The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Hierarchical Video Generation from Orthogonal Information: Optical Flow and Texture

Katsunori Ohnishi*
The University of Tokyo
ohnishi@mi.t.u-tokyo.ac.jp

Shohei Yamamoto*
The University of Tokyo
yamamoto@mi.t.u-tokyo.ac.jp

Yoshitaka Ushiku
The University of Tokyo
ushiku@mi.t.u-tokyo.ac.jp

Tatsuya Harada
The University of Tokyo
RIKEN
harada@mi.t.u-tokyo.ac.jp

Abstract

Learning to represent and generate videos from unlabeled data is a very challenging problem. To generate realistic videos, it is important not only to ensure that the appearance of each frame is real, but also to ensure the plausibility of a video motion and consistency of a video appearance in the time direction. The process of video generation should be divided according to these intrinsic difficulties. In this study, we focus on the motion and appearance information as two important orthogonal components of a video, and propose Flow-and-Texture-Generative Adversarial Networks (FTGAN) consisting of FlowGAN and TextureGAN. In order to avoid a huge annotation cost, we have to explore a way to learn from unlabeled data. Thus, we employ optical flow as motion information to generate videos. FlowGAN generates optical flow, which contains only the edge and motion of the videos to be begerated. On the other hand, TextureGAN specializes in giving a texture to optical flow generated by FlowGAN. This hierarchical approach brings more realistic videos with plausible motion and appearance consistency. Our experiments show that our model generates more plausible motion videos and also achieves significantly improved performance for unsupervised action classification in comparison to previous GAN works. In addition, because our model generates videos from two independent information, our model can generate new combinations of motion and attribute that are not seen in training data, such as a video in which a person is doing sit-up in a baseball ground.

Introduction

Video understanding is a core problem in computer vision. Given the considerable progress in video recognition (e.g., action classification, event detection), video generation and unsupervised learning of video representation (e.g., future frame prediction) have been gaining considerable attention. Automatic video generation can potentially help human designers and developers to convert their high-level concepts into pixel-level videos. Some works (Vondrick, Pirsiavash, and Torralba 2016; Saito and Matsumoto 2016) have tried to generate videos with generative adversarial networks (GANs) (Goodfellow et al. 2014) approach.

However, video generation is a very challenging task. The difficulties lie not only on that (a) each frame should be a realistic image as generated image, but also that (b) the same scene and foreground should be generated in each video and that (c) the generated video should have plausible motion.

In this study, we focus on the above-mentioned fundamental difficulties of generating a realistic video and propose the generation of videos through GAN spitting into two orthogonal information types: motion and appearance. Because previous GANs for video (Vondrick, Pirsiavash, and Torralba 2016; Saito and Matsumoto 2016) have not focused on these difficulties, they can not produce motion realistic videos.

Unsupervised video generation should learn without any annotation. Thus, we utilize optical flow as motion information. Calculating optical flow does not require annotations on each dataset and can be generally obtained with unsupervised method, which means our method still can be trained in an unsupervised manner.

Our model, Flow and Texture Generative Adversarial Networks (FTGAN), consists of two GANs: FlowGAN and TextureGAN. We first generate optical flow with FlowGAN, and then convert optical flow into RGB videos with TextureGAN. This hierarchical approach is explained in detail below. The characteristics of optical flow are as follows: it has edges of moving objects, contains time-directional continuity, and does not contain texture information. Therefore, optical flow generation is much easier than RGB video generation. Optical flow generation first assures and achieves reasonable motion and rough edges for generated videos. Next, TextureGAN colors the optical flow supplementing rough outlines. TextureGAN provides texture information to the generated optical flow while maintaining scene and foreground consistency. Thus, we can obtain more realistic videos containing plausible motion.

As an application for video generation, it is important for a model to express a wide range of datasets. However, in the existing video generation, the appeance and motion are limited to the combinations that exist in the dataset. Since motion and appearance information are learnt separately, our model can control the appearance and motion information of a generated video independently. Thus, our model can generate nonexisting motion and attribute combination in the training dataset comprising a video in which a person is do-

*indicates equal contribution.

