversarial networks, International Conference on Machine Learning, pp. 214–223 (2017).
[3] Berthelot, D., Schumm, T. and Metz, L.: BEGAN: boundary equilibrium generative adversarial networks, arXiv preprint arXiv:1703.10717 (2017).
[4] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I. and Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets, Advances in neural information processing systems, pp. 2172–2180 (2016).
[5] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-

Farley, D., Ozair, S., Courville, A. and Bengio, Y.: Generative adversarial nets, Advances in neural information processing systems, pp. 2672–2680 (2014).
[6] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. and Courville, A. C.: Improved training of wasserstein gans, Advances in Neural Information Processing Systems, pp. 5767–5777 (2017).
[7] Isola, P., Zhu, J.-Y., Zhou, T. and Efros, A. A.: Image-to-image translation with conditional adversarial networks, arXiv preprint (2017).
[8] Karras, T., Aila, T., Laine, S. and Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation, arXiv preprint arXiv:1710.10196 (2017).
[9] Kim, T., Cha, M., Kim, H., Lee, J. K. and Kim, J.: Learning to discover cross-domain relations with generative adversarial networks, arXiv preprint arXiv:1703.05192 (2017).
[10] Kingma, D. P. and Welling, M.: Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).
[11] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P.: Gradient-based learning applied to document recognition, Proceedings of the IEEE, Vol. 86, No. 11, pp. 2278–2324 (1998).
[12] Mathieu, M., Couprie, C. and LeCun, Y.: Deep multi-scale video prediction beyond mean square error, arXiv preprint arXiv:1511.05440 (2015).
[13] Mirza, M. and Osindero, S.: Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784 (2014).
[14] Odena, A., Olah, C. and Shlens, J.: Conditional image synthesis with auxiliary classifier gans, Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, pp. 2642–2651 (2017).
[15] Qi, G.-J.: Loss-sensitive generative adversarial networks on lipschitz densities, arXiv preprint arXiv:1701.06264 (2017).
[16] Radford, A., Metz, L. and Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434 (2015).
[17] Saito, M., Matsumoto, E. and Saito, S.: Temporal generative adversarial nets with singular value clipping, IEEE International Conference on Computer Vision (ICCV), Vol. 2, No. 3, p. 5 (2017).
[18] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. and Chen, X.: Improved techniques for training gans, Advances in Neural Information Processing Systems, pp. 2234–2242 (2016).
[19] Tulyakov, S., Liu, M.-Y., Yang, X. and Kautz, J.: Mocogan: Decomposing motion and content for video generation, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1526–1535 (2018).
[20] Vondrick, C., Pirsiavash, H. and Torralba, A.: Generating videos with scene dynamics, Advances In Neural Information Processing Systems, pp. 613–621 (2016).
[21] Zhao, J., Mathieu, M. and LeCun, Y.: Energy-based generative adversarial network, arXiv preprint arXiv:1609.03126 (2016).
[22] Zhu, J.-Y., Park, T., Isola, P. and Efros, A. A.: Unpaired image-to-image translation using cycle-consistent adversarial networks, arXiv preprint (2017).