Mutual Suppression Network for Video Prediction using Disentangled Features

Jungbeom Lee[1]
jbeom.lee93@snu.ac.kr
Jangho Lee[1]
ubuntu@snu.ac.kr
Sungmin Lee[1]
simon0810@snu.ac.kr
Sungroh Yoon *,[1,2]
sryoon@snu.ac.kr

[1] Department of Electrical and
Computer Engineering
Seoul National University

[2] ASRI, INMC, ISRC, and Institute
of Engineering Research,
Seoul National University

* Corresponding Author

Abstract

Video prediction has been considered a difficult problem because the video contains not only high-dimensional spatial information but also complex temporal information. Video prediction can be performed by finding features in recent frames, and using them to generate approximations to upcoming frames. We approach this problem by disentangling spatial and temporal features in videos. We introduce a mutual suppression network (MSnet) which are trained in an adversarial manner and then produces spatial features which are free of motion information, and motion features with no spatial information. MSnet then uses motion-guided connection within an encoder-decoder-based architecture to transform spatial features from a previous frame to the time of an upcoming frame. We show how MSnet can be used for video prediction using disentangled representations. We also carry out experiments to assess the effectiveness of our method to disentangle features. MSnet obtains better results than other recent video prediction methods even though it has simpler encoders.

1 Introduction

Given a sequence of frames from a video, the process of video prediction attempts to generate one or more upcoming frames. Video prediction is important in real-time systems such as robots, closed-circuit television (CCTV), and self-driving cars, and also has a place in applications such as the unsupervised learning of image representations from videos [18].

The learning of representations from images has been studied extensively, and the results now surpass human ability [4]. However, learning representations from videos remains a challenging task because of the temporal dimension, which brings a huge number of variations, and because it is not possible to annotate every frame in a video with labels. Some 'natural' labeling of videos is possible, for instance based on temporal coherence. However, the entangling of content and motion information in videos tends to make unsupervised learning challenging. In this regard, there have been previous works on decomposing videos into content and motion components [3, 17, 19, 20, 22, 24]. While the learning techniques used on images [5, 6] can be extended for the content representations [11, 23], the learning of representations of motion has not been studied so extensively. Temporal information can be obtained from optical flow [12], with reasonable results. However, optical flow estimation involves a great deal of computation and depends on having a labeled dataset, which requires tremendous effort and cost to obtain.

We propose a technique in which a mutual suppression network (MSnet) is used to disentangle motion and content features. This approach is based on the following intuitive assertions:

Separability of features: We train MSnet in such a way that information of one type is suppressed during the extraction of features of another type. This can be achieved by mutual adversarial learning.

Content from several frames: The majority of methods that encode video into motion and content obtain content features from a single frame. We argue that content features, as well as motion features, should be obtained from several frames. A single frame is not sufficient to capture content information if two objects are occluded or cannot be distinguished.

Reproducibility: Given three frames $x^{(1)}$, $x^{(2)}$, and $x^{(3)}$, the content features from $x^{(1)}$ and $x^{(2)}$, together with the motion features from $x^{(2)}$ and $x^{(3)}$, should allow us to reproduce $x^{(3)}$. This leads to motion and content features which contain semantic information.

Time-reversibility of content: While previous methods have been based on the assumption that content features are mostly time-invariant, we propose that the content features should be time-reversible, so a content feature obtained from (x_1, x_2) should be the same as that obtained from (x_2, x_1). This time-reversible property is intended to ensure that motion information is not unwittingly included in content features because we extract content features from two frames, which may be related by temporal information.

The second step is the frame prediction task using the encoders and a generator trained in the first step. To generate frames from the features from the encoders, previous methods utilize the skip connections as used in UNet [14], that transfers information direct from a previous frame to a target frame. During frame prediction, however, it is better that the generator takes information related to the target frame, not the previous frame. Therefore we introduce a motion-guided connection which modifies the information from the previous frame to become the information needed for generating the target frame by considering the motion features.

2 Related Work

There is no easy way of representing spatial and temporal information simultaneously in videos. Recent work in video representation learning has therefore focused on disentangling temporal and spatial information in natural videos. Simonyan and Zisserman used a two-stream network for action recognition in videos, motivated by the way in which the human visual cortex decouples complementary information appearing in videos [17], which has subsequently been used for various fields of video processing [1, 8, 26].