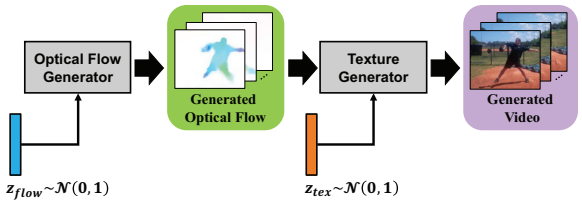

Figure 1: Generative pipeline of our FTGAN.

ing sit-up in a baseball ground.

It is also known that GAN can be used as an unsupervised feature extractor. Thus, as well as previous generative adversarial networks for video, we also perform motion recognition experiments. On the action recognition dataset UCF101 (Soomro, Zamir, and Shah 2012), our method has achieved significantly improved accuracy over previous works (Vondrick, Pirsiavash, and Torralba 2016; Saito and Matsumoto 2016).

Related Work

Image generation by using Generative Adversarial Networks (GANs) (Goodfellow et al. 2014; Radford, Metz, and Chintala 2016) has become increasingly popular. For example, in exploiting GAN, Pix2Pix (Isola et al. 2017) has succeeded in converting images with the same edges between input and target by using a U-net (Ronneberger, Fischer, and Brox 2015). For indoor-scene image generation, Style and Structure GAN (S^2GAN) (Wang and Gupta 2016) focuses on the basic principles by which indoor scene images have a 3D structure in the original world and have texture/style on their surfaces. S^2GAN first generates a 3D surface normal map and then a 2D RGB image, given the generated surface as the condition. In S^2GAN, it is important to consider fundamental information of target domains for generation. Our proposed model is based on these generative models (Goodfellow et al. 2014; Radford, Metz, and Chintala 2016; Wang and Gupta 2016; Isola et al. 2017).

Future frame prediction (Oh et al. 2015; Mathieu, Couprie, and LeCun 2016; Goroshin, Mathieu, and LeCun 2015; Srivastava, Mansimov, and Salakhutdinov 2015; Ranzato et al. 2014; Finn, Goodfellow, and Levine 2016; Lotter, Kreiman, and Cox 2017; Villegas et al. 2017a; 2017b; Xue et al. 2016) and future optical flow prediction (Walker, Gupta, and Hebert 2015; Walker et al. 2016) have shown impressive results. These studies are also related to our work in terms of video generation. However, although some methods (Mathieu, Couprie, and LeCun 2016; Villegas et al. 2017a; 2017b) exploits adversarial learning, our task completely differs from future frame prediction as follows. First, while future frame prediction gives the current frame or even the past frames as a condition, our proposed model generates a video only from Gaussian noise. Moreover, while future frame prediction has a definite ground truth as the generating target, our model discriminates a video based on whether it is real. Therefore, our task can be considered as completely different and a more difficult task than future frame prediction.

We make use of an action recognition knowledge in video generation. In action recognition, Two-stream (Simonyan and Zisserman 2014) has shown the importance of optical flow. Two-stream learns texture and motion information separately with two networks: RGB-stream and optical-flow-stream. Many later works on action recognition (Sun et al. 2015; Feichtenhofer, Pinz, and Zisserman 2016; Wang et al. 2016; Zolfaghari et al. 2017; Carreira and Zisserman 2017) employ this idea of splitting videos into orthogonal information. Thus, it turns out that optical flow is important information for action recognition. In this study, we show the importance of optical flow in video generation.

Video GAN (VGAN) (Vondrick, Pirsiavash, and Torralba 2016) has succeeded to generate scene-consistent videos by generating the foreground and background separately, which brings the generated videos scene consistency. The networks of VGAN consist of 3D convolutions aiming at learning motion information and appearance information simultaneously. However, since this method aims to capture motion and appearance only with single-stream 3D convolutional networks, the generated videos have problems with either visual appearance or motion. Some generated videos have plausible frames but no movement in the generated video. The other generated videos have movements, but their movements are implausible. Optical flow can solve these problems and provides richer motion information not only in action recognition but also in video generation. In action recognition, 3D convolutional networks only with RGB input show inferior performance to networks with RGB and optical flow inputs (Carreira and Zisserman 2017).

