Motion-Object-Background Disentangled Representation from Video

Binghao Deng(binghaod) [1] **Weizhao Shao (weizhaos)** [1] **Siyu Gao (siyug)** [1] **Yue Li (yuel5)** [1]

Abstract

We explore the idea of disentanglement learning in video prediction task. Based on existing *content-pose* disentanglement learning frameworks, we further decompose *content* feature into *background* and *object* part, and design an adversarial training network structure for next frame video prediction. To evaluate our method, we create a synthetic video dataset, along with a mask extraction network to compare different video prediction methods in pose and content prediction quality; then we compare several existing methods and demonstrate the improvement in performance after proposed disentanglement.

1. Introduction

Unsupervised learning from videos is a long-standing problem in computer vision and machine learning. The goal is to learn a representation that generalizes effectively to a range of tasks, such as semantic classification, predicting future frames of the video, classifying the dynamic activity taking place, or generating new videos from existing ones. There are several prevailing paradigms, and the most general approaches are predictive auto-encoders, with different kinds of constraints on the latent representation space (Denton et al., 2017).

Visual scenes consist of separate objects, which may have different poses and identities within their category; in natural languages, the syntax and semantics of a sentence can often be separated from one another, and the idea of disentangled learning arises naturally from these observations. Generally, disentangled methods attempt to separate the latent representation to two parts, *content* and *motion*, to capture features about objects and motions, respectively (Tulyakov et al., 2018).

Various attempts have been made in applying disentangled idea to deep generative models for videos, some achieving impressive performance in predicting future frames. But there are several drawbacks: first, the connections between disentangled and interpretability are not clear. Second, many methods lack quantitative evaluation of disentangled quality like motion and content. Third, disentangled parts are easy to blur during the prediction or generation procedure. Besides, in terms of application, experiments of all existing frameworks are done on synthetic datasets or real-world videos with simple backgrounds and motions. It remains unknown whether those methods can perform well in more complex settings. Based on the experiments with DRNET (Denton et al., 2017) and MoCoGAN (Tulyakov et al., 2018), we concluded that DRNET keeps more consistent background then MoCoGAN while the object is vague compared with constant background. MoCoGAN tracks the motion more precise than DRNET so that it has relatively clear object. Both of them combine motion into content directly and ignore the residual information of the object in disentangled parts. Object is the same as the background which has the same shape and profile without motion. Also, object is different from the background since motion only affects object.

Therefore, we proposed a slightly more complicated disentangled model Motion-Object-Background Disentangled Network (MOBNET) to reinforce the difference between background and object, which is a further separation of content. Our disentanglement is expected to have better performance in disentangling information from more complex videos since most videos involve varying backgrounds. To incorporate background information and evaluate separation quality, we constructed a new dataset and proposed new evaluation metric. We cover recent research processes in disentangled field of video and image segmentation methods in **Background & Related Work** section. In **Methods** section, we explained MOBNET. In **Evaluation Metric** section, we describe the construction and evaluation metric for our synthetic dataset in details. In **Experiments** section, we evaluate the disentangled quality of object in prediction task with DRNET (Denton et al., 2017) and MoCoGAN (Tulyakov et al., 2018), and demonstrate improvement in disentangled quality of our model. Besides, we verified that our model could accomplish prediction task of video with different object, background or motion.

*Equal contribution [1]Carnegie Mellon University, Pittsburgh, PA 15213, USA.

10708 Class Project, Spring, 2019.

2. Background & Related Work

The key idea behind the unsupervised learning of disentangled representation is that data in real world is generated by a few explanatory factors of variation which can be recovered by unsupervised learning. Disentangled representation is an effective approach to understand high-dimensional data through distilling knowledge into useful representation since it captures the independent features of a given scene where if one feature changes, the others remain unaffected. Several work has argued disentangled representation is an significant step towards better representation learning (Hsu et al., 2017; Lenc & Vedaldi, 2015; Peters et al., 2017). The idea of disentangled representation has already been explored for video in (Denton et al., 2017; Villegas et al., 2017; Tulyakov et al., 2018).

Disentangled-Representation Net(DRNET) (Denton et al., 2017) is a disentangled image representation model targeting at videos which utilizes two separate encoders to produce distinct representations of content and pose for each frame in the video. In order to induce the desired factorization between these two encoders and ensure that the pose features do not contain content information, they introduced a novel adversarial loss on the pose features preventing them from being discriminable from one video to another. The further task using the sequential image representation learned by DRNET includes forward motion prediction and motion classification. They put the pose features into an LSTM model to predict the following frames conditioning on the content feature extracted from the last observed frames. Although the model is intuitively simple compared to other video generation models, it can still obtain reasonable frame prediction in some instances.

