Self-supervised Spatiotemporal Feature Learning by Video Geometric Transformations

Longlong Jing
The Graduate Center
The City University of New York
ljing@gradcenter.cuny.edu

Yingli Tian
The City College and Graduate Center
The City University of New York
ytian@ccny.cuny.edu

Abstract

To alleviate the expensive cost of data collection and annotation, many self-supervised learning methods were proposed to learn image representations without human-labeled annotations. However, self-supervised learning for video representations is not yet well-addressed. In this paper, we propose a novel 3DConvNet-based fully self-supervised framework to learn spatiotemporal video features without using any human-labeled annotations. First, a set of pre-designed geometric transformations (e.g. rotating 0°*,* 90°*,* 180°*, and* 270°*) are applied to each video. Then a pretext task can be defined as "recognizing the pre-designed geometric transformations." Therefore, the spatiotemporal video features can be learned in the process of accomplishing this pretext task without using human-labeled annotations. The learned spatiotemporal video representations can further be employed as pre-trained features for different video-related applications. The proposed geometric transformations (e.g. rotations) are proved to be effective to learn representative spatiotemporal features in our 3DConvNet-based fully self-supervised framework. With the pre-trained spatiotemporal features from two large video datasets, the performance of action recognition is significantly boosted up by* 20.4% *on UCF101 dataset and* 16.7% *on HMDB51 dataset respectively compared to that from the model trained from scratch. Furthermore, our framework outperforms the state-of-the-arts of fully self-supervised methods on both UCF101 and HMDB51 datasets and achieves* 62.9% *and* 33.7% *accuracy respectively.*

1. Introduction

With more videos flourishing on the internet, video-based applications such as action recognition, action localization, and video captioning [4, 5, 33, 38, 39, 47] have drawn more and more attention. Due to the powerful ability to learn rich hierarchy visual features, both 2DConvNets and 3DConvNets have been widely applied to video action recognition [4, 5, 11, 13, 33, 38].

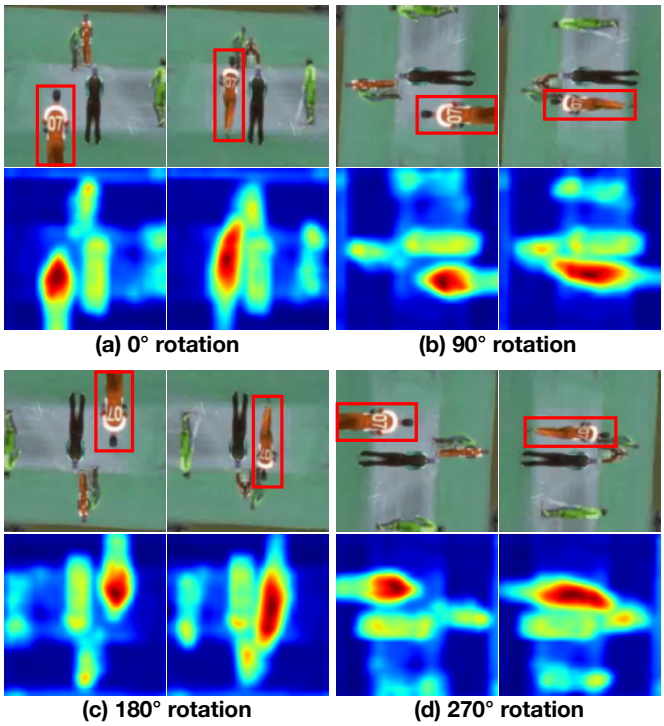

Figure 1. Video frames and their corresponding attention maps generated by our proposed 3DConvNet-based fully self-supervised model at each rotation. Note that both spatial (e.g. locations and shapes of different persons) and temporal features (e.g. motions and location changes of persons) are effectively captured. The hottest areas in attention maps (corresponding to the bounding boxes in images) indicate the person with the most significant motion. The attention map is computed by averaging the activations of the first convolution layer in each grid which reflects the importance of that grid.

However, training of deep ConvNet generally requires a large-scale dataset with expensive and time-consuming human-labeled annotations. To make the training of very deep CNN feasible, larger and larger video datasets are col-

lected and labeled. For example, the Sport-1M dataset contains about 1.1 million videos which have been annotated with 487 categories [15]. The Kinetics dataset consists of approximately $500,000$ videos belong to 600 human actions [16].

To mitigate the cost of large-scale dataset collection and annotation, many self-supervised methods have been proposed to learn image features with deep convolution neural networks (ConvNets) without using human-labeled annotations [6, 8, 18, 23, 27, 30, 31, 34]. In these self-supervised learning methods, a pretext task is usually designed, and visual features that represent image semantic context information are learned by CNNs through the process of accomplishing the pretext task [3, 6, 8, 18, 23, 27, 30, 31, 34]. For example, in the pretext task of image inpainting, Pathak *et al.* designed a self-supervised 2DConvNet to predict the missing regions in an image by learning the concept and the structure of the image [31]. These self-supervised image representation learning methods are proved to be very effective and the gap of the performance between supervised and self-supervised learning is getting smaller. However, almost all of those methods focus on learning image representations by 2DConvNets. Fully self-supervised learning for video representations is not yet well-addressed. Although a few work attempted to utilize video temporal orders, they only focused on the image representation learning without learning the spatiotemporal video features [44, 23].

Due to the powerful ability to simultaneously capture both spatial and temporal representations, 3DConvNets showed a great potential for video understanding tasks [4, 5, 11, 14, 38]. However, millions of training videos are needed for 3DConvNets to obtain a good performance [11, 38]. The pre-trained model on large-scale datasets [15, 16] generally can boost up the testing performance on small datasets compared to directly train from scratch [11, 38]. Few work has been done about fully self-supervised learning the spatial-temporal representations with 3DConvNets [41, 42]. Inspired by image-based self-supervised learning methods [9, 31, 30, 27, 23], in this paper, we propose a very effective 3DConvNet-based fully self-supervised learning framework to learn the spatiotemporal representations from video geometric transformations. For simplicity, we refer our fully self-supervised model as 3DRotNet in the following sections.

In order to learn spatiotemporal representation from videos without using human-labeled annotations, we first apply a set of geometric transformations (e.g. $0°$, $90°$, $180°$, and $270°$) to videos as shown in Fig. 1. Then a pretext task can be defined as "recognizing the set of geometric transformations." The 3DRotNet is trained for the pretext task to recognize how many degrees each input video is rotated. During the process of recognizing the rotation geometric transformations of input videos, 3DRotNet is able to learn the video representations including spatial appearance features in image frames (e.g. location, shape, and color of objects), as well as temporal features during frames (e.g. motion of objects). Fig. 1 illustrates the video frames and their corresponding attention maps generated by our 3DRotNet at each rotation. It demonstrates the effectiveness of our proposed framework to capture spatiotemporal video features. In order to quantitatively evaluate the quality of the learned features by our models, we further transfer the learned features to human action recognition task on two datasets and obtained very good performance, which demonstrate that geometric transformations (e.g. rotation) are effective and efficient to learn representative spatiotemporal features in the proposed 3DConvNet-based fully self-supervised framework. In summary, our main contributions are:

- We propose a novel and effective 3DConvNet-based fully self-supervised framework to learn spatiotemporal features in videos.
- By only using video geometric transformations without any human-labeled annotations, both spatial and temporal video features can be captured.
- The video features learned in an unsupervised manner can be served as pre-trained models to be transferred to other video processing tasks when only small datasets are available. With the pre-trained spatiotemporal features from *Moment in Time* and *Kinetic* datasets, the performance of action recognition is significantly boosted up by 20.4% on the UCF101 dataset and 16.7% on HMDB51 dataset respectively compared to that from the models trained from scratch.
- Compared to the state-of-the-art of fully self-supervised methods, the proposed model outperforms them on both UCF101 and HMDB51 datasets.