Temporal GAN (TGAN) (Saito and Matsumoto 2016) aims to simplify 3D convolutions in order to split appearance/motion information and train network parameters more efficiently. TGAN first applies 2D convolutions to RGB images several times, and then applies 1D temporal convolutions to activations resulting from the 2D convolutions. However, if consecutive frames are made to have a small resolution after applying several downsamplings, almost no spatial change could be seen in the time direction. Actually, in action recognition, TCL path of $\mathrm{F_{ST}CN}$ (Sun et al. 2015), which has an architecture similar to that of TGAN, does not show improved performance over the RGB stream of Two-stream, which is trained on RGB images frame by frame, on the action recognition dataset UCF101 (Soomro, Zamir, and Shah 2012). Therefore, when TGAN is trained on datasets that contain a variety of scenes, although each frame becomes a realistic still image, different appearance and scenes appear in the video, and the movement and consistency can not be assured.

To generate appearance-and-motion-realistic videos, we generate motion information and appearance information in two separate stages. We first generate optical flow to maintain the consistency of motion, and then provide texture information to the generated optical flow maintaining the consistency of appearance.

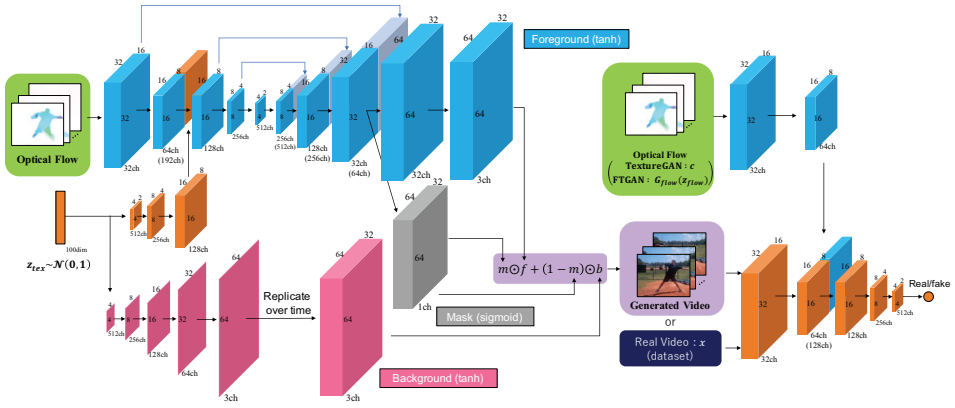

Figure 2: **Architecture of Texture GAN:** Given the optical flow video from datasets and z_{tex} as input, Texture Generator learns to generate RGB videos. For foreground generation, we apply 3D convolution to all layers and use skip connection in several layers as a U-net (Ronneberger, Fischer, and Brox 2015). For background generation, we apply 2D convolution to all layers. The size of input and output video is 64x64 resolution and 32 frames, which is a little over a second.

Preliminaries

Generative Adversarial Networks

Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) consist of two networks : Generator ($\boldsymbol{G}$) and Discriminator ($\boldsymbol{D}$). $\boldsymbol{G}$ attempts to generate data appearing similar to the given dataset. The input for $\boldsymbol{G}$ is a latent variable $\boldsymbol{z}$, which is randomly sampled from distribution $p_{\boldsymbol{z}}$ (e.g., a Gaussian distribution). Furthermore, $\boldsymbol{D}$ attempts to distinguish between real data and fake data generated from $\boldsymbol{G}$. A GAN simultaneously updates these two networks, and the objective function is given as follows :

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

Generative Adversarial Network for Video

Generative Adversarial Network for Video (VGAN) (Vondrick, Pirsiavash, and Torralba 2016) is a video generative network based on the concept of GANs. VGAN also consists of Generator and Discriminator. The generator of VGAN has a mask architecture to separately generate static background and moving foreground:

$$G(\boldsymbol{z}) = m(\boldsymbol{z}) \odot f(\boldsymbol{z}) + (1 - m(\boldsymbol{z})) \odot b(\boldsymbol{z}) \quad (2)$$

where $\odot$ represents the element-wise multiplication, $m(\boldsymbol{z})$ is a spatiotemporal matrix with each pixel value ranging from 0 to 1; it selects either the foreground $f(\boldsymbol{z})$ or the background $b(\boldsymbol{z})$ for each pixel (x, y, t). To generate a consistent background, $b(\boldsymbol{z})$ produces a spatial static image replicated over time. During learning, to encourage background image, L1 regularization $\lambda\|m(\boldsymbol{z})\|_1$ for $\lambda = 0.1$ is added on the mask to the GAN's original objective function.

