Enhancing Unsupervised Video Representation Learning by Decoupling the Scene and the Motion

Jinpeng Wang[1, 2*], Yuting Gao[2*], Ke Li[2], Jianguo Hu[1], Xinyang Jiang[2], Xiaowei Guo[2], Rongrong Ji[3], Xing Sun[2†]

[1] Sun Yat-sen University, Guang Zhou, China
[2] Tencent Youtu Lab, Shanghai, China
[3] Xiamen University, Xiamen, China
winfredsun@tencent.com

Abstract

One significant factor we expect the video representation learning to capture, especially in contrast with the image representation learning, is the object motion. However, we found that in the current mainstream video datasets, some action categories are highly related with the scene where the action happens, making the model tend to degrade to a solution where only the scene information is encoded. For example, a trained model may predict a video as playing football simply because it sees the field, neglecting that the subject is dancing as a cheerleader on the field. This is against our original intention towards the video representation learning and may bring scene bias on a different dataset that can not be ignored. In order to tackle this problem, we propose to decouple the scene and the motion (DSM) with two simple operations, so that the model attention towards the motion information is better paid. Specifically, we construct a positive clip and a negative clip for each video. Compared to the original video, the positive/negative is motion-untouched/broken but scene-broken/untouched by *Spatial Local Disturbance* and *Temporal Local Disturbance*. Our objective is to pull the positive closer while pushing the negative farther to the original clip in the latent space. In this way, the impact of the scene is weakened while the temporal sensitivity of the network is further enhanced. We conduct experiments on two tasks with various backbones and different pre-training datasets, and find that our method surpass the SOTA methods with a remarkable 8.1% and 8.8% improvement towards action recognition task on the UCF101 and HMDB51 datasets respectively using the same backbone.

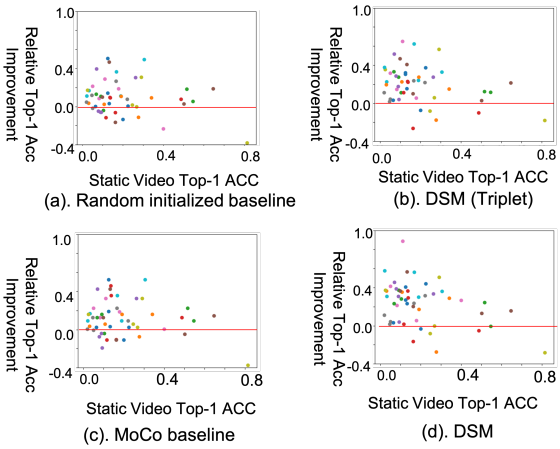

Figure 1: Relative Top-1 accuracy improvement using the normal video over the artificially constructed static video. Each dot represents a semantic category. It can be seen from (a) that for some of the categories, *i.e.*, dots below and around the red line, using a normal video does not help the model to give better prediction. This means that the scene and the motion are coupled together and models can give correct prediction by merely learning the scene. This phenomenon is relieved by our DSM method in (b) and (d), where the average relative improvement is notably lifted, meaning that the motion information is better exploited by the model. For a fair comparison, a MoCo baseline is given in (c).

Introduction

Unsupervised representation learning has received widespread attention in the last few years. In the field of image representation learning, recent approaches(He et al. 2020; Chen et al. 2020) have nearly surpassed their supervised counterparts. Nevertheless, in the field of video representation learning, there still exists a gap between unsupervised methods and supervised methods. In contrast with the image representation learning, one significant factor we expect video representation learning to capture is the *object motion*, since a video usually contains continuous states of an object. While most commonly used image classification datasets are *object-dominated*, *i.e.*, object occupies a major part of the image, video datasets are usually *scene-dominated*, *i.e.*, object is relatively small and the discriminative information contained in the object motion is sometimes overwhelmed by statistics of the scene. To verify the side effect of this phenomenon, we conduct

*The first two authors contributed equally. This work was done during Jinpeng Wang's internship at Tencent Youtu Lab.
†Corresponding Author

an experiment on the HMDB51(Kuehne et al. 2011) dataset by training models with two types of clips. One type of the clips are formed by repeating a frame that is randomly selected from a video, which have lost motion information and only the scene and static status are left. The other type of the clips are normal videos, which contains motion as expected. During testing, all samples are normal clips and results are shown in Figure 1. Generally, we would expect all categories to show up upon the red line, meaning that motions are one of the key factors for video representation learning. However, we find that there are nearly 24% of the categories show less than 5% or no improvement with the help of motion information. This phenomenon may cause the model being lazy and only learn the scene without paying attention to the motion patterns that are what really matters. At the same time, certain actions in some datasets only occur in specific scenes, making the model prone to couple the motion pattern with the scene tightly. For example, a trained model may binding *squats* to *gyms*, thus misjudges when *squats* happen in other scenes. It means that although scene and motion do promote each other sometimes, a strong couping between the two may make the learned representations generalize poorly and are easy to overfit to a specific training set.