The prediction of video frames requires the ability both to understand previous frames and to produce realistic new frames. These tasks can be facilitated by decomposing a video into motion and content components using techniques based on a two-stream network. VGAN [21] predicted upcoming frames by modeling the foreground separately from the background. MCnet [20] used an encoder-decoder technique to separate the motion and content information of a video: a content encoder extracts spatial features from the most recent frame of a video, and a motion encoder captures motion dynamics from pixel-wise differences between previous pairs of frames. However, few of these differences contain any semantic information about motions. DRnet [2] used a content discriminator to separate the pose attributes from the content attributes in a frame. The content discriminator examines whether two pose features relate to the same content or not. The pose features acquired in this way are used to predict future pose features, from which upcoming frames can be generated. However, DRnet cannot catch pure content information because it only uses one-way suppression. For example, when predicting the frames using videos in the KTH dataset [15], DRnet sometimes changed the identities of human in the predicted frames. In addition, poses tend to be more ambiguous than motions in videos, so DRnet sometimes swapped the locations of two numerals when applied to the Moving MNIST dataset [18]. DRnet is only concerned with a series of absolute locations (poses), and not with relative locations (motion).

3 Proposed Method

The proposed method consists of two steps. The first step is frame reproduction, which obtains disentangled features from a consideration of semantics in the frames. The second step is video prediction using the disentangled features obtained during frame reproduction.

3.1 Frame Reproduction

Let x_t denote the t^{th} frame in video x. The frame reproduction is to reproduce x_{t+k} from the known frames x_t, x_{t+1}, and x_{t+k}. Following the ‘Reproducibility’ assertion presented in the Introduction, we reproduce x_{t+k} from the content of x_t, x_{t+1} and the motion of x_{t+1}, x_{t+k}, with the aim of obtaining disentangled motion and content features by considering semantics. We describe our network architecture in Section 3.1.1 and our training procedure in Section 3.1.2.

3.1.1 Network Architecture

The structure of MSnet is presented in Figure 1. One encoder extracts content features and another extracts motion features. From these features, a generator reproduces x_{t+k}. Specifically, The content encoder E_c obtains the content information $E_c(x_t, x_{t+1})$ from two successive frames x_t and x_{t+1}. The motion encoder E_m extracts motion information $E_m(x_{t+1}, x_{t+k})$ from frames x_{t+1} and x_{t+k}, which do not have to be adjacent. The generator G reproduces the last frame of the input x_{t+k}. The motivation for these settings is shown in the appendix.

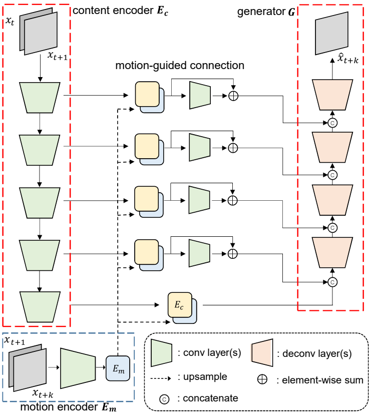


Figure 1: Architecture of MSnet, showing multi-scale motion-guided connections.

Motion-guided Connection: The generator G is connected to the content encoder E_c by block-wise motion-guided connections, which play a similar role to the skip connections in UNet [14], but each motion-guided connection performs an additional convolution operation guided by motion feature. This reduces ghosting in the reproduced frame: a standard skip connection tends to preserve information about previous frames x_t and x_{t+1} (but not x_{t+k}), which causes a ghost of x_t and x_{t+1} to remain in the reproduced x_{t+k}. We concatenate the features of each convolutional block and a bi-linearly upscaled motion feature $E_m(x_{t+1}, x_{t+k})$, and then pass the concatenated features into a 1×1 convolutional layer to adjust the number of channels, and add residual connections. These motion-guided connections use motion information to modify the spatial information from previous frames, so that it can be effectively transferred to the target frames.

Discriminators for adversarial training: We apply adversarial learning to train MSnet, with three discriminators. For realistic and sharp results, we use a frame discriminator. We use two additional discriminators to disentangle motion and content features. More details of these are given in Section 3.1.2.