Method

Video generation is a difficult task in terms of not only *generating natural frame images* but also *generating a consistent moving video*. Therefore, to obtain a video comprising plausible motion and realistic frames, we split the generative models into two networks: FlowGAN and TextureGAN. After training each network separately, we conduct joint learning. Figure 1 shows the overview of our FTGAN.

FlowGAN: Optical Flow Generation Model

The architecture of FlowGAN is based on VGAN (Vondrick, Pirsiavash, and Torralba 2016). However, considering that the background optical flow should be zero if the camera is fixed, our model does not comprise a background stream in the generator. Instead of learning the background generator, we give the zero matrix as b, which is equal to using only the foreground stream of VGAN.

$$G_{flow}(\boldsymbol{z}_{flow}) = m(\boldsymbol{z}_{flow}) \odot f(\boldsymbol{z}_{flow}) \quad (3)$$

TextureGAN: Optical Flow Conditional Video Generation Model

As one of the simplest ways to generate video using optical flow, we can warp the first image simply along the optical flow. However, the videos generated by such a method tend to be collapsed (Villegas et al. 2017b) especially on relatively long videos. It is more appropriate to fuse image and optical flow after feature-encoding them. Thus we take the following approach. As shown in Figure 2, our TextureGAN model considers the optical flow and z_{tex} as inputs and outputs a video. The architecture of our generator is based on VGAN and Pix2Pix (Isola et al. 2017). We employ the idea of foreground/background separation from VGAN. For foreground generation, we leverage optical flow which already comproses rough edges of the target video. Thus, we utilize the U-net architecture (Ronneberger, Fischer, and Brox 2015) as Pix2Pix.

Figure 3: Results of TextureGAN conditioned on ground truth optical flow that the network does not observe in the training on two datasets. (Left: Penn Action, Right: SURREAL). The first line in each result shows the input optical flow and the middle line shows the frames generated from the first line's optical flow. The last line shows the ground truth frames corresponding to input optical flow. In the third row of Penn Action, we can see that our model generates not only a person but also the bowling ball. Animated gifs of these results can be seen in the supplemental material. Note that in this figure, t represents the frame number in the videos.

Our Texture Generator G_{tex} is as follows :

$$G_{tex}\left(\boldsymbol{z}_{tex}, \boldsymbol{c}\right) = m\left(\boldsymbol{z}_{tex}, \boldsymbol{c}\right) \odot f\left(\boldsymbol{z}_{tex}, \boldsymbol{c}\right) + \left(1 - m\left(\boldsymbol{z}_{tex}, \boldsymbol{c}\right)\right) \odot b\left(\boldsymbol{z}_{tex}\right) \tag{4}$$

where, $\boldsymbol{c}$ is the ground truth optical flow. Note that we provide a *ground truth* optical flow to the discriminator as the condition for TextureGAN training.

Then, the loss functions for discriminator and the generator are as follows :

$$\begin{aligned} L_{D_{tex}}\left(\boldsymbol{x}, \boldsymbol{z}_{tex}, \boldsymbol{c}\right) &= \log\left(1 - D_{tex}\left(G_{tex}\left(\boldsymbol{z}_{tex}, \boldsymbol{c}\right), \boldsymbol{c}\right)\right) \\ &\quad + \log\left(D_{tex}\left(\boldsymbol{x}, \boldsymbol{c}\right)\right) \\ L_{G_{tex}}\left(\boldsymbol{z}_{tex}, \boldsymbol{c}\right) &= \log\left(D_{tex}\left(G_{tex}\left(\boldsymbol{z}_{tex}, \boldsymbol{c}\right)\right)\right) \end{aligned} \tag{5}$$

where, $\boldsymbol{x}$ is the ground truth video.

FTGAN: Both Optical Flow and Video Generation with Joint-Learning

Figure 1 shows system overview of FTGAN. We first train FlowGAN and TextureGAN independently, and then merge these networks through joint learning, during which the generator of the FlowGAN is updated depending on the loss propagated from not only its own discriminator but also the discriminator of TextureGAN. However, the loss from TextureGAN is a supplemental loss for FlowGAN; thus, we set the weighting parameter $\lambda = 0.1$ as S^2GAN (Wang and Gupta 2016). Through this joint learning, we consider that FlowGAN will become able to generate a complementary optical flow suitable for video generation. Note that we use the *generated* optical flow as a condition to the discriminator.

$$L^{joint}_{G_{flow}}(\boldsymbol{z}_{flow}, \boldsymbol{z}_{tex}) = L_{G_{flow}}(\boldsymbol{z}_{flow}) + \lambda \cdot L_{G_{tex}}(G_{flow}(\boldsymbol{z}_{flow}), \boldsymbol{z}_{tex}) \tag{6}$$

where $\boldsymbol{z}_{tex}$ and $\boldsymbol{z}_{flow}$ are independent variables.