To alleviate the scene bias problem, many efforts have been paid in the supervised setting. Simonyan et al.(2014) and Feichtenhofer et al.(2019) use a two-way convolutional neural network to capture appearance feature and temporal characteristic respectively at a cost of computation complexity. Zhao et al.(2018) propose a new ConvNet architecture which can derive disentangled components, *i.e.*, static appearance, apparent motion and appearance changes, from low-level visual feature maps. Girdhar et al.(2020) build a synthetic video dataset with observable and controllable scene bias, forcing the model to understand both the spatial and temporal information to give correct prediction. Choi et al.(2019) propose to mitigate scene bias by augmenting the standard cross-entropy loss with an adversarial loss for scene type and a confusion loss of human mask. Wang et al.(2018) explicitly pulls actions from context through an auxiliary two class classifier.

In this work, we try to alleviate the *scene-dominated bias* in an unsupervised manner and propose to decouple the scene and the motion (DSM) with two simple operations. Specifically, we formulate self-supervised video representation learning as a data-driven metric learning problem and construct a positive clip and a negative clip for each video. Compared to the original video, the positive/negative is motion-untouched/broken but scene-broken/untouched by *Spatial Local Disturbance* and *Temporal Local Disturbance*. Our objective is to pull the positive closer while pushing the negative farther to the original clip in the latent space. In this way, our model is more scene independent and more motion sensitive. As shown in Figure 1(b) and (d), our method notably improve the overall feature representation ability, especially for categories that strongly rely on temporal information.

Our contributions are summarized as follows:

- We formulate the self-supervised video representation learning into a data-driven metric learning, and decouple the scene and the motion to alleviate the negative impact of the scene and the motion coupling problem which is commonly seen in the current video datasets.
- We design two effective strategies to construct positive and negative sample pairs which consider both the spatial and the temporal characters of the video data.
- Our method greatly improves the performance of unsupervised video representation learning and achieves state-of-the-art results on both UCF101(Soomro, Zamir, and Shah 2012) and HMDB51(Kuehne et al. 2011) datasets.

Related Work

Video Representation Learning

The most significant characteristic of the video representation learning is the requirement for temporal modeling compared to the image representation learning. Early works first use 2D CNNs to capture appearance features at the frame level and then do average pooling or adopt LSTM over the temporal dimension to learn motion patterns(Wang et al. 2018; Zhou et al. 2018). Another type of method(Simonyan and Zisserman 2014) use a two-way ConvNet to capture spatial appearance features and temporal motion patterns respectively. As a natural evolution, 3D CNNs (Tran et al. 2015; Carreira and Zisserman 2017; Hara, Kataoka, and Satoh 2018) are later used to capture spatio-temporal patterns at the same time. Feichtenhofer et al.(2019) uses two-pathway on the basis of 3D network and achieves good results. However, Li et al.(2018) and Girdhar et al.(2020) point out that the current commonly used video datasets are plagued with implicit biases over scene and object structure and they propose two datasets, which requires a complete understanding of spatio-temporal information for a model to give correct prediction. Choi et al.(2019) and Wang et al.(2018) propose to mitigate scene bias from the perspective of training strategy.

Self-supervised Learning

Self-supervised learning has received extensive attention in the field of image classification. One common way is to define a pretext that are related to downstream tasks(Noroozi and Favaro 2016a; Gidaris, Singh, and Komodakis 2018; Jenni, Jin, and Favaro 2020). Another mainstream type is based on metric learning, which aims to minimize the distance between similar samples while pushing away dissimilar samples in the feature space. Contrastive loss(Hadsell, Chopra, and LeCun 2006) proposes to decrease the distance between positive pairs while pushing the negative pairs to a certain margin. Triplet loss(Schroff, Kalenichenko, and Philbin 2015) makes further improvements by introducing triplets, which minimizes the distance between an anchor and a positive sample and maximizes the distance between the anchor and a negative sample. Recent works based on contrastive loss (Wu et al. 2018; He et al. 2020; Chen et al. 2020) have achieved excellent results on multiple visual

tasks and narrowed the gap between the supervised learning and the unsupervised learning. The core idea in contrastive learning is to strengthen the invariance of the network to various data augmentations. In this article, we explore video representation learning under the framework of self-supervised learning.

Self-supervised Video Representation Learning

Recently, many self-supervised video representation learning methods have been proposed. Among them, one prominent direction is to design a surrogate signal that can been used as supervision such as sequence order of frames(Misra, Zitnick, and Hebert 2016), space-time cubic puzzles (Kim, Cho, and Kweon 2019), video clip order (Xu et al. 2019; Luo et al. 2020) and video playback rating (Yao et al. 2020; Benaim et al. 2020). Besides, Gan et al.(2018) and Wang et al.(2019a) use the statistics of optical flow as supervision. Another mainstream category is based on contrastive learning, whose core is to construct positive samples and negative samples. TCN(Sermanet et al. 2018) treats the same actions under different cameras as positive samples and different time periods of the same video as negative samples. IIC(Tao, Wang, and Yamasaki 2020) regards multi-modal data as positive samples, and videos with shuffled frame order as negative samples. CVRL(Qian et al. 2020) proposes temporally consistent spatial augmentation with simple operations and treats the generated results as positive samples. In this work, we design two simple but effective strategies to construct positive and negative samples.