3.1.2 Training Procedure

Based on the intuitive assertions presented in the Introduction, we define the following objective terms: To train the encoders, we use

$$L_1 = L_{\text{rec}} + \alpha L_{\text{rev}} + \beta(L_{\text{advC}} + L_{\text{advM}} + L_{\text{advF}}), \tag{1}$$

where α and β are hyperparameters. To train the discriminators, we use

$$L_2 = L_{\text{DF}} + L_{\text{DC}} + L_{\text{DM}}. \tag{2}$$

We optimize L_1 and L_2 alternately. The loss terms in L_1 and L_2 are described below. In what follows, $\hat{x}_{t+k}$ denotes the reproduced frame $G(E_c(x_t, x_{t+1}), E_m(x_{t+1}, x_{t+k}))$.

Reconstruction and time-reversal Losses: Based on the 'Reproducibility' and 'Time-reversibility' assertions presented in the Introduction, we define L_{rec} and L_{rev} as follows:

$$L_{\text{rec}} = \|\hat{x}_{t+k} - x_{t+k}\|_2^2, \tag{3}$$

$$L_{\text{rev}} = \|E_c(x_t, x_{t+1}) - E_c(x_{t+1}, x_t)\|_2^2, \tag{4}$$

where k represents the temporal distance between the target frame and the reference frame.

Frame adversarial loss: DRnet [2] uses mean squared error loss alone, which tends to produce blurry results in image reproduction [10]. We thus introduce an extra frame adversarial loss, using a technique similar to that employed in the pix2pix network [7]. The frame discriminator D_f is trained to determine whether its input is a real pair of frames or not, and D_f is trained by L_{DF} which is expressed as follows:

$$L_{\text{DF}} = -\log D_f(x_t, x_{t+k}) - \log(1 - D_f(x_t, \hat{x}_{t+k})) \tag{5}$$

The adversarial loss L_{advF} expresses the extent to which synthetic frames produced by the generator G manage to deceive the discriminator. The generator G is trained by L_{advF} to synthesize realistic frames with the aim of deceiving the frame discriminator, and L_{advF} is expressed as follows:

$$L_{\text{advF}} = -\log D_f(x_t, \hat{x}_{t+k}). \tag{6}$$

Disentangling adversarial loss: The notion of 'Separability of features' described in the Introduction is realized by the content discriminator D_c and motion discriminator D_m. The content discriminator is trained to determine whether two motion features come from the same video, which requires it to discover the content information in these features. Thus, to deceive the content discriminator, the motion encoder must generate motion features that contain as little content information as possible. We train the content discriminator to discover content information in motion features using the loss L_{DC}, and the loss L_{advC} is used to train the motion encoder in such a way that the motion discriminator cannot make a

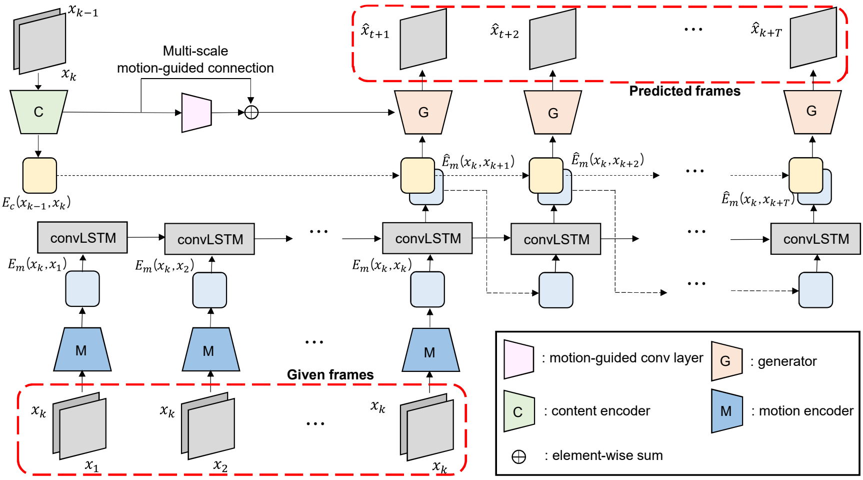


Figure 2: The frame prediction network. Given k frames, the network predicts upcoming T frames.