Network Configuration

First, we train the FlowGAN and TextureGAN independently by using the Adam (Kingma and Ba 2014) optimizer with an initial learning rate $\alpha = 0.0002$ and momentum parameter $\beta_1 = 0.5$. The learning rate is decayed to $1/2$ from its previous value six times during the training. The latent variables $\boldsymbol{z}_{tex}$ and $\boldsymbol{z}_{flow}$ are Gaussian distributions with 100 dimensions. We set a batch size of 32. Batch normalization (Ioffe and Szegedy 2015) and Rectified Liner Unit (ReLU) activation are applied after every up-sampling convolution, except for the last layer. For down-sampling convolutions, we also apply batch normalization and LeakyReLU (Xu et al. 2015) to all layers but only apply batch normalization to the first layer. After training them independently, we join both networks and train our FTGAN full-network. Following S^2GAN, we set a small learning rate $\alpha = 1e-7$ for the FlowGAN and set $\alpha = 1e-6$ for the TextureGAN during the joint learning.

Experiments

Evaluation of generative model is difficult due to its lack of appropriate metrics (Theis, Oord, and Bethge 2015). Thus, we evaluate our method on generation and recognition tasks following previous works (Vondrick, Pirsiavash, and Torralba 2016; Saito and Matsumoto 2016). We first present our experiment of video generation. Next, we present our experiment on classifying actions to explore our model's capability to learn video representations in an unsupervised manner.

Dataset and Settings

For evaluating video generation, we conduct experiments on two video datasets of human actions. For the real world human video dataset, we use Penn Action (Zhang, Zhu, and Derpanis 2013), which has 2326 videos of 15 different classes and 163841 frames. We employ the original train/test split. For the Computer Graphics (CG) human video dataset, we use SURREAL (Varol et al. 2017), which is made by synthesizing CG humans and LSUN (Yu et al. 2015) images, and consists of 67582 videos. In the original train/test splits, even test videos have 12538 videos, which contain 1194662 frames. Thus, we use original test videos for training and 1659 subset videos from the original train for testing. The TexGAN and FlowGAN are trained in 60000 iterations respectively on each dataset. We conducted joint learning with 10000 iterations on each dataset. For optical flow computation, we use Epic flow (Revaud et al. 2015). We resize all frames and optical flow images to 76×76 resolution, and augment them by cropping them into 64×64 resolution images and randomly applying horizontal flips during training. Note that we remove a few videos in each dataset because they have less than 32 frames.

TextureGAN: Video Generation Results from Ground Truth Optical Flow and $\boldsymbol{z}_{tex}$

Figure 3 shows the results of our TextureGAN given the ground truth optical flow on Penn Action and SURREAL. TextureGAN generates videos with plausible motion. Also, on Penn Action, our method generates videos in which not only the motion of the humans appears but also the motion of the objects (e.g. the barbell in the second row , the bowling ball in the last row) appear. This is because the optical flow represents all moving objects as well as people, unlike keypoint information, (e.g., human joint positions), which only represent human motion (Villegas et al. 2017b). Moreover, the first row of Penn Action shows that our model generates the video in which a person is doing sit-up in a baseball ground; this combination of appearance and motion information does not exist in this dataset.

FTGAN: Video Generation Results from $\boldsymbol{z}_{flow}$ and $\boldsymbol{z}_{tex}$

In Figure 4, the results of our FTGAN are presented from only the latent variables $\boldsymbol{z}_{flow}$ and $\boldsymbol{z}_{tex}$. In this figure, although the results are less clear than the videos generated through the ground truth optical flow, we can observe plausible moves in the results. For example, in Penn Action, whereas VGAN does not generate videos with plausible motion, our TextureGAN generates a video in which a person is pitching a baseball. For another example, in SURREAL, whereas VGAN generates non-human like frames and disappearing movement, our methods generate human like frames and consistent motion.