Methodology

We address video representation learning in a self-supervised manner, and the core idea is to learn an embedding space in which temporally similar/dissimilar but context-variant/-invariant video clips are close/far. In particular, we propose two simple but powerful augmentation strategies to construct clip pairs from the spatio-temporal structure of a single video. In the following sections, we first give an overview of the entire architecture, and then introduce the augmentation methods and objective functions in details.

Overall Architecture

The entire framework is shown in Figure 2. Formally, given an unlabeled video dataset X that contains N videos, we sample T frames from the video for each clip and input the clip to the network. Random cropping is performed on each input to generate three clips with different spatial regions but maintaining temporal consistency, denoted as c_1, c_2 and c_3. Afterwards, we apply basic augmentation b, spatial warping s and motion disturbance t on these three clips respectively to construct a triplet, *i.e.*, anchor $a = b(c_1)$, positive sample $p = s(c_2)$ and negative sample $n = t(c_3)$. Compared to a, p destroys the structural information of scene but maintains temporal semantics. Meanwhile, n disturbs the local motion pattern of the moving subject but retains the spatial information. This triplet is fed into a 3D backbone f to extract

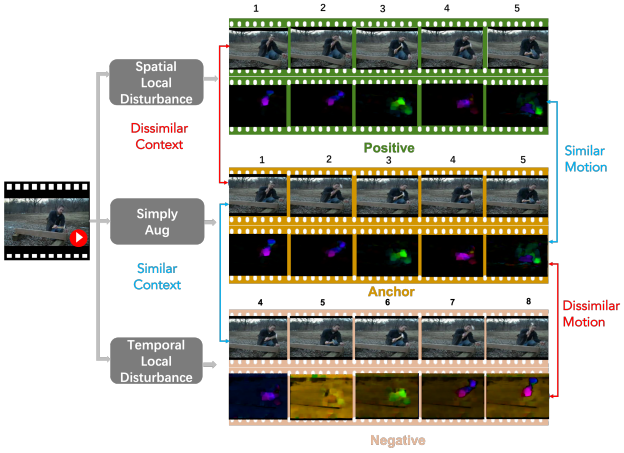

Figure 2: Overview of our method. We first construct positive and negative samples. Positive samples are constructed by *Spatial Local Disturbance* while negtive samples are constructed by *Temporal Local Disturbance*. Then we propose to learn video representations by training a convnet to push away temporal-dissimilar/spatial-similar pairs but pulling temporal-similar/spatial-dissimilar pairs closer.

spatio-temporal features which are then projected to a D-dimension feature space followed by L2 normalization, and we denote the decoded feature as z_a, z_p and z_n. In this way, the triplet is projected into a normalized embedding space $(z_a, z_p, z_n) \in \mathcal{R}^D$. We then perform spatio-temporal representation learning in the normalized embedding space using two metrics: intra-video triple loss and contrastive loss, which will be introduced in details in the later section.

Spatial Local Disturbance

The core idea of positive sample construction is to break local contexts while keeping motion semantics basically unchanged with data augmentation. Spatial data augmentations, *e.g.*, rotation, color jittering, have been widely used in the image-related task. However, it is underexplored in the video domain. A naive way for video augmentation may be applying existing image spatial augmentations to each frame of the video. However, some of these operation may damage the motion semantic. For example, if different rotation angles are used for consecutive frames, the generated video will looks like suffering a severe camera shake, making the video difficult to recognize. In order to make the temporal abstraction of the entire video remains similar, all consecutive frames of a video should perform the same transformation, and a video data augmentation that meets this requirement is *temporally consistent*.

Thin-Plate-Spline (TPS) is widely used in the OCR field to rectify distorted text regions(Jaderberg et al. 2015; Shi et al. 2016). In contrast, we aim to damage the statistics of the scene but keep motion pattern unchanged with TPS. Specifically, we select N uniformly distributed destination points on the target video as $\mathcal{D} = \{d_i\}_{i=1,\dots,N} \in \mathbf{R}^{2\times N}$. For each destination point d_i, we add a small offset $\Delta_i =$

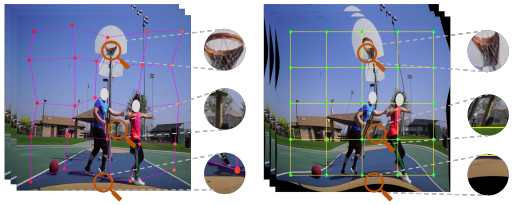

Figure 3: Illustration of the Spatial Warping, which randomly warps spatial regions in each epoch. Though the local statistics is broken, the global statistics is maintained.

$[\Delta x_i, \Delta y_i]^T$ to generate a corresponding source point $s_i = d_i + \Delta_i$ on the original video, making up a source point set $\mathcal{S}$. Both the horizontal and vertical offsets are randomly sampled from $[-C, C]$, where C is typically one-tenth of the frame size. Then s and d are used to compute the parameters of TPS. At last, the grid P is computed by the TPS transformation and the final warped video is generated by a bilinear sampler given P and the original video. An illustration of spatial warping is shown in Figure 3. In this way, although spatial local statistics are modified, the global context is maintained. In each training epoch, due to the randomness of the grid, the generated warped videos are different and the pixels in the local area always show huge differences from the original videos. Therefore, the network needs to focus more on motion pattern and pay less attention to spatial changes to extract consistent representations for the original and warped videos. Since all frames in the video perform the same transformation operation, it is a *temporally consistent* augmentation.