decision, which means that the entropy becomes maximized. Note that Eq 8 can be simplified to $L_{\text{advC}} = -\log x - \log(1-x)$ if we set $x = Dc()$, and the function has the minimum value when $x = 1/2$. The result is that the motion encoder obtains a pure motion feature. These two losses are formulated as follows:

$$L_{\text{DC}} = -\log D_c(E_m(x_a, x_{a+1}), E_m(x_b, x_{b+1})) - \log(1 - D_c(E_m(x_a, x_{a+1}), E_m(y_b, y_{b+1}))) \quad (7)$$

$$L_{\text{advC}} = -\log D_c(E_m(x_a, x_{a+1}), E_m(x_b, x_{b+1})) - \log(1 - D_c(E_m(x_a, x_{a+1}), E_m(x_b, x_{b+1}))) \quad (8)$$

where a and b are different frame numbers, and x and y are different videos.

In a similar way, the motion discriminator is trained to determine whether two content features are from sequential or non-sequential frames, which requires it to discover the motion information from the content feature. The content encoder can deceive the motion discriminator if it generates content features that do not contain motion information. We train the motion discriminator to discover motion information in the content feature by the loss L_{DM}, and the content encoder is trained to deceive the motion discriminator by the loss L_{advM}; so that the content encoder can obtain a pure content feature. These two losses are formulated as follows:

$$L_{\text{DM}} = -\log D_m(E_c(x_a, x_{a+1})) - \log(1 - D_m(E_c(x_a, x_b))), \quad (9)$$

$$L_{\text{advM}} = -\log D_m(E_c(x_a, x_{a+1})) - \log(1 - D_m(E_c(x_a, x_{a+1}))), \quad (10)$$

where x_a and x_{a+1} are sequential frames, and x_a and x_b are non-sequential frames.

3.2 Video Frame Prediction

We apply the motion and content encoders trained during frame reproduction to video prediction. MSnet is given k frames $(x_1, \cdots, x_k)$ and trained to predict the following T frames $(x_{k+1}, \cdots, x_{k+T})$, using the network illustrated in Figure 2. The motion encoder extracts motion features from the pairs $(x_k, x_1), (x_k, x_2), \cdots, (x_k, x_k)$, and the content encoder extracts content features from (x_{k-1}, x_k). Note that the first frame in each pair is always x_k during motion extraction. A convolutional LSTM network

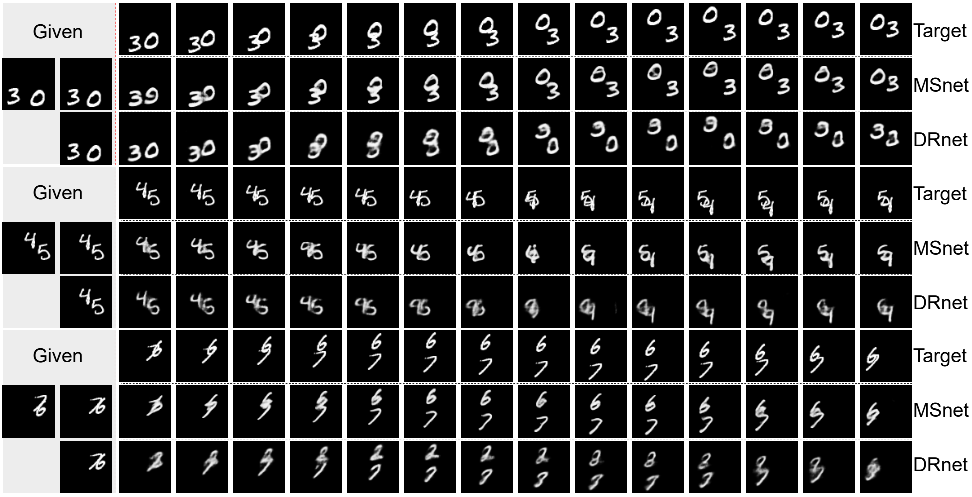


Figure 3: Qualitative results of frame reproduction task on the Moving MNIST dataset.