Note that looking at the generated videos, it seems like the frames towards the middle of a video are generally better than frames at the beginning and the end. We believe this is caused by the zero-padding in up-sampling convolutions. In each convolutional layer, the beginning and the end frames are partially convolved with zero-padding features. This probably affects the quality of the borders of the generated videos. The spatial borders appear at the edge of each frame, thus they are not usually noticeable. However, the temporal borders appear at the beginning and the end frames (not only edge, but full frames), thus they are noticeable.

Quantitative Comparison to Previous Study: Human Evaluation

To evaluate our method quantitatively, it is desirable to allow humans to check the generated results. Thus, we use the Amazon Mechanical Turk (AMT) to compare the results. Table 1 shows A/B testing performances of the comparison between our methods and VGAN on AMT. We ask 160 unique workers to indicate which video looks more realistic on 50 videos, and obtained 8000 opinions.

Although TextureGAN shows slightly inferior performance to VGAN on SURREAL, our methods shows improved performance over VGAN. Especially our methods outperform VGAN on Penn Action, which contains more varied motion than SURREAL. We can infer that it is important to divide generation process into motion and appearance as the dataset increases the complexity of motion.

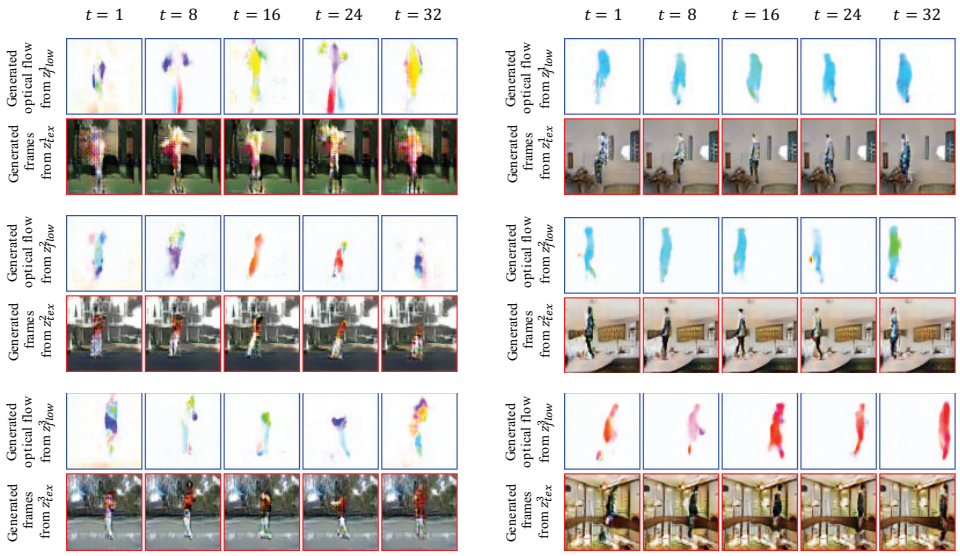

Figure 4: Results of FTGAN generated from z_{flow} and z_{tex}. (Left: Penn Action, Right: SURREAL). Unlike TextureGAN, our networks also generate optical flow images. We can observe that the generated optical flow contains plausible motion. For example, the first row of Penn Action looks like jumping jacks. Although the generated images are clearer when we use the ground truth optical flow than when we also generate optical flow, FTGAN generates the videos in which we can understand what action is being done. For example, in the second row of Penn Action, the person is pitching a ball. In the third row of Penn Action, the person is swinging a baseball bat. In the all results of SURREAL, we can confirm that the optical flow images have reasonable human shape and the generated frames also look realistic frames. Animated gifs of these results can be seen in the supplemental material.

Table 1: We compare our methods and VGAN with A/B testing. The table shows the percentage of that workers who prefer videos generated from each of our model instead of VGAN.

"Which human video looks more realistic ?"	SURREAL	Penn Action
Prefer TextureGAN (ours) over VGAN	44%	72%
Prefer FTGAN (ours) over VGAN	54%	58%

Qualitative Comparison to Previous Study: Visualized Results

We also compare the videos generated from our methods (TextureGAN, FTGAN) with those generated from VGAN, as shown in Figure 5. In this figure, we introduce examples of videos used for comparison in AMT. On the each dataset, the videos generated by VGAN do not show realistic moves, and the contours of humans are not maintained during the whole video. On the other hand, the videos generated by our methods have plausible motion, and humans are seen in the whole video consistently. We show several randomly sampled gif animations in the supplemental material.