2. Related Work

Recently, many self-supervised learning methods were proposed to learn image representations [6, 7, 23, 26, 31, 34]. Based on the pretext tasks, these methods fall into two categories: texture-based methods which mainly utilize the texture information of images as the supervision information such as the boundary of the objects [22, 34], the context of images [18, 31], and the similarity of two patches from an image [6, 12, 27, 28, 45], and temporal-based methods which mainly utilize the temporal connection between consecutive frames from videos as the supervision information [8, 19, 23, 30, 37, 40].

Self-supervised learning from images: The similarity between two patches from the same image is often used as a supervision signal for the self-supervised image feature

learning [6, 12, 21, 27]. Noroozi and Favaro proposed an approach to learn visual representations by solving Jiasaw puzzles with 9 patches from same image [27]. Doersch *et al.* proposed to learn visual features by predicting the relative positions of two patches from same image [6]. Li *et al.* proposed to mine the positive and negative image pairs with graph constraints in the feature space and the mined pairs are used to train the network to learn visual features [21]. Caron *et al.* proposed the DeepCluster to iteratively train the network with categories that are generated by clustering [3]. The context information of images such as structure [31], color [18] and relations of objects is another type of supervision signal for self-supervised image feature learning. Gidaris *et al.* proposed to learn visual features by training a ConvNet to recognize 2D image rotations which is proved to be a powerful supervision signal for image feature learning [9]. Larsson *et al.* proposed to use the image colorization as the pretext to learn semantic features of images [18]. Zhang *et al.* proposed the split-brain autoencoder to predict a subset of image channels from other channels [49]. Ren and Lee proposed to learn image features from synthetic images generated by a game engine based on a generative adversarial network [34]. All these methods focus on the image representation learning.

Self-supervised learning from videos: Although there are some work about self-supervised learning from videos, most of them still employed 2DConvNets to learn image representations by using temporal information in videos as the supervision information. Pathak *et al.* proposed to train a 2DConvNet to segment moving objects that unsupervised segmented from videos [30]. Misra *et al.* proposed to train a 2DConvNet to verify whether a sequence of frames is in a correct temporal order [23]. Wang and Gupta proposed a Siamese-triplet network with a ranking loss to train a 2DConvNet with the patches from a video sequence [44]. Fernando *et al.* proposed to learn the video representation by odd-one-out networks to identify the odd element from a set of related elements with a 2DConvNet [8]. Lee *et al.* proposed to take shuffled frame sequence as input to a 2DConvNet to sort the sequences [19]. In addition, LSTM can also be used to learn the visual features from videos especially to model the temporal information among frames [37, 40].

3DCovnNets have been widely used to simultaneously model both spatial and temporal information in videos [4, 11, 13, 33, 38], however, only Vondrick and his colleagues recently attempt to apply it for self-supervised learning [41, 42]. Vondrick *et al.* proposed a method by employing Generative Adversarial Network to generate videos with a 3DCovnNet without human-annotated labels [41]. They also proposed to learn video features by colorizing videos with 3DConvNet [42]. Compared to image self-supervised learning, the spatiotemporal feature learning with 3DConvNet is lacking of study. In this paper, we propose a novel 3DConvNet-based fully self-supervised framework, called 3DRotNet, to learn spatiotemporal video features from unlabeled videos and further transfer the learned video features for action recognition task. Our self-supervised model outperforms the state-of-the-arts on UCF101 and HMDB51 datasets among existing fully self-supervised methods.

3. The Proposed Method

3.1. Model Parametrization

In this paper, we employ 3DConvNets $F(X, \theta)$ to learn spatiotemporal features from a set of pre-applied videos geometric transformations $G(y)$, while X denotes the input video, θ represents the parameters of the network, and y indicates the parameters of the geometric transformations.

This schema can be implemented in two ways: regression or classification. For the regression task, the network predicts the parameters of the geometric transformations that are applied to videos as a continuous variable. For the classification task, a set of discrete geometric transformations are pre-defined, then the network is trained to recognize the types of geometric transformations that are applied to videos. We will describe each implementation respectively.

For a given transformation $G(y)$, the schema of the regression implementation can be formulated as:

$$loss(X_i, \theta) = (F(g(X_i|y)|\theta) - y)^2, \tag{1}$$

while X_i is the input video, y is the parameter of the geometric transformation, and $g(X_i|y)$ is the video after applying the geometric transformation. The network $F(X, \theta)$ is trained to predict each type of the geometric transformation respectively, while usually L1 loss or L2 loss is computed as the regression loss to optimize the network.

When formulate the problem as a classification task, a set of K discrete geometric transformations $G = \{g(.|y)\}_{y=1}^{K}$ is defined while $g(.|y)$ is the operator that is applied to a video X with label y that yield output video $X^y = g(X|y)$. In this case, the network $F(.)$ is optimized by minimizing the cross entropy loss between the predicted probability distribution over K categories and the geometric transformation type y. The loss function is:

$$loss(X_i, \theta) = -\frac{1}{K}\sum_{y=1}^{K} log(F^y(g(X_i|y)|\theta)), \tag{2}$$

while X_i is the input video of size $t \times c \times h \times w$ (t is the total number of frames in each video, c is the number of channels, h is the frame height, and w is the frame width), $g(X_i|y)$ is the operator that transform the input video clip X_i, and θ is the parameter of the network.

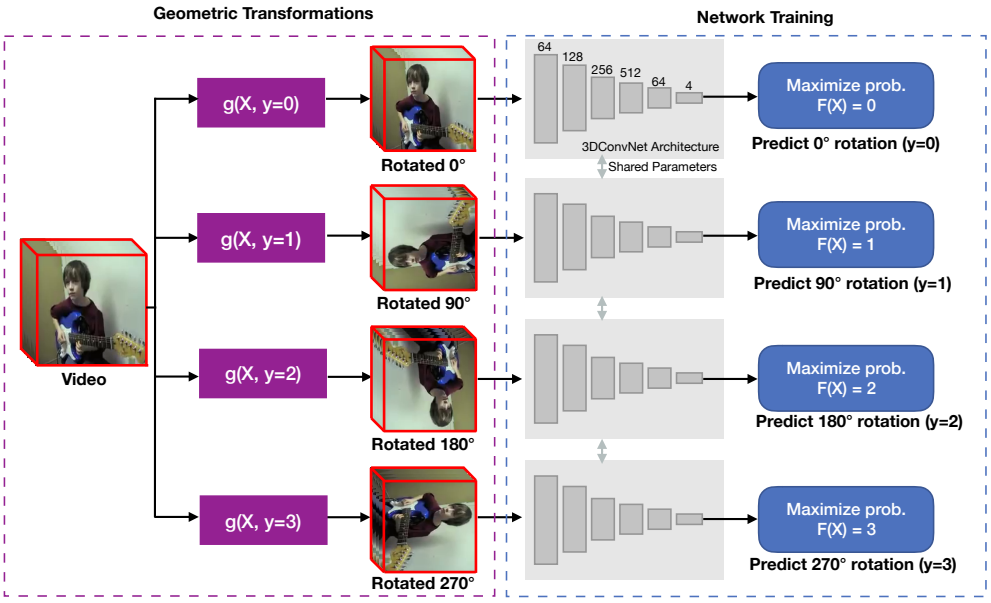

Figure 2. The pipeline of the proposed self-supervised spatiotemporal representation learning. Each video is rotated with four different degrees $(0, 90, 180, 270)$, and a 3DConvNet is trained to recognize the geometric transformations that applied to input videos.