Motion-Content Network (MCNet) (Villegas et al., 2017) is a deep generative model that tackles the frame prediction problem. Given the history up to the t^{th} frame, MCNet generates pixel value predictions for future frames. Built upon the Encoder-Decoder Convolutional Neural Network and Convolutional LSTM, this model decomposes content and motion, and captures the spatial information of an image and the corresponding temporal dynamics in an unsupervised fashion. The objective function of training this model is a weighted combination of generator loss in adversarial training and prediction loss. The limitation of this model is that the content encoder is a CNN using the spatial information of one image. Also, this model does not use any object level information, which we believe could be a powerful representation to eliminate blurry and other pixel deficiency in the generation.

Motion and Content decomposed Generative Adversarial Network (MoCoGAN) (Tulyakov et al., 2018) is another deep generative framework for video generations and frame predictions. It further provides a way to disentangle the content and motion information such that the framework is able to control each part for the generation process. They have experimented with generating videos for the same content with different motions and the same motion for different contents, and the results are visually reasonable. The network consists of four parts: a recurrent neural network that generate motion vectors for each time step, a Generator that takes the motion vectors with a content vector (sampled once for an entire video clip) to generate per frame images, a Discriminator that criticize individual images and another video-level Discriminator that provides criticism based on the entire video clip. In addition, the network proposed to model the categorical dynamics involved in videos by augmenting the input into the recurrent neural network with a categorical random variable. The objective function is therefore augmented by the lower bound of the mutual information between the random variable and the generated video clip. The constraints on the generated motions can be further improved by given categorical labels.

Video frame prediction has several solutions such as recurrent network architecture inspired from language modeling and LSTM model. As mentioned above, models like DRNET with adversarial loss still has blurry affect because it mix the object and background. It has been proved that generative adversarial training could be successfully applied for the next frame prediction (Mathieu et al., 2015). All these image representation models have their own limitations and specific applications. Inspirited by these models, disentangled object, motion ,and content in each frame is an interesting and possible direction to explore.

When it comes to the model evaluation, the Inception score is a popular metric for judging the image outputs of Generative Adversarial Networks. However, five shortcomings of the inception score have identified in (Barratt & Sharma, 2018). The problems with the Inception Score fall into two categories: 1. Suboptimalities of the Inception Score itself; 2. Problems with the popular usage of the Inception Score. Even though the metric is used in MoCoGAN and DRNET, it has limitation for the synthetic datasets which is far away from the training dataset ImageNet.

To solve the evaluation problem for the synthetic dataset, we proposed the evaluation metric based on the segmentation ideas in (He et al., 2017; Long et al., 2015). The fully convolutional network in (Long et al., 2015) takes input of arbitrary size and produces correspondingly-sized output with efficient inference and learning. Since the disentagled quality of object is our main concern, the elegant framework of object instance segmentation in (He et al., 2017) provides a basic thinking of object extraction.

3. Methods

Motion/object/background disentanglement

We propose the MOB encoder, an encoder network to jointly give latent embedding for **M**otion, **O**bject and **B**ackground information of video clips. Formally, for a video sequence $\boldsymbol{v} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$, we write $\Delta\boldsymbol{x}_t = \boldsymbol{x}_t - \boldsymbol{x}_{t-1}$ for $1 \leq t \leq T$, where $\boldsymbol{x}_0 = \mathbf{0}$, and take $\{\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_{T-1}\} = \{(\boldsymbol{x}_1, \Delta\boldsymbol{x}_1), \ldots, (\boldsymbol{x}_{T-1}, \Delta\boldsymbol{x}_{T-1})\}$ as input to generate latent embedding $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{T-1}\}$, where $\boldsymbol{z}_t = \left(\boldsymbol{z}_t^M, \boldsymbol{z}_t^O, \boldsymbol{z}_t^B\right)$. Three coordinates of $\boldsymbol{z}_t$ belong to motion, object and background subspace, respectively: $\boldsymbol{z}_t^M \in Z_M = \mathbb{R}^{d_M}$, $\boldsymbol{z}_t^O \in Z_O = \mathbb{R}^{d_O}$ and $\boldsymbol{z}_t^B \in Z_B = \mathbb{R}^{d_B}$. The background subspace models static components in videos, the motion subspace captures trajectory of moving objects, and the object subspace captures time-independent information of moving parts. Object and motion part can be used to produce the moving parts, and then combining with background part gives full video reconstruction. For example, in a video of moving ball and static background, Z_B represents the background, Z_M models trajectory of ball, and Z_O captures information about the ball itself.

We further assume $\{\Delta\boldsymbol{x}_t\}$ captures all the information of $\left\{\boldsymbol{z}_t^M\right\}$, and $\{\boldsymbol{x}_t\}$ can be used to give $\left\{\boldsymbol{z}_t^O\right\}$ and $\left\{\boldsymbol{z}_t^B\right\}$. Instead of forcing $\boldsymbol{z}_t^O$ and $\boldsymbol{z}_t^B$ to be constant across time, we use a similarity loss during training(see (Denton et al., 2017)) to make object and background almost time-invariant. For motion part, to learn physically plausible movement pattern, we use recurrent neural network structures to learn $\left\{\boldsymbol{z}_t^M\right\}$. This encoder framework is very flexible, and we provide a realization in Experiment Section and Figure 1. In the following text, we denote this encoder network by E_{MOB}, and write $E_{MOB}(\tilde{\boldsymbol{x}}) = \left(E_{MOB}^M(\tilde{\boldsymbol{x}}), E_{MOB}^O(\tilde{\boldsymbol{x}}), E_{MOB}^B(\tilde{\boldsymbol{x}})\right)$.