Unsupervised Action Classification

As part of the comparison with other GAN methods, we conducted an experiment to investigate the unsupervised feature expression learning capability of FTGAN, as the same way with VGAN and TGAN. Although there are many other unsupervised feature extraction methods, we chose the recent GAN methods to compare the performance of our model since it is also GAN-based.

Table 2 shows the action classification performance on UCF101 (Soomro, Zamir, and Shah 2012). We train FTGAN on UCF101 with random initialized weights over three splits. As TGAN, we extract the activation of the last layer in the discriminator and learn a Linear SVM. To obtain video features throughout the video, we use a sliding window with overlapping 16 frames to extract features from an entire video as in C3D (Tran et al. 2015). For optical flow estimation, we employ the algorithm proposed by Brox et al. (Brox et al. 2004) following Two-stream (Simonyan and Zisserman 2014).

Table 2 shows that both of our discriminators outperform the discriminators of VGAN and TGAN. This improvements suggests that separating information ensures the capture of much richer video characteristics. Moreover, the fusion of both discriminators achieves significantly improved results, in the same way as Two-stream, a supervised method,

Figure 5: Qualitative comparison of our methods with VGAN. We also show several randomly sampled results of our methods and VGAN in the supplemental material.

achieves improved results by fusing the RGB-stream and the Flow-stream. This indicates that each discriminator learns complementary information. In other words, FlowGAN and TextureGAN are considered to learn mainly about motion and appearance information, respectively. The reason why TextureGAN does not comprise the motion information of FlowGAN although TextureGAN uses 3D convolution, could be that the discriminator of TextureGAN receives an optical flow as a condition. Thus, TextureGAN can focus on learning appearance information. Note that the results of "VGAN + Logistic Reg" and "VGAN + Fine Tune" have been achieved by 49.3% and 52.1% accuracy when pre-trained on 5000 hours video datasets; this is not commensurable with TGAN, "VGAN + Random Init", and our methods. However, the combination of both our discriminators still shows superior performance over them even without pre-training on the 5000 hours video dataset.

Conclusion and Future Work

In this study, we propose a FTGAN consisting of two GANs; FlowGAN and TextureGAN. Each network deals with complementary information: motion and appearance. Although the generated video has low resolution, is only a few seconds long, our model is able to generate videos successfully with more plausible motion than the other methods (i.e., VGAN). In addition, we have succeeded in capturing significantly improved unsupervised video representations, confirming that each discriminator learns complementary information. Through this paper, we argue that it is important not only for video recognition but also for video generation to focus on a basic underlying principle of video.

We believe finding other architectures specific for optical flow is a very important future work. It would also be interesting to investigate whether the length of synthesized video plays a factor in the comparison of previous works. The separate treatment of optical flow and texture can potentially reduce the generation space and is expecting better results on longer videos.

Acknowledgement

The authors would like to thank Antonio Tejero-de-Pablos, Hiroharu Kato, and Takuhiro Kaneko for providing useful discussions. We also thank Yusuke Mukuta for helping with the optical flow extraction. This work was partially supported by the Ministry of Education, Culture, Sports, Science and Technology (MEXT) as "Seminal Issue on Post-K Computer".

References

Brox, T.; Bruhn, A.; Papenberg, N.; and Weickert, J. 2004. High accuracy optical flow estimation based on a theory for warping. In *ECCV*.

Carreira, J., and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*.

Feichtenhofer, C.; Pinz, A.; and Zisserman, A. 2016. Convolutional two-stream network fusion for video action recognition. In *CVPR*.

Finn, C.; Goodfellow, I.; and Levine, S. 2016. Unsupervised learning for physical interaction through video prediction. In *NIPS*.

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

Goroshin, R.; Mathieu, M. F.; and LeCun, Y. 2015. Learning to linearize under uncertainty. In *NIPS*.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv:1502.03167*.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *CVPR*.

Table 2: Accuracy of unsupervised action classification on UCF101. We train our models with randomly initialized weights. Note that while the first group from the top shows the methods trained with randomly initialized weights, the second group from the top shows the methods pre-trained on 5000 hours video datasets. In addition, while Two-stream (Simonyan and Zisserman 2014) is trained in a supervised manner, the other methods, including our proposed methods, are trained in an unsupervised manner.