(cLSTM) [25] takes the motion features $E_m(x_k, x_t)(1 \leq t \leq k)$ extracted from each given pairs of frames and predicts the motion features of the subsequent frames $\hat{E}_m(x_k, x_{t+1})$ until the k^{th} frame.

$$\text{cLSTM}(E_m(x_k, x_t)) = \hat{E}_m(x_k, x_{t+1}) \; (1 \leq t \leq k). \tag{11}$$

For subsequent unknown frames, the predicted motion features are fed back into the cLSTM and the motion features of the next upcoming frames are predicted. By repeating this step, we can predict the motion features of the following T frames.

$$\text{cLSTM}(\hat{E}_m(x_k, x_t)) = \hat{E}_m(x_k, x_{t+1}) \; (k < t < T) \tag{12}$$

The cLSTM is trained using the following objective function:

$$L_{\text{lstm}} = \|\text{cLSTM}(E_m(x_k, x_k)) - E_m(x_k, x_{k+1})\|^2 + \sum_{t=k+1}^{T-1} \|\text{cLSTM}(\hat{E}_m(x_k, x_t)) - E_m(x_k, x_{t+1})\|^2 \tag{13}$$

Finally, the generator produces $\hat{x}_t$ from the t^{th} $(t > k)$ predicted motion features $\hat{E}_m(x_k, x_t)$, together with the content features $E_c(x_{k-1}, x_k)$. By repeating this step, we can generate the required number of upcoming frames.

4 Experiments

We performed experiments using the Moving MNIST and KTH datasets [15, 18]. First, we performed frame reproduction using the Moving MNIST to compare MSnet with DRnet [2]. Then, we present frame reproduction, frame prediction and disentangling experiments (feature-based nearest retrieval and t-SNE visualization) on the KTH dataset.

4.1 Moving MNIST

The Moving MNIST dataset [18] contains 10,000 video sequences, each consisting of 20 frames. In each video sequence, two digits move independently around the frame, which has a spatial rsolution of 64 × 64 pixels. The digits frequently intersect with each other and bounce off the edges of the frame. We used 8,000 sequences for training and 2,000 for testing. We used motion features with a 4 × 4 spatial map and 4 channels, and content features with a 4 × 4 spatial map and 8 channels. We use more channels for the content features because the motions occurring in the Moving MNIST videos are not

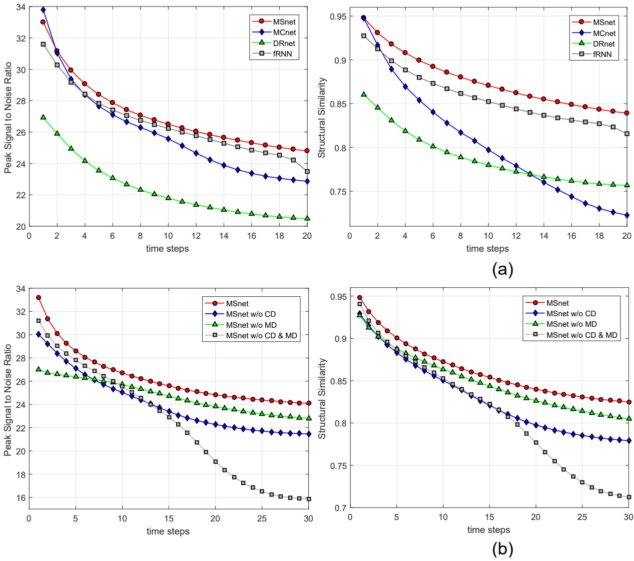


Table 1. The average SSIM and PSNR over predicted 20 frames on KTH dataset. # params denotes the number of parameters in motion and content encoders.

	SSIM	PSNR	# params
ConvLSTM [25]	0.712	23.58	-
TrajGRU [16]	0.790	26.97	-
DRnet [2]	0.788	22.32	23.3M
MCnet [20]	0.804	25.94	3.5M
fRNN [13]	0.857	26.53	-
MSnet (Ours)	**0.876**	**27.08**	3.2M

Table 2. The average SSIM and PSNR over predicted 20 and 30 frames on KTH dataset

	20 frames		30 frames	
Method	SSIM	PSNR	SSIM	PSNR
MSnet	**0.876**	**27.08**	**0.862**	**26.28**
MSnet w/o MGC*	0.868	26.31	0.854	25.39
MSnet w/o CD	0.850	25.08	0.829	24.06
MSnet w/o MD	0.863	25.29	0.848	24.71
MSnet w/o both	0.851	24.94	0.811	22.29

* MGC denotes motion-guided connection

Figure 4: Quantitative results of the frame prediction on the KTH dataset. (a) Comparison with state-of-the-art methods. (b) Comparison with ablation settings.

as complicated as those in natural videos. We used values of the temporal distance k between 0 and 5 in the frame reproduction process, and we set $\alpha = 1.0$ and $\beta = 3.3 \times 10^{-5}$ in Eq. (1).