Temporal Local Disturbance

In contrast to p, the major difference between a and n is the motion pattern. A straightforward idea may be using other videos directly as n as in recent contrastive learning method (Sermanet et al. 2018). However, besides the temporal information, there still exists many artificial cues to distinguish two videos, which are easier to solve for the network (Kim, Cho, and Kweon 2019). Therefore, it is not guaranteed that the network will focus on the motion. To overcome this limitation, we propose to generate n with large temporal abstraction differences but similar context from a using *Temporal Local Disturbance* (TLD). TLD comprises two transformations and are described in details as below.

Optical-flow Scaling. We first denote a video as $I(x, y, t)$, where x, y are spatial coordinates and t is time. Under the brightness constancy assumption (Horn and Schunck 1981), the relation between I and the corresponding optical flow (V_x, V_y) can be formulated as:

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t} = 0, \tag{1}$$

where V_x and V_y are the horizontal and vertical components of the velocity. It should be noted that given a video frame at time t and the corresponding optical flow, we can compute the $t+1$ frame by:

$$I(x, y, t+1) = I(x + V_x, y + V_y, t) \tag{2}$$

By applying the equation, we can accelerate or decelerate the video motion without changing background pixels too much. In particular, given a scale factor $\phi(t)$, we have

$$\hat{I}(x, y, t+1) = I(x + \phi(t)V_x, y + \phi(t)V_y, t) \tag{3}$$

where $\phi(t)$ is randomly sampled from [0, M] over time t, and M is a hyperparameter controlling the amplitude. We find that a too big M may result in unnatural videos with wide black boundary, and setting M to 5 gives best result.

Temporal Shift. The purpose of Temporal Shift is to distinguish the temporal differences of various videos that contain similar scene. We assume that a video and its corresponding temporal shifted version has different motion patterns. Given a video x, we randomly and uniformly sample a shift scalar τ from $[\alpha_1, \alpha_2]$, then the new video is generated by:

$$\hat{x}_i = x_{i+\tau}, i \in 1, 2...T \tag{4}$$

That is, we extend the origin a from indexs $1:T$ to n with indexs $1+\tau : T+\tau$. If the index of $\hat{x}_i$ exceeds the length of the untrimmed video, we loop the index from the beginning.

Intuitively, the choose of τ determines the similarity between x_i and $\hat{x}_i$. When τ approaches zero, the generated $\hat{x}$ looks very similar with x. To differentiate the n and a apart, the encoder network must focus on global rather local statistics.

Objective Function

We employ two objective functions to optimize the model. One is intra-video triplet learning, which generates negative sample by itself. The other one is the contrastive learning, which takes in other video as negative samples. While contrastive loss has been widely and successfully used in self-supervised methods, we verify that our methods does not rely on such loss design and generalize well with triplet loss.

Intra-video Triplet Learning. We first optimize the network with the following objective function in the form of triplet loss.

$$\mathcal{L}_t = \sum_{i=1}^{N} max\{d(z_{a_i}, z_{p_i}) - d(z_{a_i}, z_{n_i}) + margin, 0\} \tag{5}$$

where $d(z_a, z_p) = ||z_a - z_p||_2$, $d(z_a, z_n) = ||z_a - z_n||_2$ and $margin$ is a hyperparameter to restrain the distance between $d(z_a, z_p)$ and $d(z_a, z_n)$.

Contrastive Learning. Contrastive learning, *e.g.*, InfoNCE (Hjelm et al. 2018), learns to obtain representations by maximizing similarity of similar pairs over dissimilar pairs. Given a query q, a corresponding positive sample k^+ and other negatives $\{k^-\}$, InfoNCE defines the contrastive loss as follows:

$$\mathcal{L} = -log\frac{exp(q \cdot k_+/\tau)}{exp(q \cdot k_+/\tau) + \sum_{k^-} exp(q \cdot k^-/\tau)} \tag{6}$$

where τ is a temperature hyperparameter that is used to scale the distribution of distance. For simplicity, we set $\tau = 1$ by

default. Then we introduce (z_a, z_p, z_n) into InfoNCE and the final objective function is as follows:

$$\mathcal{L}_c = -log \sum_{i=1}^{N} \frac{exp(z_{a_i} \cdot z_{p_i})}{sim(z_{a_i}, z_{p_i}, z_{n_i}) + \sum_{j=0}^{K} exp(z_{a_i} \cdot z_{a_j})} \tag{7}$$

where $sim(z_{a_i}, z_{p_i}, z_{n_i}) = exp(z_{a_i} \cdot z_{p_i}) + exp(z_{a_i} \cdot z_{n_i})$ and K is the number other samples. We use a memory bank with size K to save features of a. Compared to the intra-video triplet, inter-video samples are also used as negative. We adopt MoCo (He et al. 2020) as the basic framework of contrastive representation learning for its efficacy and efficiency.

It can be seen from the equation 5 and equation 7 that the learning of embedding space depends on the quality of the generated positive and negative samples. By applying DSM, we expect the prediction not to be determined by the spatial context and take more motion pattern into account.