In both scenarios, given a set of N training videos $D = \{X_i\}_{i=0}^{N}$, the training loss function is defined as:

$$loss(D) = \min_{\theta} \frac{1}{N} \sum_{i=1}^{N} loss(X_i, \theta). \tag{3}$$

Generally, classification loss performs better than regression loss on many tasks such as image colorization [48], saliency prediction [29], and depth estimation [20]. Therefore, we choose the classification loss for our self-supervised model.

3.2. Geometric transformation Design

Different kinds of image transformations are designed as supervision information to train 2DConvNets for image representation learning including image colorization [18], image rotation [9], and image denoise [1]. In this paper, we propose to use video rotation as the supervision signal to learn video features. Specifically, 3DConvNets are trained to model both the spatial and temporal features which are representative for the semantic context of videos. Inspired by [9, 18], we formulate the problem as a classification task in which the network is to recognize K types of discrete rotations that are applied to videos.

Choosing rotations as the geometric transformations for learning video features has the following advantages: (1) The problem is well-defined. Most of the videos in real-world environments are filmed in an upright way that the objects in the videos tend to be upright. (2) Compared to other pretext tasks, the rotation is easy to implement by the flip and transpose operations without adding much time complexity to the network. (3) Unlike other self-supervised learning methods need to take a lot of efforts to avoid the network to learn trivial solutions [6], the rotation operation leaves no artifacts in an image which can ensure the network learn meaningful semantic features through the process of accomplishing this pretext task. Following [9], we design a set of discrete video geometric transformations G as four types of rotations at $0°$, $90°$, $180°$, and $270°$. Therefore, for each video X with the type of rotation y, the output video after the transformation is $G = \{g(X|y)\}_{y=1}^{4}$, where $g(X|y) = Rot(X, (y-1)90)$. The $Rot(X, \omega)$ is the rotation operation that rotates all the frames in a video with ω degrees.

3.3. Proposed Framework

Fig. 2 illustrates the pipeline of the proposed 3DRotNet to learn spatiotemporal video features by predicting rotation geometric transformations. In our implementation, four kinds of rotations $G = \{Rot(X, (y-1)90)\}_{y=1}^{4}$ are applied to each video respectively. Then these four types of videos along with their rotation categories y are used to train the 3DRotNet which predict the probability over all possible rotations for each video. The cross entropy loss is computed between the predicted probability distribution $F(X)$ and the rotation categories y and is minimized to update the weights of the network.

We choose the 18-layer 3DResNet since it has fewer parameters but is capable to learn spatiotemporal features for large-scale datasets [11]. There are five convolution blocks,

while the first one consists of one convolution layer, one batch normalization layer, one Relu layer followed by one a max-pooling layer, and the rest four convolution blocks are 3D residual blocks with skip connection. The number of kernels in each convolution block is illustrated in Fig. 2. After the five convolution blocks, the average pooling is applied to the activation of the fifth convolution block to obtain a 512-Dimension vector. During the rotation prediction training, this 512-Dimension vector is followed by two fully connected layers with the dimensions of 64 and 4 to generate the prediction probability, while in the fine-tuning on action recognition task, this 512-Dimension vector is followed by only one fully connected layer of size equals to the number of action classes in the dataset of action recognition.

Unlike other self-supervised learning methods [23, 27, 30] which usually involve massive data preparation, our framework is very straightforward to implement without massive data preparation. As in [30], masks of moving objects need to be generated in advance, while heavy data augmentation is applied to avoid the network learn trivial solutions [6]. Furthermore, there is no extra efforts needed to avoid trivial solutions since the rotation operations do not generate image artifacts.

4. Evaluation Metrics

To evaluate the quality of the learned image features, previous self-supervised learning methods for image representations usually use the learned image features as a start point and fine-tune on other visual tasks such as image classification, object detection, and semantic segmentation. The performance of the transfer learning on these high-level visual tasks are compared to evaluate the generalization ability of the self-supervised learned features. If the self-supervised learning model can learn representative semantic features, then it can be served as a good start point and leads to better performance on these high-level visual tasks. In addition to the quantitative evaluation, the previous method also analyzes the kernels and features to provide qualitative visualization results [3, 6, 23, 34].

Following the image self-supervised learning evaluation metrics [18, 27, 30, 31], we evaluate the quality of the learned video features with the following evaluation metrics:

1. Qualitatively analyze the kernels of the first convolution layer in the 3DResNet learned with the proposed method and compare the kernels with that of the state-of-the-art supervised models.

2. Analyze the feature maps produced by the proposed models and compare them with that of the state-of-the-art supervised models.

3. Transfer the pre-trained 3DRotNet to video action recognition tasks and compare the performance with the existing self-supervised methods on public benchmarks.

4. Conduct ablation studies to evaluate the impact of the configuration of the geometric transformations on the quality of the features learned by 3DRotNet.

5. Experimental Results

In this section, we conduct extensive experiments to evaluate the effectiveness of the proposed framework and the quality of learned spatiotemporal features for action recognition task from videos.

5.1. Datasets

For the self-supervised video feature learning, the network is trained on videos from two large-scale datasets: Moment in Time [25] and Kinetics [16] without using any of their annotations. During the transfer learning to action recognition task, the network is trained on two action recognition benchmark datasets respectively: UCF101 [36] and HMDB51 [17].

Moment in Time (MT): The MT dataset is a large balanced and diverse dataset for video understanding [25]. The MT dataset consists of around 1 million videos that cover 339 Moment classes, and each video lasts around 3 seconds. The average number of videos for each class is $1,757$ with a median of $2,775$. This data is used to train our self-supervised learning model without using the class labels.

Kinetics: The Kinetics is a large-scale and high-quality video dataset collected from YouTube [16]. The dataset consists of around $500,000$ videos belong to 600 human action classes with at least 600 videos for each class. Each video lasts around 10 seconds and is assigned with a single class label. We downloaded around $480,000$ videos and all of them are used to train our self-supervised model without using the class labels.

UCF101: The UCF101 is a widely used benchmark for action recognition task. It consists of $13,320$ videos that cover 101 human action classes. Videos in this dataset have a spatial resolution of 320×240 pixels.

HMDB51: The HMDB51 is another widely used benchmark for action recognition. It consists of $6,770$ videos belong to 51 actions. Each video has a spatial resolution of 320×240.