Qualitative results from this experiment are shown in Figure 3. Note that Denton and Birodkar (2017) used self-generated colored digits to train DRnet, thus we re-trained it with the publicly available Moving MNIST data to make a fair comparison with MSnet. MSnet obtains content features from the given frames and motion features from the last given and target frames. DRnet obtains content features from the given frames and pose features from the target frames.

In the first example of Figure 3, DRnet generates digits in the wrong places. We attribute this to the way in which DRnet encodes temporal attributes into pose features, and not into motion features, like those used by MSnet. This suggests that motion is a more natural attribute of video than pose. In the second example, DRnet produces blurry results where the two digits overlap in target frames. In the third example, DRnet cannot identify two digits which overlap in a given frame. MSnet can identify these overlapping digits correctly because it obtains content features from two frames. More results with Moving MNIST are presented in the supplementary material.

4.2 KTH Dataset

The KTH dataset [15] contains videos of 25 people performing six actions. For our experiments, we resized the frames in the videos to 128 × 128 pixels. We used person 1-16 for training and person 17-25 for testing, following the widely used baseline method MCnet [20]. We used SSIM, PSNR, and inception score as evaluation metrics. We used motion features with a 8×8 spatial map and 8 channels and content features with a 8×8 spatial map and 8 channels. We used values of the temporal distance k between 0 and 10 in the frame reproduction process. We set $\alpha = 1.0$ and $\beta = 4 \times 10^{-5}$ in Eq. (1). For the following figures, we denote the motion and content discriminators as MD and CD, respectively.

4.2.1 Video frame prediction

We used the same experimental settings used in MCnet for frame prediction experiments. All baseline networks were trained by taking 10 frames from the KTH dataset and using them to predict the following 10 frames. During testing, 3,559 sequences of 30 frames were used: 10 given frames and 20

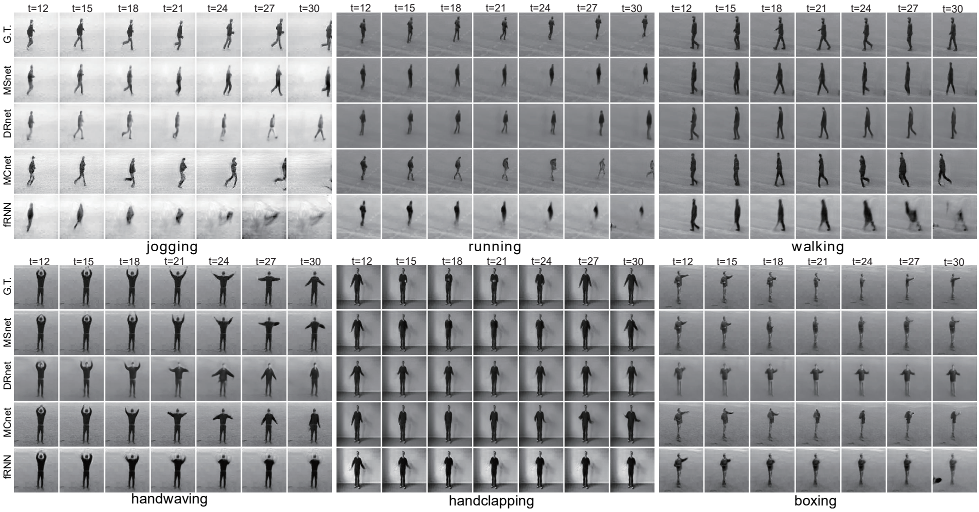


Figure 5: Qualitative results of frame prediction on the KTH dataset. Given 10 frames, the following 20 frames are predicted. We show every 3 frames.

frames to be predicted. The published DRnet model was trained on person 1-20, so we re-trained DRnet on person 1-16 for fair comparison with other baseline methods. Quantitative results are presented in Figure 4(a) and (b).