For training data video $\boldsymbol{v}$ in our experiment, we have ground truth for object mask $\boldsymbol{m}^t$, where $\boldsymbol{m}^t(i,j) = \mathbf{1}\{\text{pixel } (i,j) \text{ belongs to object part of } t\text{th frame of } \boldsymbol{v}\}$.
Use $A \circ B$ to denote the Hadamard product of matrix A and B, where $A \circ B(i,j) = A(i,j)B(i,j)$. Then the object image of $\boldsymbol{x}^t$ is $\boldsymbol{o}^t = \boldsymbol{x}^t \circ \boldsymbol{m}^t$. When $\boldsymbol{m}^t$ is unknown, we can use various image segmentation methods (like (He et al., 2017) and (Long et al., 2015)) to get estimated object mask.

Adversarial Training Network

With the MOB encoder, we are able to construct a adversarial learning network structure for video prediction. After the MOB encoder produces disentanglement embedding $(\boldsymbol{z}_t^M, \boldsymbol{z}_t^O, \boldsymbol{z}_t^B)$ for each time step, we put $(\boldsymbol{z}_t^M, \boldsymbol{z}_t^O, \boldsymbol{z}_t^B)$ into generator G to get predicted next frame $\hat{\boldsymbol{x}}^{t+1}$, and put $(\boldsymbol{z}_t^M, \boldsymbol{z}_t^O, \mathbf{0})$ into the same generator G to produce predicted next object image $\hat{o}^{t+1}$. Then the three discriminators, namely, image discriminator D_I, object discriminator D_O and video discriminator D_V take pair $(\hat{\boldsymbol{x}}^{t+1}, \boldsymbol{x}^{t+1})$, $(\hat{\boldsymbol{o}}^{t+1}, \boldsymbol{o}^{t+1})$ and $(\left\{\hat{\boldsymbol{x}}^{t+1}\right\}, \left\{\boldsymbol{x}^{t+1}\right\})$ as input, respectively. Among these networks, E_{MOB} and G try to fool the discriminator, while three discriminators try to distinguish the output of G, thus an adversarial learning framework is constructed. See more details in Figure 1.

Ideally, D_V should be enough for training G and E_{MOB}, because D_V provides feedback on both static image and dynamic mechanism. However, D_I and D_O significantly improve the convergence and performance of training and prediction, because they are more focused on image/object.

Loss function and training: First we have the adversarial loss of three discriminators. Note that $\hat{\boldsymbol{v}}$ is predicted video sequence $\{\hat{\boldsymbol{x}}^t\}$.

$$\mathcal{F}_I(E_{MOB}, G, D_I) = \sum_t \left[-\log D_I(\boldsymbol{x}_t) - \log(1 - D_I(\hat{\boldsymbol{x}}_t))\right],$$

$$\mathcal{F}_O(E_{MOB}, G, D_O) = \sum_t \left[-\log D_O(\boldsymbol{o}_t) - \log(1 - D_O(\hat{\boldsymbol{o}}_t))\right],$$

$$\mathcal{F}_V(E_{MOB}, G, D_V) = -\log D_V(\boldsymbol{v}) - \log(1 - D_V(\hat{\boldsymbol{v}})).$$

Inspired by (Denton et al., 2017), we use similarity loss to enforce background and object to be approximately time-independent. Similarity loss for video $\boldsymbol{v}$ is defined as

$$\mathcal{L}_{sim}(E_{MOB}) = \sum_t \left[\|E_{MOB}^O(\boldsymbol{x}^t) - E_{MOB}^O(\boldsymbol{x}^{t+1})\|_2^2 + \|E_{MOB}^B(\boldsymbol{x}^t) - E_{MOB}^B(\boldsymbol{x}^{t+1})\|_2^2\right].$$

We also use the standard ℓ_2 loss between predicted future frame(both with and without background) $\hat{\boldsymbol{x}}^{t+1}$ and the actual future frame $\boldsymbol{x}^{t+1}$ as reconstruction loss:

$$\mathcal{L}_{rec}^{obj}(E_{MOB}, G) = \sum_t \left[\|G(E_{MOB}^M(\tilde{\boldsymbol{x}}^t), E_{MOB}^O(\tilde{\boldsymbol{x}}^t), \mathbf{0}) - \boldsymbol{o}^{t+1}\|_2^2\right],$$

$$\mathcal{L}_{rec}^{img}(E_{MOB}, G) = \sum_t \left[\|G(E_{MOB}^M(\tilde{\boldsymbol{x}}^t), E_{MOB}^O(\tilde{\boldsymbol{x}}^t), E_{MOB}^B(\tilde{\boldsymbol{x}}^t)) - \boldsymbol{x}^{t+1}\|_2^2\right].$$