Experiments

Implementation Details

Datasets. All the experiments are conducted on three video classification benchmarks, UCF101, HMDB51 and Kinetics (Kay et al. 2017). UCF101 consists of 13,320 manually labeled videos in 101 action categories and HMDB51 comprises 6,766 manually labeled clips in 51 categories, both of which are divided into three train/test splits. Kinetics is a large scale action recognition dataset that contains 246k/20k train/val video clips of 400 classes.

Networks. We use C3D(Tran et al. 2015), I3D(Carreira and Zisserman 2017) and 3D ResNet-34(Hara, Kataoka, and Satoh 2018) as base encoders followed by a global average pooling layer and a fully connected layer to project the representations into a 128-dimensional latent space.

Default Settings. All the experiments are conducted on 16 Tesla V100 GPUs with a batch size of 128. For each video clip, we uniformly sample 16 frames with a temporal stride of 4 and then resize the sampled clip to $16 \times 3 \times 224 \times 224$. The margin of triplet loss is set to 0.5 and the smoothing coefficient m of momentum encoder in contrastive representation learning is set to 0.99 following MoCo(He et al. 2020). The memory bank size K is set to 6536.The boundary of temporal shift operation α_1 is 2 and α_2 is 20 for UCF101. Since the average length of Kinetics is larger than UCF101, α_1 is set to 4 and α_2 is set to 30.

Pre-training Settings. We pre-trained the network for 200 epochs and adopt SGD as our optimizer with a momentum of 0.9 and a weight decay of 5e-4. The learning rate is initialized as 0.003 and decreases to 1/10 at the 80, 120 and 160 epoch.

Fine-tuning Settings. After pre-training, we transfered the weights of base encoder network to two downstream tasks, action recognition and video retrieval. We fine-tuned on each dataset for 45 epochs. The learning rate is initialized as 0.1 and multiplied by 0.1 every 10 epochs.

Evaluation Settings. For action recognition, follow common practice (Wang et al. 2019b), the final result of a video is the average of the results of 10 clips that uniformly sampled from it during testing time.

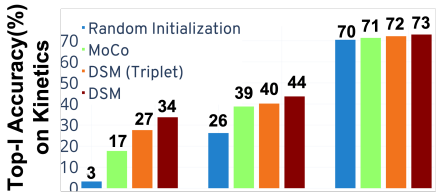

Figure 4: Top-1 accuracy with 1%, 10% and 100% Kinetics labels on Kinetics. Our method brings prominent improvement especially with a small amount of labeled data. Compared to MoCo baseline, DSM brings an increase of 16% with 1% labels, an increase of 4.8% with 10% labels. When using a large amount of labeled data (100% Kinetics labels), DSM can still increase the accuracy of MoCo baseline from 71.4% to 73%.

Action Recognition

Transfer Learning. We fine-tuned the whole model on UCF101 and HMDB51 with all labels, and the results are shown in Table 1. We also compare the results of the I3D network pretrained with all labels of ImageNet and Kinetics in a supervised manner. It can be seen from the experimental results that the encoder pre-trained with DSM can significantly surpass the random initialized counterpart across various network architectures and benchmarks. Compared to SpeedNet, one of the state-of-the-art methods, DSM brings 8.1% and 8.8% improvement on UCF101 and HMDB51 respectively. We can also observe that our approach surpasses all the other self-supervised methods on both UCF101 and HMDB51 datasets using the same backbone.

Test with Limited Labeled Data. Following SimCLR (Chen et al. 2020), we sampled 1% and 10% labeled data from Kinetics in a class-balanced way (∼6 and ∼57 videos per class) and fine-tuned the whole model with the sampled data. Figure 4 exhibits the comparison results of our method with the MoCo baseline at 1%, 10% and 100% labeled data on the validation set of Kinetics. We also report the results of the random initialized model for reference. All the experiments in this part use I3D as backbone. It can be seen from the figure that DSM significantly exceeds the MoCo baseline at all the volumes of labeled data. At the same time, DSM brings more prominent improvement with a small amount of labeled data. Specifically, with 1% labeled data, the accuracy increases from 17.7% to 33.7%, and with 10% labeled data, the accuracy increases from 38.8% to 43.6%. It is worth noting that the results of DSM(Triplet) also excel the MoCo baseline at each volume of the labeled data, and are quite close to DSM. This further proves that the intra-video positive and negative sample construction strategy is indeed effective. Moreover, when the amount of labeled data is very large, that is, using 100% Kinetics labeled data, DSM can still increase the accuracy of MoCo baseline from 71.4% to 73%, which indicates that our method is well generalized.