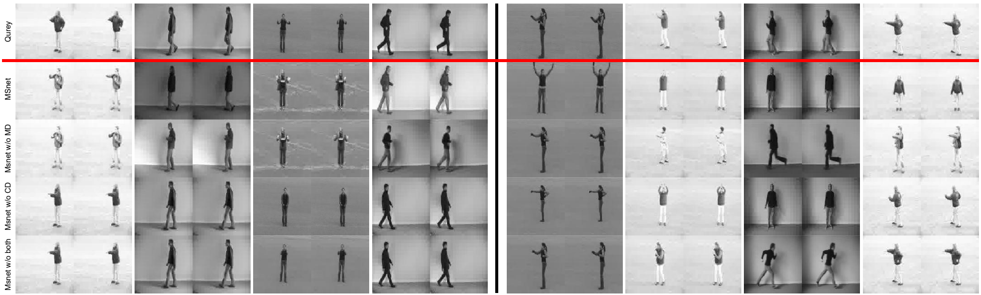
Figure 4(a) shows that MSnet obtained better results than other state-of-the-art methods on three evaluation metrics even though it has simpler motion and content encoders. Note that we do not report the number of parameters for ConvLSTM [25], TrajGRU [16], and fRNN [13], because it does not decompose videos into motion and content streams. Qualitative results are shown in Figure 5. DRnet produces the wrong motion in the boxing video, and changes the identity of the person in the handwaving video. We attribute these problems to DRnet's use of a basic UNet and one-directional suppression. MCnet produces a person with an unrealistic shape, which we attribute to its poor disentangling of features and a lack of semantic information in its motion features. fRNN has difficulty when the person in the frame makes a large motion, and we attribute this to the way in which it considers motion and content information simultaneously. These results suggest that meaningful features are obtained by mutual suppression and motion-guided connection. More results are presented in the supplementary material.

In ablation experiments, we removed each discriminator in turn to show the effects of mutual suppression on the disentangled features. Figure 4(b) shows that the results from MSnet are worse when either the motion or content discriminator is removed. Without the content discriminator (blue and gray lines), prediction performance drops significantly across subsequent frames because the motion encoder generates impoverished motion feature. With only the content discriminator (green line), the content encoder cannot extract meaningful content features, so it performs poorly on the first predicted frame. However, its performance does not drop significantly across subsequent frames as meaningful motion features can be extracted with the content discriminator. These results demonstrate how mutual suppression disentangles motion and content features.

4.2.2 Disentangling Experiments

We present t-SNE visualization [9] and feature-based nearest retrieval results to show the disentangling effect.

Feature-based frame retrieval: The aim of feature-based frame retrieval is to fetch the frame



(a) Nearest motion retrieval (b) Nearest content retrieval

Figure 6: Feature-based nearest frame retrieval results.

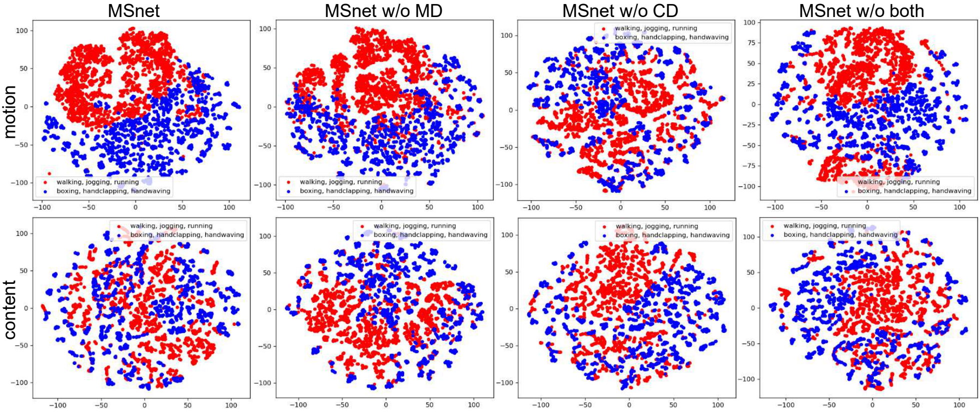


Figure 7: Visualization by t-SNE dimensional reduction of dynamic actions (walking, jogging, and running) (the red points), and static actions (boxing, handclapping, and handwaving) (the blue points), with and without the content and motion discriminators. The first row shows the distribution of motion features, and the second row shows the distribution of content features.

which is closest to a query frame in terms of motion and content features. More formally,

$$x_{\mathrm{r}} = \operatorname*{argmin}_{x} \|E(x_q) - E(x)\|, \qquad E \in \{E_c, E_m\}, \tag{14}$$

where x_{q} is a query frame, x is a frame other than the query frame, and x_{r} is the retrieved frame.