With this notation, the overall training objective of the network becomes

$$\max_{\{G, E_{MOB}\}} \min_{\{D_I, D_O, D_V\}} \left[\alpha(\mathcal{F}_I + \mathcal{F}_O + \mathcal{F}_V) + \mathcal{L}_{sim} + \mathcal{L}_{rec}^{obj} + \mathcal{L}_{rec}^{img}\right],$$

with each term as addressed above. We train the MOBNET using the alternating gradient update algorithm as in (**?**). Specifically, in one step, we update D_I, D_O, D_V one by one while fixing E_{MOB} and G. In the alternating step, we update E_{MOB} and G while fixing D_I, D_O and D_V.

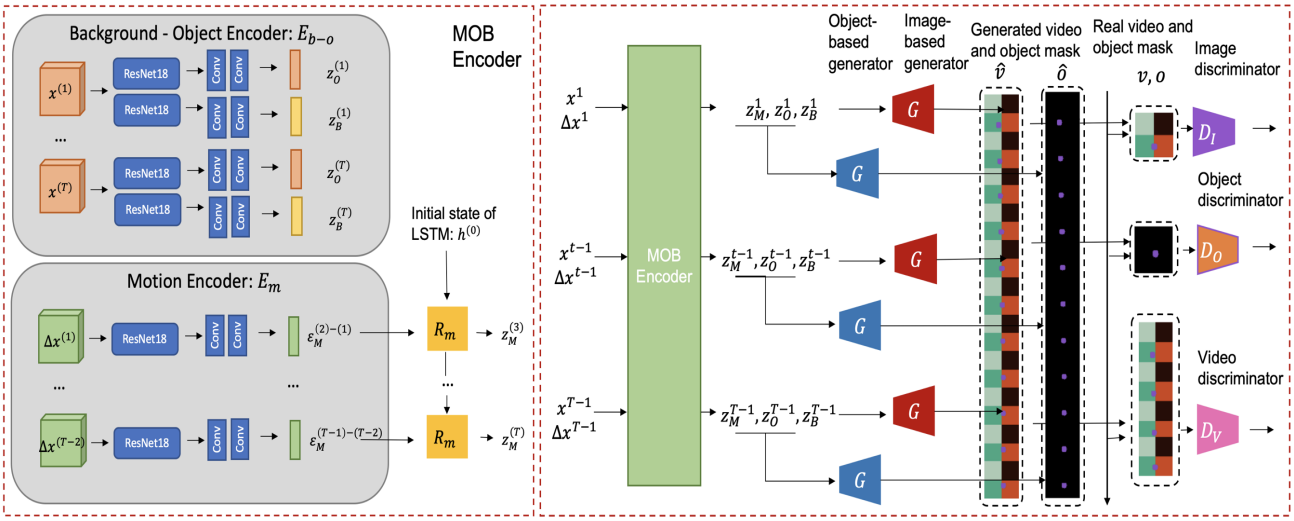

Figure 1. Adversarial network training network structure and MOB encoder. The MOB encoder produces embedding for object and background directly after ResNet and convolution layers, while motion embedding needs additional GRU sequence. After MOB encoder, object and motion produce predicted object image, and further adding background in the same generator yields predicted next frame. The predicted image, object and whole video then serve as input for three discriminators along with their ground-truth. While training, discriminators and encoder/generator are updated alternatively.

Networks: MOBNET consists of 5 sub-networks: E_{MOB}, G, D_I, D_O and D_V. The structure of E_{MOB} is shown in Figure 1; the R_M part could also be $LSTM$ or GRU cell. In our implementation, we use ResNet followed by convolution layers for encoding, and GRU cells for forward step. Roughly, we use convolution layers for discriminators and conv2D layers for G; see details in Experiment Section.

Forward prediction and generation : The proposed network has a natural way to predict future frames of video. We can repetitively take into network predicted $t+k$ frame $(\hat{\boldsymbol{x}}^{t+k}, \Delta\hat{\boldsymbol{x}}^{t+k})$ to get predicted frame $\hat{\boldsymbol{x}}^{t+k+1}$. The model could also be modified to have generation ability, in the same way as (Tulyakov et al., 2018).

Variations: Aside from the main model structure, we have experimented with multiple ways to enforce the disentanglement between object and motion vectors. Our basic idea is to generate images only with object vectors while setting the background vectors and motion vectors all to zero, and provide an additional objective in optimizing the images to be valid objects. This is a difficult task since we do not have the "ground truth" for the generated object images, so we decided to use the similar idea from GAN by applying a discriminator loss and a generator loss. We have tried to simply add the losses for the generated object images to the original loss, but the experiment results show that the objects generated are still visually far from the true objects and it frequently falls into loopholes where blank images are generated. This means that the object information is actually encoded in the motion vectors, which is not as we have expected. The current approach is to substitute the original GAN objectives on object images, which are generated by object and motion vectors together, with GAN objectives on the images generated only by object vectors. In this way, the losses will only be able to back-propagated to the object vector and we can therefore provide some level of enforcement on the information that the object vector should contain. However, as we are still running some related experiments, current results are not as convincing. It is difficult to balance between the reconstruction losses and the generator losses in the training process, so we still maintained the original structure, but we will look into this direction in future works.