5.2. Implementation Details

Self-supervised learning: The videos in Kinetics and MT datasets are evenly down-sampled into 160 and 90 frames respectively and then are re-sized to a spatial resolution at 136×136. During training, 16 consecutive frames are randomly selected from each video as a training clip, and a 112×112 patch is randomly cropped from each frame to form a clip of size 3 channels $\times$ 16 frames $\times 112 \times 112$ pixels. Each video is horizontally flipped with

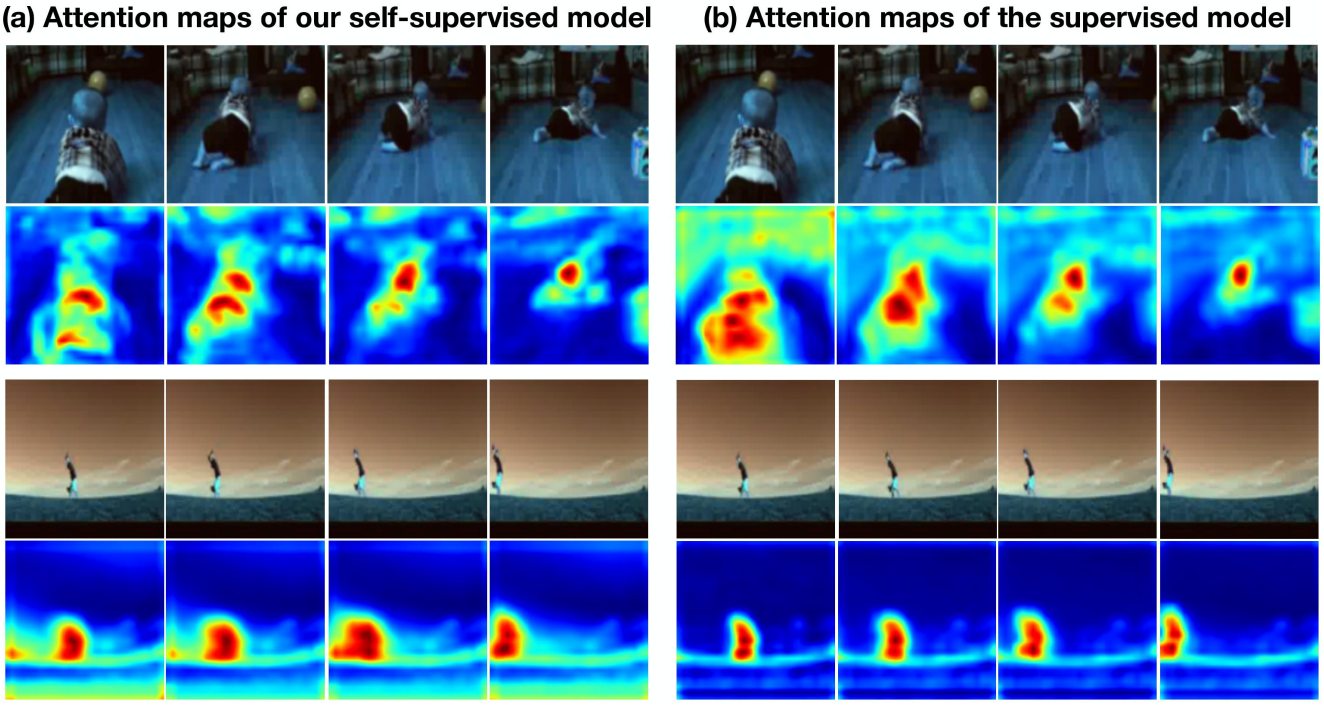

Figure 3. Sampled video frames and their corresponding attention maps generated by our proposed self-supervised 3DRotNet and by supervised model. The attention maps show that our model can capture both spatial and temporal information within videos. Moreover, the proposed self-supervised model can capture the main objects and their motions in a video as the supervised model.

50% probability to augment the dataset. For each video, all the frames are rotated with four different degrees, and the four rotated videos are simultaneously fed to the network. Similar to 3DResNet [11], we also perform the mean subtraction, which means each channel is subtracted with the mean value of all videos from the dataset. The network is optimized by the Stochastic Gradient Descent (SGD) with $10,4000$ iterations and with a batch size of 32. The initial learning rate is set to 0.1 and is multiplied by 0.1 every $2,4000$ iterations.

Transfer learning: To evaluate the quality of the learned features, we fine-tune the learned model to conduct action recognition task on two public datasets: HMDB51 [17] and UCF101 [36]. During training, 16 consecutive frames are randomly selected from a video and resized to a spatial size of 136×136 pixels, then a 112×112 patch is cropped from each frame within the clip to form a tensor of size 3 channels $\times$ 16 frames $\times$ 112 $\times$ 112 pixels. The cross entropy loss is computed and optimized by SGD with 100 epochs. The initial learning rate is set to 0.008 and is multiplied by 0.1 every $4,000$ iterations.

5.3. Can 3DRotNet Recognize Video Rotations?

The hypothesis of our idea is that if a network can recognize video rotations, then it should be able to capture the semantic information in videos which is essential to other visual tasks such as action recognition. Therefore, we first test the performance of recognizing four types of different video rotations ($0°$, $90°$, $180°$, and $270°$). The proposed 3DRotNet is trained on two large-scale video datasets, Kinetics [16] and MT [25], to recognize video rotations. During training, the class labels of the videos in the two datasets are discarded and videos after applied with the four rotations are used to train the 3DRotNet.

Training dataset	UCF101 (%)	HMDB51 (%)
Kinetics	92.79	93.66
MT	93.21	89.88

Table 1. Accuracy of recognizing video rotation on UCF101 and HMDB51 datasets. The 3DRotNet can accomplish this task with a accuracy of more than 89%.

After trained on the two large-scale datasets, the network is tested on the UCF101 and HMDB51 datasets. During testing, all the videos in UCF101 and HMDB51 datasets are first applied with the four types of rotations. Then these rotated videos are fed to the 3DRotNet to predict the rotation degrees. The rotation recognition accuracy is shown in Table 1. The accuracy of video rotation recognition on the two small datasets are higher than 89% which demonstrate that the proposed 3DRotNet is able to capture representative spatial appearance features for videos to recognize their rotations. However, it is still unclear if the 3DRotNet can effectively capture the temporal features.

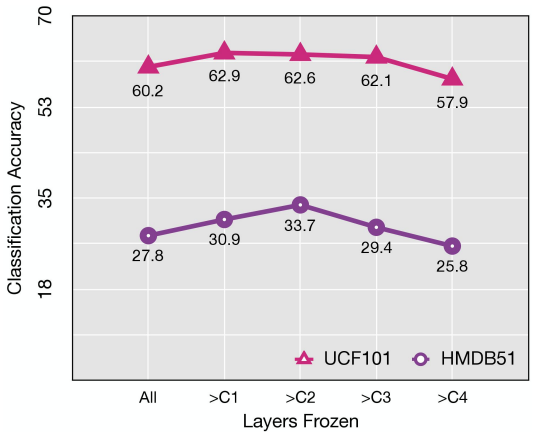

Figure 4. Finetune results on UCF101 and HMDB51 datasets. Cn means the n-th convolution block. >Cn means the blocks before the n-th convolution block are frozen during fine-tune.

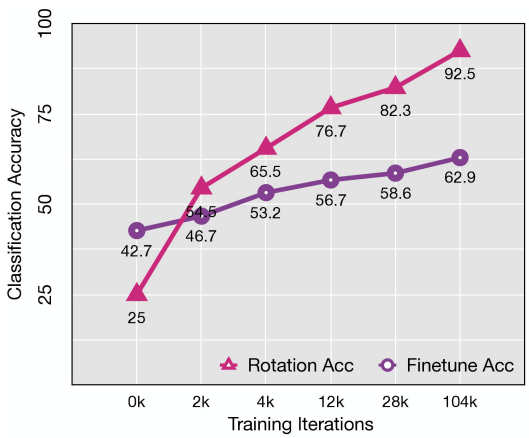

Figure 5. Plot the accuracy of the rotation recognition accuracy and the action recognition accuracy as a function of training iterations. The performance of action recognition increases along with the improvement of the accuracy of rotation recognition.