Method	Year	Resolution	Pretrained	Architecture	UCF101	HMDB51
Supervised						
Random init	-	224×224	-	I3D	47.9	29.6
ImageNet inflated	-	224×224	ImageNet	I3D	67.1	42.5
Kinetics supervised	-	224×224	Kinetics	I3D	96.8	74.5
Self-supervised						
Puzzle (Kim, Cho, and Kweon 2019)	AAAI'19	112×112	UCF101	C3D	65.0	31.3
VCP (Luo et al. 2020)	AAAI'20	112×112	UCF101	C3D	68.5	32.5
PRP (Yao et al. 2020)	CVPR'20	112×112	UCF101	C3D	69.1	34.5
MoCo (He et al. 2020) ◊	CVPR'20	112×112	UCF101	C3D	60.5	27.2
DSM (Triplet)	-	112×112	UCF101	C3D	68.4	38.2
DSM	-	112×112	UCF101	C3D	70.3	40.5
Clip Order (Xu et al. 2019)	CVPR'19	112×112	Kinetics	R(2+1)D	72.4	30.9
DPC (Han, Xie, and Zisserman 2019)	ICCVW'19	224×224	Kinetics	3D-ResNet34	75.7	35.7
AoT (Wei et al. 2018)	CVPR'18	224×224	Kinetics	T-CAM	79.4	-
SpeedNet (Benaim et al. 2020)	CVPR'20	224×224	Kinetics	I3D	66.7	43.7
MoCo (He et al. 2020) ◊	CVPR'20	224×224	Kinetics	I3D	62.3	36.5
DSM (Triplet)	-	224×224	Kinetics	I3D	70.7	48.5
DSM (Triplet)	-	224×224	Kinetics	3D-ResNet34	76.9	50.3
DSM	-	224×224	Kinetics	I3D	74.8	52.5
DSM	-	224×224	Kinetics	3D-ResNet34	**78.2**	**52.8**

Table 1: The top-1 accuracy (%) of our method compared with previous approaches on the UCF101 and HMDB51 dataset. DMS(Triplet) is optimized with triplet loss. All the accuracy is averaged over three splits and ◊ means a custom implementation.

Method	**Net**	**1**	**5**	**10**	**20**	**50**
Jigsaw (Noroozi and Favaro 2016b)	CFN	19.7	28.5	33.5	40.0	49.4
OPN (Lee et al. 2017)	OPN	**19.9**	28.7	34.0	40.6	51.6
Clip Order (Xu et al. 2019)	C3D	12.5	29.0	39.0	50.6	66.9
Clip Order (Xu et al. 2019)	R3D	14.1	30.3	40.0	51.1	66.5
SpeedNet(Benaim et al. 2020)	S3D-G	13.0	28.1	37.5	49.5	65.0
DSM	C3D	16.8	33.4	43.4	54.6	70.7
DSM	I3D	17.4	**35.2**	**45.3**	**57.8**	**74.0**

Table 2: Recall-at-topK (%). Accuracy under different K values on UCF101.

Method	**Net**	**1**	**5**	**10**	**20**	**50**
Clip Order (Xu et al. 2019)	C3D	7.4	22.6	34.4	48.5	70.1
Clip Order (Xu et al. 2019)	R3D	7.6	22.9	34.4	48.8	68.9
VCP (Luo et al. 2020)	C3D	7.8	23.8	35.3	49.3	71.6
DSM	C3D	**8.2**	**25.9**	**38.1**	52.0	75.0
DSM	I3D	7.6	23.3	36.5	**52.5**	**76.0**

Table 3: Recall-at-topK (%). Accuracy under different K values on HMDB51.

Video Retrieval

In this section, we evaluated DSM on video retrieval task. Following Clip Order and SpeedNet, the network is fixed as a feature extractor after pre-training with DSM on the split 1 of UCF101. Then the videos from both UCF101 and HMDB51 are divided into clips in units of 16 frames. All the clips in the training set constitute a *Gallery*, and each clip in the test set is used as a *query* to retrieve the most similar clip in the *Gallery* with cosine distance. If the category of the query appears in the K-nearest neighbors is retrieved, then it is considered as a hit. It should be noted that in order to keep the scale of representations generated by each 3D architecture consistent, we replaced the original global average pooling with an adaptive max pooling, yielding representations with a fixed scale of $1024 \times 2 \times 7 \times 7$. We show the accuracy when $K = 1, 5, 10, 20, 50$ and compare with other self-supervised methods on UCF101 and HMDB51 in Table 2 and Table 3 respectively. It can be seen that when using the same backbone C3D, DSM surpasses the mainstream method Clip Order on the UCF101, and surpasses Clip Order and VCP on the HMDB51, which proves that the representations extracted by DSM are more discriminative.

Ablation Study

In this section, we explore the effectiveness of each component in the DSM. Results are shown in Table 4, from which we can conclude that all of these components lead to better results, and in the way of generating negative samples, both scaling optical-flow and temporal-shift are effective and the combination of the two can bring further gains.

Analysis

Visualizing salient regions. In order to analyse which space-time regions our model focus on, we visualize the en-

Positive	Negative	UCF101	HMDB51
MoCo baseline			
-	-	62.3	36.5
Our Method			
S-warping	-	66.4 (+4.1)	41.3 (+4.8)
-	M-disturb	67.7 (+5.4)	44.0 (+7.5)
S-warping	Scaling-Of	69.4 (+7.1)	47.5 (+11.0)
S-warping	T-Shift	71.2 (+8.9)	50.4 (+13.9)
S-warping	M-disturb	74.8 (+12.5)	52.5 (+16.0)

Table 4: Ablation study of each component on the UCF101 and HMDB51. All the methods are pre-trained on Kinetics with I3D backbone. S-warpping, M-disturb, Scaling-Of and T-Shift denote spatial warpping, motional disturbance, scaling optical-flow and temporal-shift respectively.