We also experimented with how different color space could affect the results. Along this horizon, we tested with formulating the reconstruction loss and using data in either on RGB or LAB color space. The former is common yet the latter is interesting to explore. It is argued that the LAB color space is more preferred by human vision system. Hence, same Euclidean distance in this space as in RGB space could be more preferred by human. We explored this horizon but the results are not as expected. We found that using LAB space, the reconstruction loss fails to go as low as the one of using RGB.

4. Evaluation Metric

We investigated evaluation metrics used by DRNET (Denton et al., 2017), MCNET (Villegas et al., 2017), and MoCoGAN (Tulyakov et al., 2018). Also, we proposed a new evaluation metric to evaluate generation and disentanglement quality of background, object, and motion separately.

Three major evaluation metrics are used for quantitative comparison: structural similarity index measure (SSIM) (Wang & Gupta, 2015), peak signal to noise ratio (PSNR) (Mathieu et al., 2016), and inception score (Salimans et al., 2016). The first two can be used when ground truth is available. They are variants of L2 pixel-wise distance and shown to be better as a metric for evaluating generative/predictive model (Mathieu et al., 2016). Inception score is used when ground truth is not available and one wants to look at the quality of the generation. The idea is to first train a temporal-spatial CNN classifier that accepts a video as input and output the class of it. For example, for a motion dataset, the classes are walking, jogging, etc.

DRNET mainly used inception score. It is compared with MCNET on KTH dataset (Schuldt et al., 2004). The author argues that this is a better metric compared to SSIM and PSNR, because the latter two put too many weights on the accuracy of velocity prediction. Even though the object generated remains sharp along the motion, a small error in velocity prediction could contribute to large error in the two metrics. MoCoGAN also used inception score. It is compared with VGAN and TGAN on KTH dataset. MCNET mainly used SSIM and PSNR for comparison with its baseline model (conv+LSTM), several variants, and a "copy from last frame" baseline on UCF101, KTH and WEIZMANN action datasets. Note that DRNET and MCNET are predictive models, in which first several frames are used to predict later ones. However, MoCoGAN is a generative model, whose object is to learn a distribution from which we can directly generate videos.

Therefore, we propose new evaluation metrics for specifically for the prediction task. To begin with, we first need a separate image segmentation network that is able to extract the position of the objects and information of the background simultaneously. The network is trained with ground truth positions, which our new shape motion dataset is able to provide. With the trained image segmentation model, we are then able to extract the objects from the background and perform further evaluations on either objects or their background separately. The network structure we use are shown in Figure 2.

For prediction tasks, we have the ground truth image for the time frame, so we proposed the evaluation metric as pixel-level relative distances, similar to L_2 distance. We apply the image segmentation network on generated images and ground truth images to extract object pixels for each. At the same time, background pixels are extracted as well. For color reconstruction evaluation, consider the object part, we compute an L_2 distance using common part of the two extractions in RGB space. Similarly we compute this distance for the background part. We could have directly using the ground truth as one of the extractions but we apply the segmentation model anyway to remove effects of imperfect performance of the image segmentation model. For motion evaluation, the error is represented as how well aligned the two extractions for object are. Hence, we formulate the error using the parts of extractions that are not overlapped. Specifically, we count the numbers of pixels that are not overlapped.

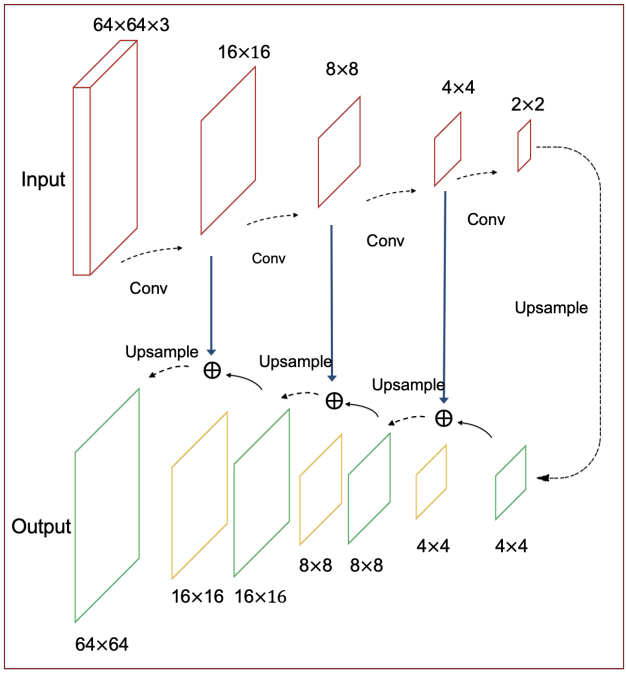

Figure 2. Structure of the segmentation network we use. We use a ResNet50 backbone.