5.4. Can 3DRotNet Learn Spatiotemporal Video Features?

In order to verify whether the 3DRotNet learned from video rotations can capture both spatial and temporal features from videos such as moving objects, or whether the 3DRotNet may learn trivial solutions such as using the lines in videos to determine their rotations instead of meaningful features, we visualize the attention maps of our self-supervised 3DRotNet by averaging the activations of the first convolution layer in each grid which reflect the importance of that grid.

As shown in Fig. 3, we visualize the attention maps of the proposed self-supervised model and compare with that from the supervised model. Both the proposed 3DRotNet and the supervised 3DConvNet are trained on the Kinetics dataset and tested on the UCF101 dataset. The only difference is that the 3DRotNet trained without the human-annotated category labels. The attention maps show that the 3DRotNet mainly focuses on important objects in the videos and can capture moving objects as the supervised model. As shown in the first two images of the video of baby crawling (the right-bottom video in Fig. 3,) the network can capture the moving baby on the ground. This confirms that our 3DRotNet can capture spatiotemporal information within videos. More attention maps can be found in Figs. 7-10.

5.5. Transfer to Action Recognition Task

In order to evaluate the generalization of the learned video features by our self-supervised models from simple geometric transformations, we further conduct action recognition task on two different datasets (UCF101 and HMDB51) by using the learned video features as a start point and then finetuned on the action recognition datasets. The experimental results on the first split of UCF101 and HMDB51 datasets are shown in Table. 2.

Models	UCF101 (%)	HMDB51 (%)
3DResNet (scratch) [11]	42.5	17.0
Ours (Kinetics)	62.9 **(+20.4)**	33.7 **(+16.7)**
Ours (MT)	62.8 **(+19.2)**	29.6 **(+12.6)**

Table 2. Results of transfer learning of the self-supervised model on the action recognition task on UCF101 and HMDB51 datasets. With the pre-trained spatiotemporal features from the Kinetics dataset, the performance of action recognition is significantly boosted up by 20.4% on UCF101 dataset and 16.7% on HMDB51 dataset respectively compared to that from the model trained from scratch.

As shown in Table. 2, when the 3DResNet is trained from scratch on the two action recognition datasets it only achieves 42.5% on UCF101 and 17.0% on HMDB51 due to over-fitting. However, when fine-tuned our self-supervised models on each dataset with using the learned video features, the performance has a significant improvement of 20.4% (achieves 62.9%) on UCF101 and 16.7% (33.7%) on HMDB51 which proves that the proposed self-supervised learning method is effective and indeed can provide a good start point for the 3DResNets on the small datasets.

Following other self-supervised methods [30], the performance of ConvNets layers frozen with different extent are compared and shown in Fig. 4. The models are pre-trained on Kinetics dataset and then finetuned on HMDB51 and UCF101 datasets. For UCF101 dataset, the network has the best performance when the first convolution block is frozen, and has the worst performance when all the convolution blocks are frozen during training. For HMDB51 dataset, the network has the best performance when the first two convolution blocks are frozen, and has the worst per-

formance when all the convolution blocks are frozen. This probably is because the lower layers learn the general low-level feature, while deeper layers learn the high-level task-specific features. When fine-tuned on the small dataset, the parameters of lower layers need to be preserved and deeper layers need to be further tuned for specific tasks. We also study the relationship between the accuracy of rotation recognition and the accuracy of action recognition on UCF101 dataset. The results are shown in Fig. 5. The performance of action recognition increases along with the improvement of the accuracy of rotation recognition which validates that the proposed 3DRotNet can learn meaningful features for high-level video tasks through simple recognition of rotation geometric transformations.

5.6. Ablation Study of Impact of Rotations

We further conduct experiments on the Kinetics dataset to evaluate the impact of the combination of different geometric transformation degrees to the accuracy of action recognition task under four situations: (a) Combining $0°$ and $90°$ rotations, (b) Combining $0°$, $90°$, and $180°$ rotations, (c) Combining $0°$, $90°$, $180°$, and $270°$ rotations, and (d) Combining $90°$, $180°$, and $270°$ rotations. These networks are trained on Kinetics dataset and finetuned on UCF101 dataset.

Table 3. The comparison of the performance of networks to recognize different number of rotations on UCF101 dataset. The network that recognizes 4 kinds of rotations has the best performance among all the networks.

Rotations	Combination			
$0°$ rotation	√	√	√	
$90°$ rotation	√	√	√	√
$180°$ rotation		√	√	√
$270°$ rotation			√	√
Performance	50.94%	59.24%	62.90%	58.79%

Table 3 illustrates the effects of the number of rotations to the transfer learning. The network trained for four rotations has the best performance on the transfer learning, and the network based only two rotations has the worst performance. When only two kinds of rotations are available, the finetune performance on the UCF101 dataset is only 50.94% which is 11.96% lower than the performance of the network trained with four rotations. This is because the network trained to recognize 4 rotations received more supervision signal than the network trained to recognize 2 rotations.

5.7. Kernel Comparison Between Supervised and Self-supervised Models

Here, we visualize all the kernels of the first convolution layer of the proposed self-supervised 3DRotNet and the kernels of the fully supervised model in Fig. 6. Both the proposed 3DRotNet and the supervised 3DConvNet are trained on the Kinetics dataset. The only difference is that the 3DRotNet is trained without the human-annotated category labels. As shown in Fig 6, the self-supervised model learned the similar kernels as the supervised model.

5.8. Compare with Other Self-supervised Methods

In this section, we compare our 3DRotNet with other self-supervised methods on action recognition task including the 2DConvNet-based [10, 23, 24, 44] and the 3DConvNet-based methods [41]. Following [23, 32, 46], the performance of the RGB models of these methods on the first split of the two datasets are compared.

Table 4. Comparison with the existing self-supervised methods for action recognition on the UCF101 and HMDB51 datasets. Our proposed method outperforms all the existing self-supervised methods on both datasets.

	Method	UCF101	HMDB51
2DConvNet (RGB)	Wang *et al.* [44]	40.7	15.6
	Mobahi *et al.* [24]	45.4	15.9
	Hadsell *et al.* [10]	45.7	16.3
	Misra *et al.* [23]	50.9	19.8
	Purushwalkam *et al.* [32]	55.4	23.6
	Lee *et al.* [19]	56.3	22.1
	Büchler *et al.* [2]	58.6	25.0
	Wei *et al.* [46]	58.6	—
	Sayed *et al.* [35]	59.3	27.7
3DConvNet (RGB)	Vondrick *et al.* [41]	52.1	—
	Ours	**62.9**	**33.7**

Table 4 shows the recognition accuracy on UCF101 and HMDB51 datasets. Our 3DRotNet outperforms the state-of-the-arts of fully self-supervised methods on both UCF101 and HMDB51 datasets and achieves 62.9% and 33.7% accuracy respectively.

The supervised models of the 2DConvNet-based and 3DConvnet-based methods have the state-of-the-art performance of over 90% on the UCF101 dataset [11, 43]. These models usually involve the fusion of different modalities such as the Optical Flow, RGB, and the difference between the two frames. It is unfair to directly compare with the supervised model or compare with different modalities. Therefore, we only compare with the RGB models with these state-of-the-art self-supervised methods.