ergy of the extracted representations with CAM(Zhou et al. 2016). For comparison, we pre-train two models using I3D as backbone on the UCF101 under two settings: *i.* fully supervised, *ii.* self-supervised using DSM. We select some videos with obvious motion from HMDB51 and use a sliding window to generate multiple clips for each video, and then visualize the corresponding activation maps of these clips in Figure 5. Specifically, assuming that each video has L frames, a sliding window with a scale of 16 and a stride size of 4 slides on the temporal dimen sion, generating $(L-16)/4$ clips fors each video. All clips of each video are input to the above two models and the extracted feature representations of the last 3D layer before global average pooling is of the shape of $1024 \times T \times N \times N$, where T and N are the scale of the temporal and spatial dimension. Afterwards, we average over all channels to compress these features into the shape of $T \times N \times N$, then upsample and mask these heatmaps to the original videos. Visualization results under setting *i* are displayed in the first row, and those under setting *ii* are shown in the second row. It can be observed that the supervised approach is severely affected by the scene bias and falsely focus on the static background. On the contrary, DSM suffer less from scene bias and correctly focus on moving objects. Moreover, for setting *ii*, we average and normalize the feature of each clip into a scalar, which is recorded as response value, then we plot the curve of all clip response values over time. We find the curve has a low value when there is a inconspicuous movement, such as a clip that is about to end an action, and the alternate phase of a cyclic action, which is consistent with our original intention to enhance the temporal sensitivity of the model.

Adversarial examples. Since each action is carried out by a subject, a natural question comes out: does the model only learn to focus on the human body or it really learns to focus on the movement areas? To verify this question, we generate some adversarial samples in Figure 6. First, using a static video (copy one single frame multiple times) as input, the model shows random response. Then we paste another human body or introduce a static frame as noise, our method still correctly focus on movement area. The experiments in this part prove that the feature representation extracted by

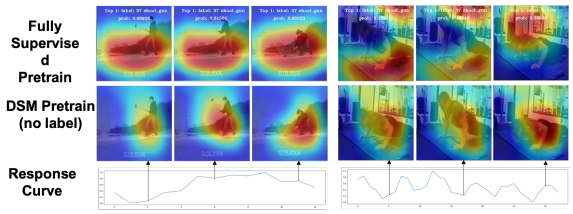

Figure 5: Which space-time regions does the trained model focus on? Notice that these action categories did not appear during the training time. The model trained with labels focus more on background and shows poor generation on new classes while the model pre-trained using DSM without any label pay more attention to the moving area. In addition, our method has a high response value for clips with strong motion information, and vice versa.

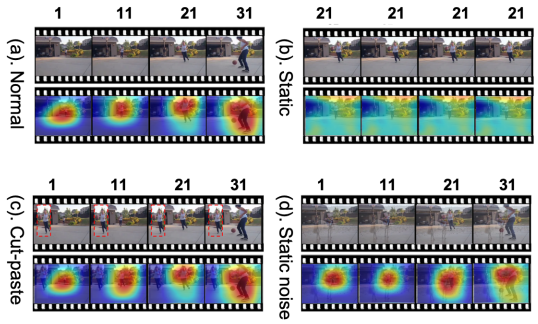

Figure 6: Does the model really learn to focus on motion regions? The video *dribble* is selected from HMDB51 and the model is pre-trained with DSM on Kinetics. It can be concluded that: *i*. When the input is a static video, DSM doesn't know where to focus on. *ii*. When pasting another static human body, DSM still focus on the real movement area, which indicates that our method is even robust to human body distraction. *iii*. Using a static frame as noise has no effect on our model.

DSM have a fully understanding of space-time.

Conclusion

Due to the ubiquitously existing scene and motion coupling problem in the current video dataset, there are many actions that can be recognized simply from a static background. However, only focusing on the background does not generalize the model well in the open scene and may dwarft the temporal modeling. This paper presents DSM, a novel self-supervised method to overcome the influence of implicit bias over scenes. Combined with the metric learning, our method has a high tolerance towards the scene variants. We evaluate DSM both quantitatively and qualitatively. On the two popular benchmarks UCF101 and HMDB51, the proposed methods improves the state-of-the-art notably. And by visualizing the model focus map, our method correct focus on the motion instead of the unrelated area. We extend our method to retrieval and detection and good results are also achieved.

References

Benaim, S.; Ephrat, A.; Lang, O.; Mosseri, I.; Freeman, W. T.; Rubinstein, M.; Irani, M.; and Dekel, T. 2020. SpeedNet: Learning the Speediness in Videos. In *CVPR*, 9922–9931.

Carreira, J.; and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 6299–6308.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709* .

Choi, J.; Gao, C.; Messou, J. C.; and Huang, J.-B. 2019. Why Can't I Dance in the Mall? Learning to Mitigate Scene Bias in Action Recognition. In *NeurIPS*, 853–865.

Feichtenhofer, C.; Fan, H.; Malik, J.; and He, K. 2019. Slowfast networks for video recognition. In *ICCV*, 6202–6211.

Gan, C.; Gong, B.; Liu, K.; Su, H.; and Guibas, L. J. 2018. Geometry guided convolutional neural networks for self-supervised video representation learning. In *CVPR*, 5589–5597.