In terms of inception score, we find that it does not work on our new shape dataset. We find that all motions are measured to have same score, even using the ground truth data as input. Hence, we did not proceed with inception score on our evaluation.

In our experiments below, we use the following evaluation scores for M/O/B part:

$$\ell_{obj} = \gamma \ell_{position} + \ell_{obj-RGB},$$
$$\ell_{bgd} = \ell_{bgd-RGB},$$
$$\ell_{total} = \lambda_1 \ell_{obj} + \lambda_2 \ell_{bgd}$$

5. Experiments

Dataset Construction

To explore the effect of varying interactions between background and objects, we constructed a new dataset based on the original shape dataset mentioned in (Tulyakov et al., 2018). The original dataset was consist of moving shapes (circles and squares) with different sizes and colors, but the background was all set to black. The motion was sampled from Bezier curves from vertical and horizontal movements. To introduce additional background information, we separated the image into 4 blocks and filled them with randomly sampled colors different from the shape colors. In total, there were 8000 videos, the image resolution was 64×64 and the video length was 32. For experiments with MoCoGAN, the images were up-sampled to 96×96. Example videos are shown in Figure 3.

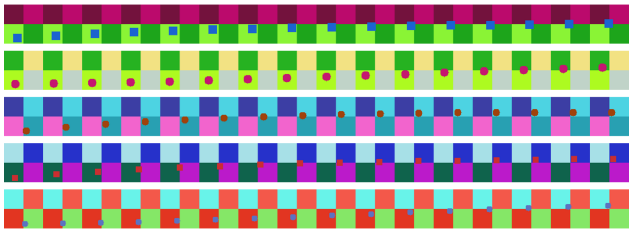

Figure 3. 5 example video sequences from the generated synthetic dataset. The background color is randomly sampled, and the trajectory of the object follows Bezier curves.

Implementations

The dataset construction is implemented in MATLAB. For the image segmentation model and the proposed MOBNET are implemented using PyTorch framework and are trained on a GeForce GTX 1080 Ti GPU. For DRNET, we used Adam optimizer with 0.002 learning rate; for the image segmentation network, we used SGD with a learning rate of $3e^{-5}$ and a weight decay of 0.005, and for the training of our proposed model, we used Adam optimizer with a learning rate of 0.0002, a momentum of 0.5 and 0.999 and a weight decay of $1e^{-5}$. The same optimizer setting is applied to all network modules, including the feature encoder, the generator and the discriminators. Logs are recorded every 500 iterations, and we monitored the losses trends with TensorBoard. Experiment results in the following sections are from the best model so far. Training progress is shown in Figure 4.

As for detailed network structures, we referred to the official DRNET implementation on GitHub. Both the content and the pose encoder are 4-layer convolutions. We set `choose_content = 128` and `pose_dimension = 10`, and trained the network with a batch size of 64.

For our MOBNET, we referred to the official PyTorch ResNet18 implementation as our backbone and we extended three heads for the encoded feature vectors, which are all 128 dimension. The generator and the discriminators are of similar structures as in MoCoGAN. Some necessary changes are made to ensure the correspondence with image sizes. During training, each video is taken into the network as a batch.

For the segmentation network, we referred to the official PyTorch implementation for ResNet50 as the backbone and the upsample layers are implemented based on (Long et al., 2015). In addition, we added an additional upsample layer for more precise results and it is proved to be better from our experiments.

Qualitative Evaluation

In this section we first present the disentanglement that our model achieves by swapping background embeddings around and observing the generated videos. Then, we show comparison results with DRNET, the baseline model.

Swapping disentangled embeddings

To see how well the disentanglement is, we generate videos by using same $z_{o,m}$, motion object embeddings, and different z_b, background embeddings. Also we generated videos with same background embeddings and different motion object embeddings. Through the simple sample video shown in Fig 5 we could conclude that video generated through MOBNET did not suffer blurry problem compared with original video since object in each frame of generated vides in the second line is still sharp compared with original video. Disentangled object and background provides a clearer individual part in each frame so that we can keep the background embeddings consistent and apply the motion embeddings to the object embeddings. The tricky task of separating background and object in MOBNET solve the blurry problem in video prediction task in some certain.