Most of the 2DConvNet-based methods were trained to learn image representations with temporal supervision such as [19, 23, 32, 35]. During testing, these 2DCvonNet-based methods process RGB frames and the predictions of all the frames are fused to get the final prediction of a video. Fernando *et al.* [8] also explored to train the network with a stack of frame differences for action recognition. Büchler *et al.* proposed to learn the features with Deep Reinforcement

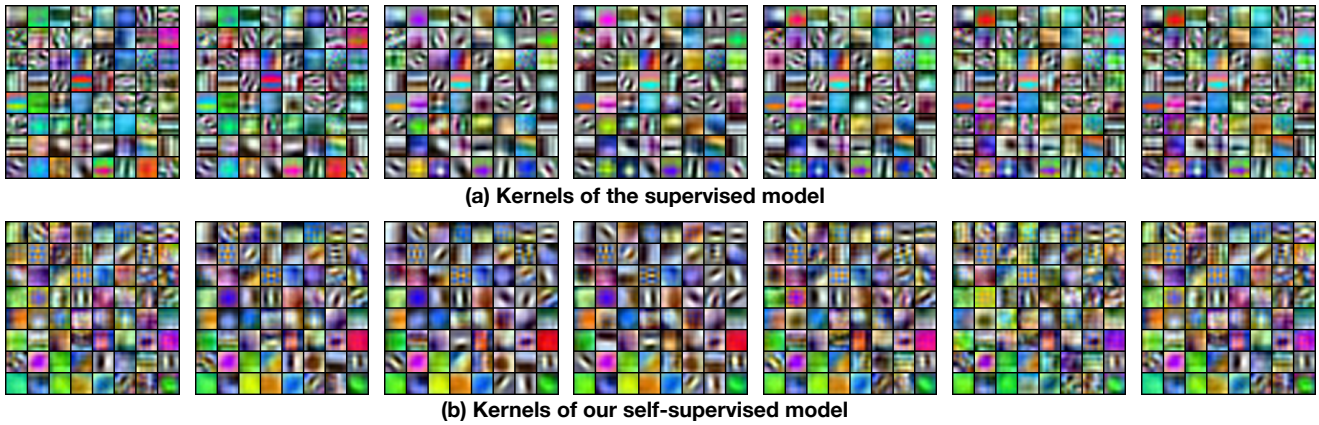

Figure 6. All the kernels of the first convolution block of our self-supervised 3DRotNet and fully supervised 3DResNet.

Learning from videos and achieved very good performance on the two datasets [2]. Sayed *et al.* proposed to learn the features from videos with the optical flow and RGB pairs and achieved the state-of-the-art performance among the 2DConvNet-based methods [35].

Due to the powerful ability of 3DConvNets to simultaneously capture both the spatial and temporal information, the 3DConvNet-based methods achieve the state-of-the-art performance on many video understanding tasks [33]. However, only Vondrick and his colleagues recently attempted to apply it for self-supervised learning [41, 42]. Vondrick *et al.* [41] employed the Generative Adversarial Network to generate videos and the discriminator network can learn the spatiotemporal features through the process without using human-labeled annotations. However, this model is not specifically designed for the self-supervised learning of spatiotemporal features, therefore, has a inferior performance. Compared to the existing self-supervised learning methods, our framework is very straightforward to implement and our network is able to simultaneously learn the spatial and temporal information.

6. Conclusion

We have proposed a straightforward and effective 3DConvNet-based approach for fully self-supervised learning of spatiotemporal features from videos. The experiment results demonstrate that video geometric transformations are able to provide essential information for the network to learn both spatial and temporal features for videos. The effectiveness of the learned video features has been evaluated on action recognition task, and the proposed framework has achieved the state-of-the-art performance on two benchmarks among all existing fully self-supervised methods.

References

[1] P. Bojanowski and A. Joulin. Unsupervised learning by predicting noise. *arXiv preprint arXiv:1704.05310*, 2017. 4

[2] U. Büchler, B. Brattoli, and B. Ommer. Improving spatiotemporal self-supervision by deep reinforcement learning. In *ECCV*, pages 770–786, 2018. 8, 9

[3] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018. 2, 3, 5

[4] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 4724–4733. IEEE, 2017. 1, 2, 3

[5] A. Diba, M. Fayyaz, V. Sharma, A. H. Karami, M. Mahdi Arzani, R. Yousefzadeh, and L. Van Gool. Temporal 3D ConvNets: New Architecture and Transfer Learning for Video Classification. *ArXiv e-prints*, Nov. 2017. 1, 2

[6] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015. 2, 3, 4, 5

[7] C. Doersch and A. Zisserman. Multi-task self-supervised visual learning. In *ICCV*, 2017. 2

[8] B. Fernando, H. Bilen, E. Gavves, and S. Gould. Self-supervised video representation learning with odd-one-out networks. In *CVPR*, 2017. 2, 3, 8

[9] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 2, 3, 4

[10] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *null*, pages 1735–1742. IEEE, 2006. 8

[11] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet. In *CVPR*, pages 18–22, 2018. 1, 2, 3, 4, 6, 7, 8

[12] E. Hoffer, I. Hubara, and N. Ailon. Deep unsupervised learning through spatial contrasting. *arXiv preprint arXiv:1610.00243*, 2016. 2

[13] S. Ji, W. Xu, M. Yang, and K. Yu. 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013. 1, 3

[14] L. Jing, X. Yang, and Y. Tian. Video you only look once: Overall temporal convolutions for action recognition. *Journal of Visual Communication and Image Representation*, 52:58–65, 2018. 2

[15] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 2

[16] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 2, 5, 6

[17] H. Kuehne, H. Jhuang, R. Stiefelhagen, and T. Serre. Hmdb51: A large video database for human motion recognition. In *HPCSE*, pages 571–582. Springer, 2013. 5, 6

[18] G. Larsson, M. Maire, and G. Shakhnarovich. Colorization as a proxy task for visual understanding. In *CVPR*, 2017. 2, 3, 4, 5

[19] H.-Y. Lee, J.-B. Huang, M. K. Singh, and M.-H. Yang. Unsupervised representation learning by sorting sequence, 2017. 2, 3, 8

[20] B. Li, Y. Dai, H. Chen, and M. He. Single image depth estimation by dilated deep residual convolutional neural network and soft-weight-sum inference. *arXiv preprint arXiv:1705.00534*, 2017. 4

[21] D. Li, W.-C. Hung, J.-B. Huang, S. Wang, N. Ahuja, and M.-H. Yang. Unsupervised visual representation learning by graph-based consistent constraints. In *ECCV*, 2016. 2, 3

[22] Y. Li, M. Paluri, J. M. Rehg, and P. Dollár. Unsupervised learning of edges. *CVPR*, pages 1619–1627, 2016. 2

[23] I. Misra, C. L. Zitnick, and M. Hebert. Shuffle and Learn: Unsupervised Learning using Temporal Order Verification. In *ECCV*, 2016. 2, 3, 5, 8

[24] H. Mobahi, R. Collobert, and J. Weston. Deep learning from temporal coherence in video. In *ICML*, pages 737–744. ACM, 2009. 8

[25] M. Monfort, B. Zhou, S. A. Bargal, T. Yan, A. Andonian, K. Ramakrishnan, L. Brown, Q. Fan, D. Gutfruend, C. Vondrick, et al. Moments in time dataset: one million videos for event understanding. 5, 6

[26] T. N. Mundhenk, D. Ho, and B. Y. Chen. Improvements to context based self-supervised learning. CVPR, 2018. 2

[27] M. Noroozi and P. Favaro. Unsupervised learning of visual represention by solving jigsaw puzzles. In *ECCV*, 2016. 2, 3, 5

[28] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash. Boosting self-supervised learning via knowledge transfer. *arXiv preprint arXiv:1805.00385*, 2018. 2