Gidaris, S.; Singh, P.; and Komodakis, N. 2018. Unsupervised Representation Learning by Predicting Image Rotations. In *ICLR*. URL https://openreview.net/forum?id=S1v4N2l0-.

Girdhar, R.; and Ramanan, D. 2020. CATER: A diagnostic dataset for Compositional Actions and TEmporal Reasoning. In *ICLR*.

Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality reduction by learning an invariant mapping. In *CVPR*, volume 2, 1735–1742. IEEE.

Han, T.; Xie, W.; and Zisserman, A. 2019. Video representation learning by dense predictive coding. In *ICCVW*, 0–0.

Hara, K.; Kataoka, H.; and Satoh, Y. 2018. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, 6546–6555.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 9729–9738.

Hjelm, R. D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; and Bengio, Y. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670* .

Horn, B. K.; and Schunck, B. G. 1981. Determining optical flow. In *Techniques and Applications of Image Understanding*, volume 281, 319–331. International Society for Optics and Photonics.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; et al. 2015. Spatial transformer networks. In *NeurIPS*, 2017–2025.

Jenni, S.; Jin, H.; and Favaro, P. 2020. Steering Self-Supervised Feature Learning Beyond Local Pixel Statistics. In *CVPR*, 6408–6417.

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950* .

Kim, D.; Cho, D.; and Kweon, I. S. 2019. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*, volume 33, 8545–8552.

Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; and Serre, T. 2011. HMDB: a large video database for human motion recognition. In *ICCV*, 2556–2563. IEEE.

Lee, H.-Y.; Huang, J.-B.; Singh, M.; and Yang, M.-H. 2017. Unsupervised Representation Learning by Sorting Sequences. In *ICCV*, 667–676.

Li, Y.; Li, Y.; and Vasconcelos, N. 2018. RESOUND: Towards Action Recognition without Representation Bias. In *ECCV*.

Luo, D.; Liu, C.; Zhou, Y.; Yang, D.; Ma, C.; Ye, Q.; and Wang, W. 2020. Video cloze procedure for self-supervised spatio-temporal learning. *AAAI* .

Misra, I.; Zitnick, C. L.; and Hebert, M. 2016. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 527–544. Springer.

Noroozi, M.; and Favaro, P. 2016a. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 69–84. Springer.

Noroozi, M.; and Favaro, P. 2016b. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In Leibe, B.; Matas, J.; Sebe, N.; and Welling, M., eds., *ECCV*, 69–84. Cham: Springer International Publishing. ISBN 978-3-319-46466-4.

Qian, R.; Meng, T.; Gong, B.; Yang, M.-H.; Wang, H.; Belongie, S.; and Cui, Y. 2020. Spatiotemporal Contrastive Video Representation Learning. *arXiv preprint arXiv:2008.03800* .

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 815–823.

Sermanet, P.; Lynch, C.; Chebotar, Y.; Hsu, J.; Jang, E.; Schaal, S.; Levine, S.; and Brain, G. 2018. Time-contrastive networks: Self-supervised learning from video. In *ICRA*, 1134–1141. IEEE.

Shi, B.; Wang, X.; Lyu, P.; Yao, C.; and Bai, X. 2016. Robust scene text recognition with automatic rectification. In *CVPR*, 4168–4176.

Simonyan, K.; and Zisserman, A. 2014. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 568–576.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* .

Tao, L.; Wang, X.; and Yamasaki, T. 2020. Self-supervised Video Representation Learning Using Inter-intra Contrastive Framework. *arXiv preprint arXiv:2008.02531* .

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 4489–4497.

Wang, J.; Jiao, J.; Bao, L.; He, S.; Liu, Y.; and Liu, W. 2019a. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *CVPR*, 4006–4015.

Wang, J.; Jiao, J.; Bao, L.; He, S.; Liu, Y.; and Liu, W. 2019b. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *CVPR*, 4006–4015.

Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Van Gool, L. 2018. Temporal segment networks for action recognition in videos. *TPAMI* 41(11): 2740–2755.

Wang, Y.; and Hoai, M. 2018. Pulling Actions out of Context: Explicit Separation for Effective Combination. In *CVPR*.

Wei, D.; Lim, J. J.; Zisserman, A.; and Freeman, W. T. 2018. Learning and using the arrow of time. In *CVPR*.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 3733–3742.

Xu, D.; Xiao, J.; Zhao, Z.; Shao, J.; Xie, D.; and Zhuang, Y. 2019. Self-Supervised Spatiotemporal Learning via Video Clip Order Prediction. In *CVPR*, 10334–10343.

Yao, Y.; Liu, C.; Luo, D.; Zhou, Y.; and Ye, Q. 2020. Video Playback Rate Perception for Self-Supervised Spatio-Temporal Representation Learning. In *CVPR*, 6548–6557.

Zhao, Y.; Xiong, Y.; and Lin, D. 2018. Recognize actions by disentangling components of dynamics. In *CVPR*, 6566–6575.

Zhou, B.; Andonian, A.; Oliva, A.; and Torralba, A. 2018. Temporal relational reasoning in videos. In *ECCV*, 803–818.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *CVPR*, 2921–2929.