Method	Accuracy
Chance	0.9%
VGAN + Random Init (Vondrick, Pirsiavash, and Torralba 2016)	36.7%
TGAN: Image-discriminator + Linear SVM (Saito and Matsumoto 2016)	38.6%
TGAN: Temporal-discriminator + Linear SVM (Saito and Matsumoto 2016)	23.3%
FTGAN **(ours)**: Flow-discriminator + Linear SVM	**49.5%**
FTGAN **(ours)**: Texture-discriminator + Linear SVM	**50.5%**
FTGAN **(ours)**: Flow-discriminator & Video-discriminator (fusion by Linear SVM)	**60.9%**
VGAN + Logistic Reg (Vondrick, Pirsiavash, and Torralba 2016)	49.3%
VGAN + Fine Tune (Vondrick, Pirsiavash, and Torralba 2016)	52.1%
RGB-stream in Two-stream (Simonyan and Zisserman 2014)	73.0%
Flow-stream in Two-stream (Simonyan and Zisserman 2014)	83.7%
Two-stream (fusion by Linear SVM) (Simonyan and Zisserman 2014)	88.0%

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980*.

Lotter, W.; Kreiman, G.; and Cox, D. 2017. Deep predictive coding networks for video prediction and unsupervised learning. In *ICLR*.

Mathieu, M.; Couprie, C.; and LeCun, Y. 2016. Deep multi-scale video prediction beyond mean square error. In *ICLR*.

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

Radford, A.; Metz, L.; and Chintala, S. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*.

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

Revaud, J.; Weinzaepfel, P.; Harchaoui, Z.; and Schmid, C. 2015. EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In *CVPR*.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*.

Saito, M., and Matsumoto, E. 2016. Temporal generative adversarial nets. *arXiv:1611.06624v1*.

Simonyan, K., and Zisserman, A. 2014. Two-stream convolutional networks for action recognition in videos. In *NIPS*.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402*.

Srivastava, N.; Mansimov, E.; and Salakhutdinov, R. 2015. Unsupervised learning of video representations using lstms. In *ICML*.

Sun, L.; Jia, K.; Yeung, D.-Y.; and Shi, B. E. 2015. Human action recognition using factorized spatio-temporal convolutional networks. In *ICCV*.

Theis, L.; Oord, A. v. d.; and Bethge, M. 2015. A note on the evaluation of generative models. *arXiv:1511.01844*.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*.

Varol, G.; Romero, J.; Martin, X.; Mahmood, N.; Black, M. J.; Laptev, I.; and Schmid, C. 2017. Learning from synthetic humans. In *CVPR*.

Villegas, R.; Yang, J.; Hong, S.; Lin, X.; and Lee, H. 2017a. Decomposing motion and content for natural video sequence prediction. In *ICLR*.

Villegas, R.; Yang, J.; Zou, Y.; Sohn, S.; Lin, X.; and Lee, H. 2017b. Learning to generate long-term future via hierarchical prediction. In *ICML*.

Vondrick, C.; Pirsiavash, H.; and Torralba, A. 2016. Generating videos with scene dynamics. In *NIPS*.

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

Walker, J.; Gupta, A.; and Hebert, M. 2015. Dense optical flow prediction from a static image. In *ICCV*.

Wang, X., and Gupta, A. 2016. Generative image modeling using style and structure adversarial networks. In *ECCV*.

Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Van Gool, L. 2016. Temporal segment networks: towards good practices for deep action recognition. In *ECCV*.

Xu, B.; Wang, N.; Chen, T.; and Li, M. 2015. Empirical evaluation of rectified activations in convolutional network. *arXiv:1505.00853*.

Xue, T.; Wu, J.; Bouman, K.; and Freeman, B. 2016. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NIPS*.

Yu, F.; Seff, A.; Zhang, Y.; Song, S.; Funkhouser, T.; and Xiao, J. 2015. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv:1506.03365*.

Zhang, W.; Zhu, M.; and Derpanis, K. G. 2013. From actemes to action: A strongly-supervised representation for detailed action understanding. In *ICCV*.

Zolfaghari, M.; Oliveira, G. L.; Sedaghat, N.; and Brox, T. 2017. Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection. In *ICCV*.