The third line in Fig 5 remains only motion and object in each frame with blank background. Combining motion and object embeddings with different background proves the disentangled quality of object and background in MOBNET. As shown in the fourth line, swapping background embeddings leads to the blurry problem of object with deeper color and smaller outline which is caused by the overlapping pixels in the background and object. However, same background with different object motion embeddings has better performance than same object motion embeedings with different background. Considering the fact that background embeddings have more constant feature in video, motion object embeddings are more complicated parts in each frame. The performance of two type swapping has proves that MOBNET is able to disentangle motion, object and background in video and generate sharp future frame

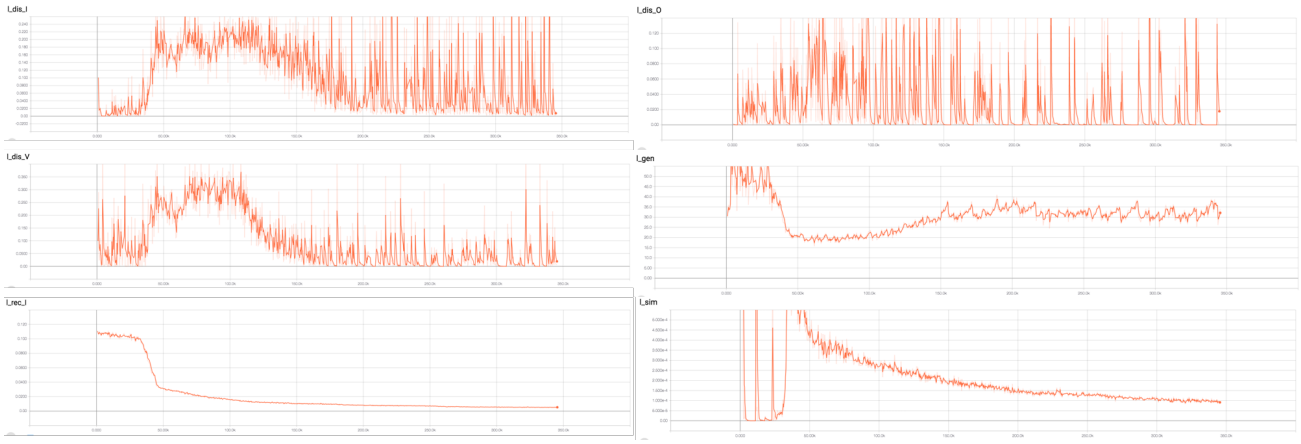

Figure 4. Training progress in which the change of several training objectives are shown: l_dis_I, l_dis_O, and l_dis_V are discriminator losses. l_gen is generator loss. l_rec is the sum of reconstruction losses of images and objects only. l_sim is the similarity loss enforced on background and object embeddings.

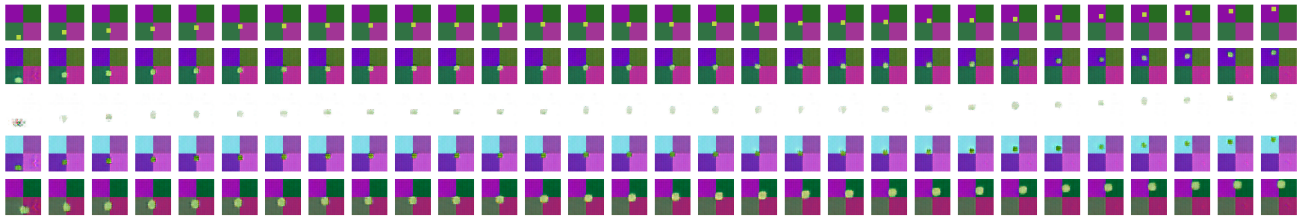

Figure 5. First line is real video; Second line is generated video by MOBNET; Third line is frame sequences with object vector and motion vector; Fourth line is result of swapping background only; Fifth line is result of swapping object and motion. Compare with other methods, MOBNET has significantly better performance in learning motion, and ability to transfer motion information to other videos.

with several settings of background embeddings and motion object embeddings.

Comparison with DRNET

To compare our model with DRNET, we generated prediction results for both models on the same set of ground truth images. As shown in Figure 6, there are two sets of experiments. The first line is the ground truth images, the second line is the prediction results from DRNET and the third line is the generated images from our MOBNET. Both models used the first 22 frames to predict for the next 10 frames. From the figure, we can see that DRNET is able to predict the image colors accurately where our MOBNET may generate results with varying colors. However, motion wise, we can clearly observe that the objects are treated by DRNET as an entity together with the background, so there is no motion involved on the objects. On the contrary, our MOBNET is able to model the motions accurately, which also partially proves that the motion is only applied on the objects, which are separated from the static backgrounds.

Furthermore, we apply our segmentation mask network to the generated figures of DRNET, MOBNET as well as original figures. See results in Figure 7. From the figures, we can see that our segmentation network has reasonably good performance when the edge of object is relatively clear; also MOBNET has better performance than DRNET from the segmented masks. Because DRNET only consider the pose and content which keep the information of object with background and motion, DRNET is easy to cause the blurry problem which is solved in MOBNET. MOBNET improve the disentangled quality of object and background so that it has better performance in the segmentation mask network.

Quantitative Evaluation

We evaluated our results quantitatively using the segmentation network, constructed metric, and compared it with results from DRNET. The results are shown in Table 1. The accuracy is computed as the correct predictions of the object class in the segmented results. The custom loss is the one defined by ourselves as introduced in the section of evaluatrion metric. Results are averaged over 1000 videos from the test dataset. We see that MOBNET achieves better accu-

racy but lower custom loss. This is because the custom loss involves the RGB pixel measurements, which our network could possibly perform worse.