[29] J. Pan, C. C. Ferrer, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, and X. Giro-i Nieto. Salgan: Visual saliency prediction with generative adversarial networks. *arXiv preprint arXiv:1701.01081*, 2017. 4

[30] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning features by watching objects move. In *CVPR*, 2017. 2, 3, 5, 7

[31] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 2, 3, 5

[32] S. Purushwalkam and A. Gupta. Pose from action: Unsupervised learning of pose features based on motion. *arXiv preprint arXiv:1609.05420*, 2016. 8

[33] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 2017. 1, 3, 9

[34] Z. Ren and Y. J. Lee. Cross-domain self-supervised multi-task feature learning using synthetic imagery. In *CVPR*, 2018. 2, 3, 5

[35] N. Sayed, B. Brattoli, and B. Ommer. Cross and learn: Cross-modal self-supervision. *arXiv preprint arXiv:1811.03879*, 2018. 8, 9

[36] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CRCV-TR*, 12-01, 2012. 5, 6

[37] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised Learning of Video Representations using LSTMs. In *ICML*, 2015. 2, 3

[38] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, 2015. 1, 2, 3

[39] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence – video to text. In *ICCV*, 2015. 1

[40] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee. Decomposing motion and content for natural video sequence prediction. *ICLR*, 2017. 2, 3

[41] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *NIPS*, pages 613–621, 2016. 2, 3, 8, 9

[42] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy. Tracking emerges by colorizing videos. *arXiv preprint arXiv:1806.09594*, 2018. 2, 3, 9

[43] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: towards good practices for deep action recognition. In *ECCV*, 2016. 8

[44] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, pages 2794–2802, 2015. 2, 3, 8

[45] X. Wang, K. He, and A. Gupta. Transitive invariance for selfsupervised visual representation learning. In *ICCV*, 2017. 2

[46] D. Wei, J. Lim, A. Zisserman, and W. T. Freeman. Learning and using the arrow of time. In *CVPR*, pages 8052–8060, 2018. 8

[47] H. Xu, A. Das, and K. Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *ICCV*, 2017. 1

[48] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *ECCV*, 2016. 4

[49] R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017. 3

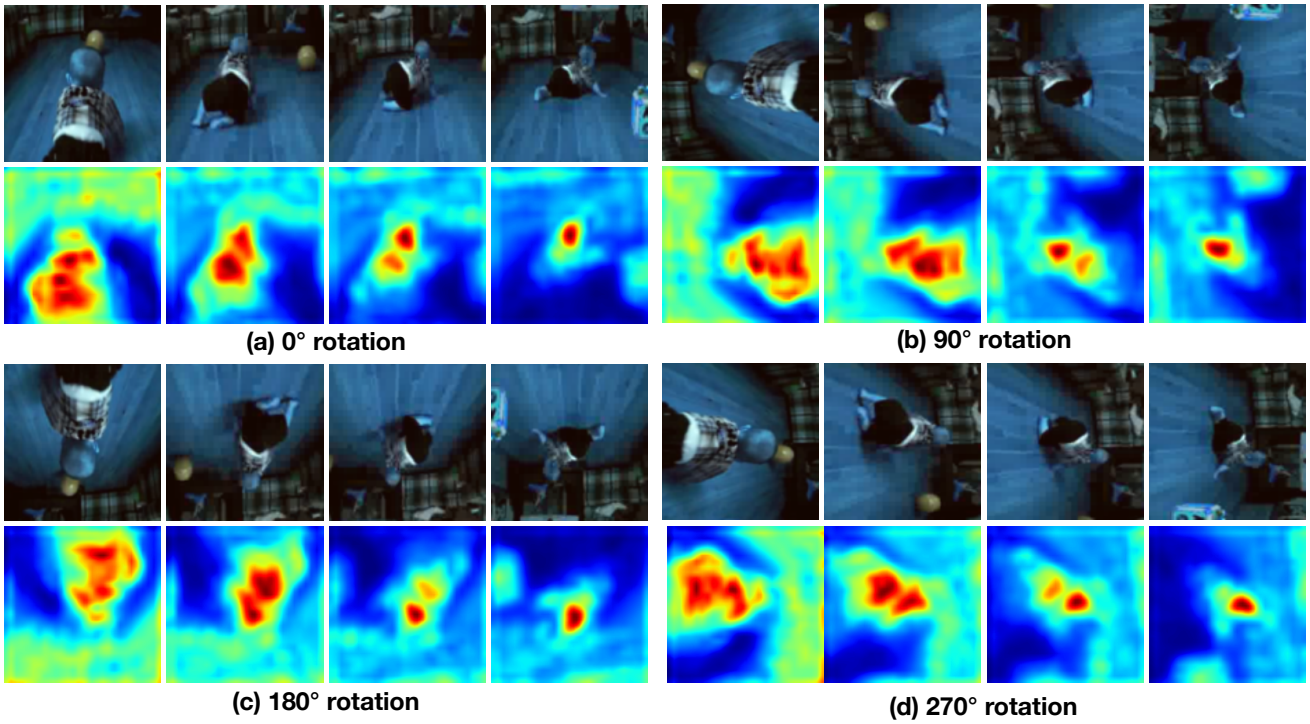

Figure 7. Sampled video frames and their corresponding attention maps generated by our proposed self-supervised 3DRotNet model at each rotation. The network focuses on the moving baby at all rotations.

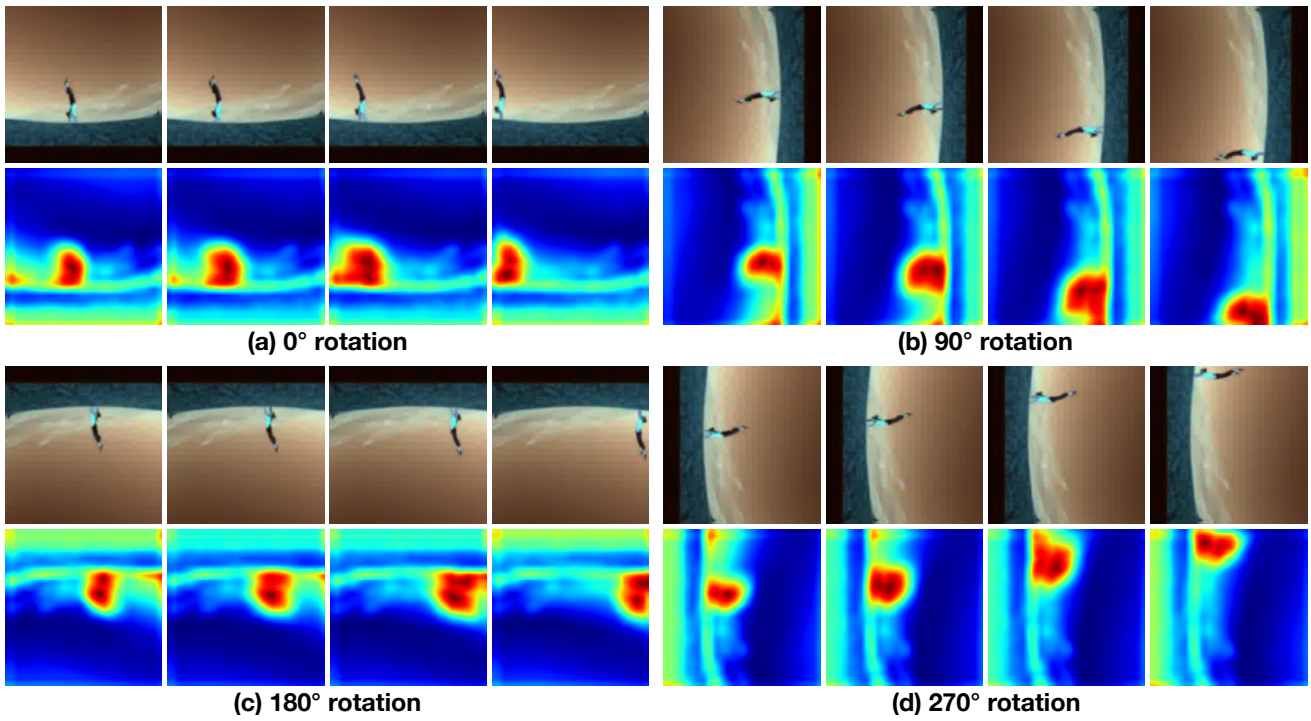

Figure 8. Sampled video frames and their corresponding attention maps generated by our proposed self-supervised 3DRotNet model at each rotation. The network focuses on the moving person in this video.