Figure 6 shows the results of nearest motion and content retrieval. Ideally, nearest motion retrieval should retrieve the most similar motion regardless of the content (the identity of the person and the background), and content retrieval should retrieve the most similar content regardless of any motion. As shown in Figure 6(a), MSnets with the content discriminator (the second and third rows) retrieve the most similar motions regardless of the identity the person and the background, because the content discriminator helps the motion encoder to extract pure motion features. In Figure 6(b), MSnets with the motion discriminator (the second and fourth rows) retrieve the most similar content regardless of the motions, because the motion discriminator helps the content encoder to extract pure content features.

t-SNE visualization: As MSnet is trained in an unsupervised manner, it cannot separate similar actions, such as running and jogging. For visualization by t-SNE dimensional reduction, we thus classify walking, jogging, and running as dynamic actions, and boxing, handclapping, and handwaving as static actions. If motion and content are disentangled as intended, motion features in the same actions should be clustered, and motion features in different actions should be separated. Conversely, points corresponding to content features which do not contain motion information should not be clustered.

The results of t-SNE visualization are shown in Figure 7. MSnet with both discriminators produces the most clustered motion features, and the most random distribution of content features. Using the content discriminator alone, motion features are reasonably well clustered, as pure motion features can still be captured effectively. However, content features are now more clustered because the omission of the motion discriminator means that content features contain unwanted temporal information. Without the content discriminator, and without both discriminators, the results are far from what we intend.

5 Conclusions

We have proposed a new method for video frame prediction. We have introduced mutual suppression adversarial training to acquire disentangled motion and content representations, and applied motion-guided connection to refine the content information from previous frames for use in the prediction of upcoming frames. MSnet has been shown to obtain well-disentangled features. This lead to better results in terms of frame prediction than other state-of-the-art methods. We have also shed light on the way in which mutual suppression disentangles features by ablation studies in the domains of t-SNE visualization and feature-based nearest frame retrieval.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science and ICT) [2018R1A2B3001628], Samsung Electronics (DS and Foundary), and the Brain Korea 21 Plus Project in 2019.

References

[1] Dahjung Chung, Khalid Tahboub, and Edward J Delp. A two stream siamese convolutional neural network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1983–1991, 2017.

[2] Emily L Denton and Vighnesh Birodkar. Unsupervised learning of disentangled representations from video. In *Advances in Neural Information Processing Systems*, pages 4417–4426, 2017.

[3] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In *Advances in Neural Information Processing Systems*, pages 3468–3476, 2016.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[6] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, page 3, 2017.

[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[8] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. *arXiv preprint arXiv:1701.05384*, 2(3):6, 2017.

[9] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[10] Michael Mathieu, Camille Couprie, Yann LeCun, and s s. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[11] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544, 2016.

[12] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4694–4702, 2015.

[13] Marc Oliu, Javier Selva, and Sergio Escalera. Folded recurrent neural networks for future video prediction. *European Conference on Computer Vision*, 2018.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[15] Christian Schuldt, Ivan Laptev, Barbara Caputo, and some. Recognizing human actions: a local svm approach. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 3, pages 32–36, 2004.

[16] Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. In *Advances in Neural Information Processing Systems*, pages 5617–5627, 2017.

[17] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[18] Nitish Srivastava, Elman Mansimov, Ruslan Salakhudinov, et al. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852, 2015.

[19] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. *arXiv preprint arXiv:1707.04993*, 2017.

[20] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. In *Proceedings of the International Conference on Learning Representations*, 2017.

[21] Carl Vondrick, Hamed Pirsiavash, Antonio Torralba, and s s. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016.

[22] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016.

[23] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. *arXiv preprint arXiv:1505.00687*, 2015.

[24] Xiaolong Wang, Ali Farhadi, and Abhinav Gupta. Actions˜ transformations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2658–2667, 2016.

[25] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, pages 802–810, 2015.

[26] Zhenheng Yang, Jiyang Gao, and Ram Nevatia. Spatio-temporal action detection with cascade proposal and location anticipation. *arXiv preprint arXiv:1708.00042*, 2017.