Table 1. Comparsion with DRNET using different evaluation metrics.

METHOD	OVERALL ACCURACY	CUSTOM LOSS
MOBNET	0.396	108.4456
DRNET	0.389	156.3913

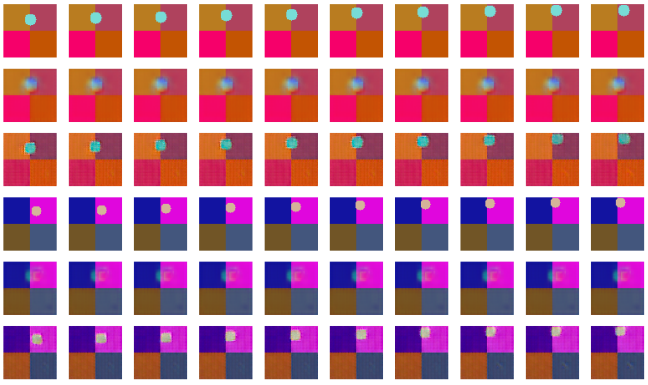

Figure 6. Comparison with DRNET. In each group of pictures, the first, second and third line are ground truth, prediction results for DRNET and prediction results for MOBNET, respectively. From the prediction results, we can see that MOBNET outperforms DRNET overall, and has especially good performance on motion detection.

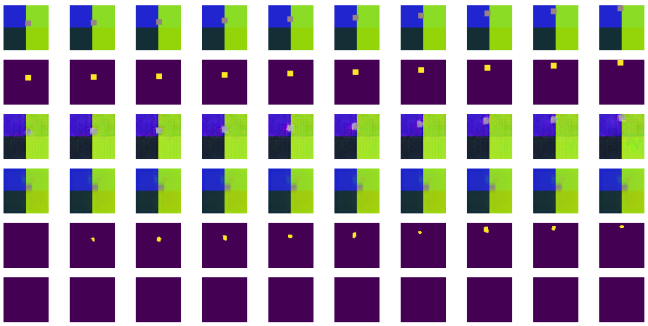

Figure 7. Segmentation Result for original video, MOBNET and DRNET. Line 1 and 2: ground truth and segmentation. Line 3 and 5: MOBNET results and segmentation; line 4 and 6: DRNET results and segmentation. From the figures, we can see that our segmentation network has reasonably good performance when the edge of object is relatively clear; also MOBNET has significantly better performance than DRNET.

6. Future Work and Conclusion

Our method can be extended in various ways. The network structure we use are relatively simple, and when applied to real-world dataset, more complex network designs could be tried to improve performance. Our model is different from GAN since there is no sampling procedure, but the structure could be easily adjusted to a GAN framework, thus we can conduct video generation task. Also it remains an interesting topic to discuss the connection between disentanglement and interpretation, a question we tried to answer but far from being solved. And a proper learned latent embedding might help in many tasks like classification.

In conclusion, we build a new network structure disentangling motion, object and background of video, and show the improvement in performance. Further, we try to answer what the disentanglement has learned, by evaluating the learning performance on M/O/B part separately, and by transforming the elements across different videos. Along with our constructed data-set, we provide an inspiring example on how to design disentanglement learning framework, and how to evaluate the performance.

References

Barratt, S. and Sharma, R. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

Denton, E. L. et al. Unsupervised learning of disentangled representations from video. In *Advances in Neural Information Processing Systems*, pp. 4414–4423, 2017.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Hsu, W.-N., Zhang, Y., and Glass, J. Unsupervised learning of disentangled and interpretable representations from sequential data. In *Advances in neural information processing systems*, pp. 1878–1889, 2017.

Lenc, K. and Vedaldi, A. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 991–999, 2015.

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

Mathieu, M., Couprie, C., and LeCun, Y. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

Mathieu, M., Couprie, C., and LeCun, Y. Deep multi-scale video prediction beyond mean square error. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.05440.

Peters, J., Janzing, D., and Schölkopf, B. *Elements of causal inference: foundations and learning algorithms*. MIT press, 2017.

Salimans, T., Goodfellow, I. J., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016. URL http://arxiv.org/abs/1606.03498.

Schuldt, C., Laptev, I., and Caputo, B. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pp. 32–36. IEEE, 2004.

Tulyakov, S., Liu, M.-Y., Yang, X., and Kautz, J. MoCoGAN: Decomposing motion and content for video generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1526–1535, 2018.

Villegas, R., Yang, J., Hong, S., Lin, X., and Lee, H. Decomposing motion and content for natural video sequence prediction. *CoRR*, abs/1706.08033, 2017. URL http://arxiv.org/abs/1706.08033.

Wang, X. and Gupta, A. Unsupervised learning of visual representations using videos. *CoRR*, abs/1505.00687, 2015. URL http://arxiv.org/abs/1505.00687.