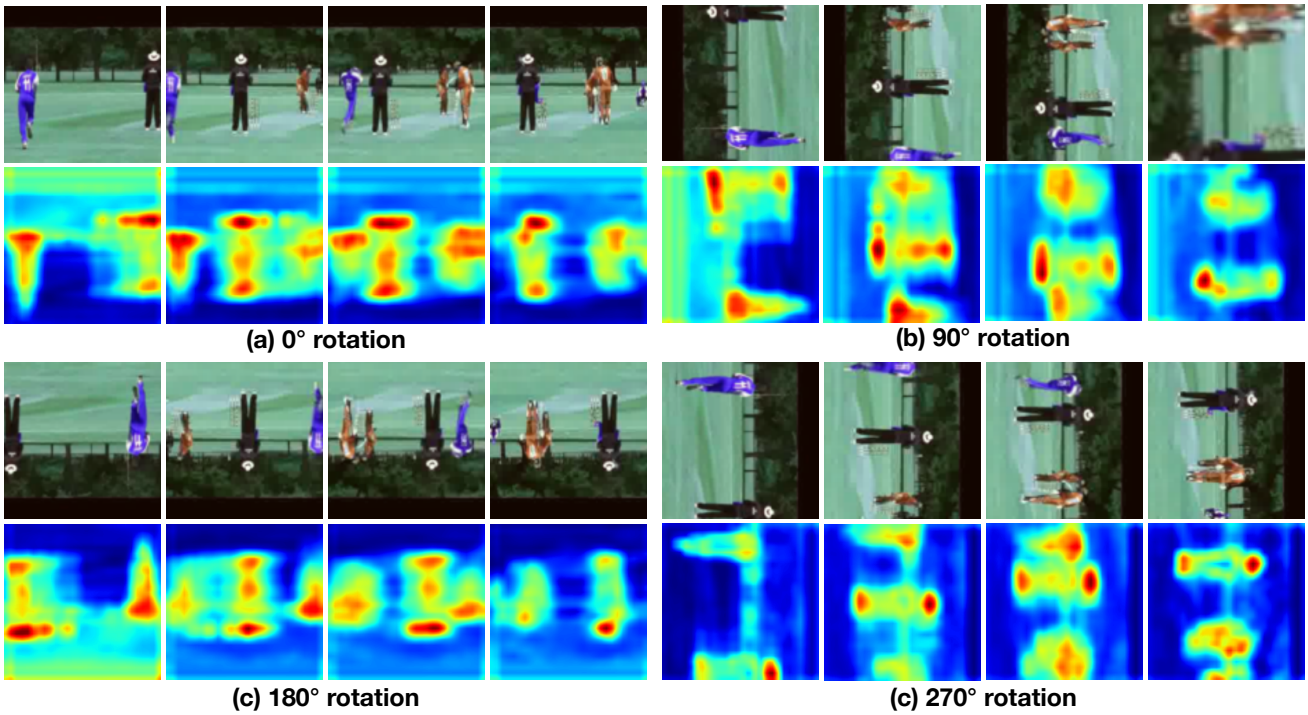

Figure 9. Sampled video frames and their corresponding attention maps generated by our proposed self-supervised 3DRotNet model at each rotation. The network can capture the multiple persons at the same time among all the frames.

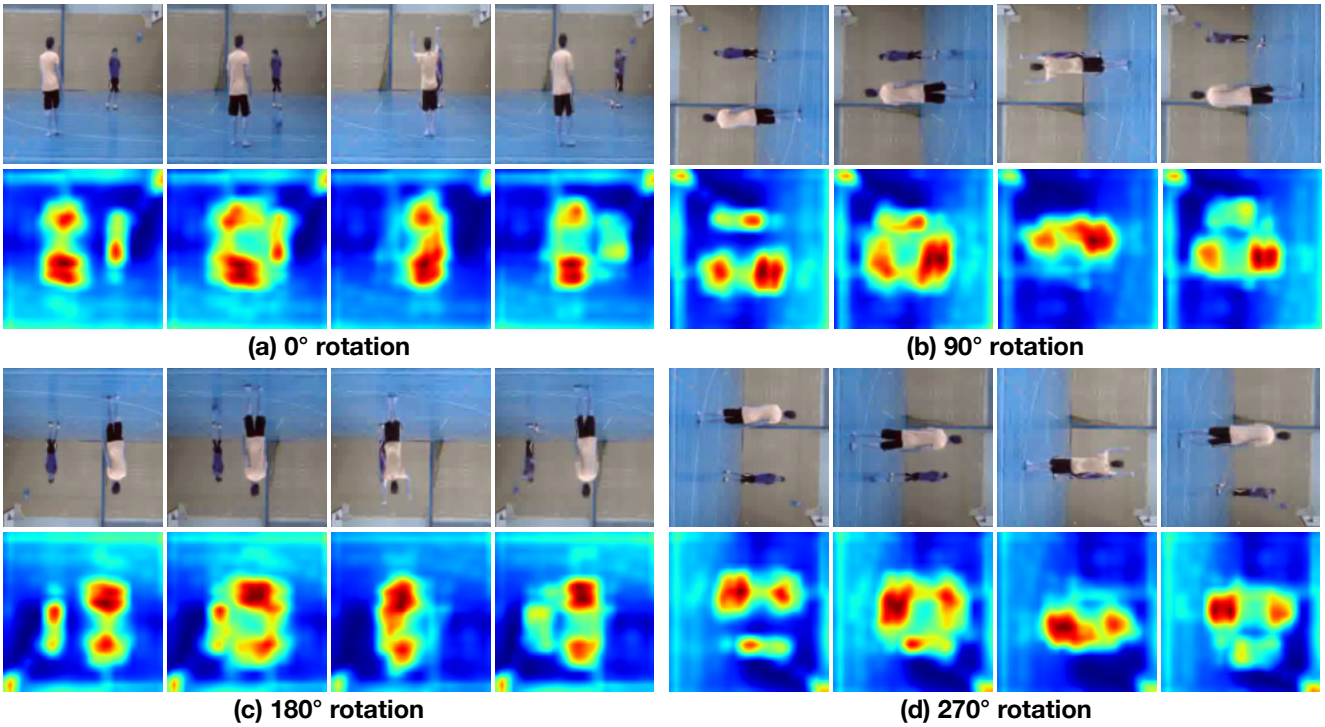

Figure 10. Sampled video frames and their corresponding attention maps generated by our proposed self-supervised 3DRotNet model at each rotation. The network can capture the two persons at the same time and focuses on the person with the most significant movement.