Temporal Contrastive Pretraining for Video Action Recognition

Guillaume Lorre, Jaonary Rabarisoa, Astrid Orcesi, Samia Ainouz-Zemouche, Stephane Canu

Temporal Contrastive Pretraining for Video Action Recognition

Guillaume LORRE[1], Jaonary RABARISOA[1], Astrid ORCESI[1], Samia AINOUZ[2], and Stephane CANU[2]

[1]CEA, LIST, Laboratoire Vision et Apprentissage pour l'Analyse de Scene, Gif-sur-Yvette, France
[2]Normandie Univ, INSA Rouen, UNIROUEN, UNIHAVRE, LITIS, France
$\{guillaume.lorre, jaonary.rabarisoa, astrid.orcesi\}$ @$cea.fr$
$\{samia.ainouz, stephane.canu\}$ @$insa-rouen.fr$

Abstract

In this paper, we propose a self-supervised method for video representation learning based on Contrastive Predictive Coding (CPC) [27]. Previously, CPC has been used to learn representations for different signals (audio, text or image). It benefits from the use of an autoregressive modeling and contrastive estimation to learn long-term relations inside raw signal while remaining robust to local noise. Our self-supervised task consists in predicting the latent representation of future segments of the video. As opposed to generative models, predicting directly in the feature space is easier and avoid incertitude problems for long-term predictions. Today, using CPC to learn representations for videos remains challenging due to the structure and the high dimensionality of the signal. We demonstrate experimentally that the representations learned by the network are useful for action recognition. We test it with different input types such as optical flows, image differences and raw images on different datasets (UCF-101 and HMDB51). It gives consistent results across the modalities. At last, we notice the utility of our pre-training method by achieving competitive results for action recognition using few labeled data.

1. Introduction

An important field in video analysis is action recognition or detection. This task allows to automatically understand the behavior of people in a video. Action recognition state of the art performances are obtained by supervised learning of deep convolutional neural networks which requires a lot of labeled data. These large datasets are costly and time consuming to acquire. That is the reason why unsupervised methods have been developed to leverage unlabeled videos and to minimize the need of annotated data. The principle of most unsupervised methods is to predict a part of the data from another one, for instance, the future of the video from the past, as in [17] or [5]. To predict future events, the model should understand the movement involved and the action performed in the video and therefore it learns useful representations for downstream tasks, such as video classification. Following this concept, we propose a new self-supervised task that predicts future frames representations. In contrast to generative methods such as [27], the model does not predict raw data which makes it possible to focus on high level information. In [28], future frame representations are also predicted but come from a pre-trained CNN. In our method, representations can be learned in the same time based on CPC.

In this work, we demonstrate the ability for the representation learned during the self-supervised task to efficiently initialize a supervised learning task of action classification. Our method improves classification accuracy on UCF-101 by 4.1 percentage point (p.p.).

Different input types (e.g. optical flows, raw images, difference between two images) are compared. Compared to state of the art, our pre-training improves results on each modality and by a large margin when using images (more 21.9 p.p. on UCF-101). However, our results using optical flows still stay better overall.

Finally, in this study, we analyze the utility of our pre-training when using few labels as it is the main challenge of unsupervised learning and as labeled data are difficult to acquire.

In the rest of the paper, related work is detailed in section 2, the presentation of our model and its evaluation follow in section 3. The setting of our experiments is described in section 4 and our results are displayed in section 5.

2. Related work

In this section, we review three main categories of prior works: action recognition, unsupervised video representa-

tion learning and self-supervised methods based on mutual information estimation. In this work, unsupervised methods include all methods that does not need human annotated information. Self-supervised methods are defined as unsupervised methods that do not generate data.

2.1. Action recognition

Given the improvements in image classification with 2D CNN, first methods in video classification aggregate temporally 2D CNN outputs to classify videos [9]. However, fine-grained movement is not taken into account with these methods. [26] proposes to use 3D convolutions that can extract spatio-temporal information. Concurrently, [21] proposes to use optical flows as input to extract movement. Optical flows and images are processed in two different streams and the class probabilites are combined by taking the average.

In our work, we chose to use 2D CNN as they showed great performances and are much more memory efficient than 3D CNN. Optical flows has proven to be an efficient representation of movement and will be therefore used in this work as well as images. Finally, even if image differences are not so common in the literature, they can represent well motion and are easier to calculate than optical flows.

2.2. Unsupervised video representation learning

Two main approaches are used for video representation learning: generative and self-supervised methods. On the one hand, generative methods predict raw data such as image frames or optical flows. Their main problem is that they need to model low level information to get back to the pixel level. On the other hand, self-supervised methods extract high level information from the data and predict this information. However, they require well engineered tasks to avoid the network from exploiting trivial solutions. In the next sections, different methods of these two families are briefly described.

2.2.1 Generative methods

Two main tasks are presented in the state of the art: future frames prediction and video generation. Most methods address the task of future frames prediction as video generation is much harder.

For future frame prediction, the majority of the approaches use an autoencoder or variational autoencoder to encode past frames and generate the future ones. Variational autoencoders are used to make diverse predictions given the same input as highlighted in [2] and [34]. A major issue in future frame generation is uncertainty. Indeed, methods using L2 reconstruction loss have blurry outputs (average of the multiple possible futures). In [16] and [30], adversarial losses are used to correct this problem. Using the autoencoder framework, pixels are predicted that is why all low-level information should be kept in all layers. To avoid this constraint, methods that transform previous frames to obtain the future ones have been used. For instance, [30] and [5] methods output parameters of a convolution that will be applied to the previous frames to obtain the future ones. Other methods such as [19] have used optical flows estimation to predict future frames. Finally, [20] combines the two precedent approaches by using displaced convolutions.

For video generation, Vondrick in [29] generates videos from noise using a Generative Adversarial Network (GAN). Two streams are used, one for the background and one for the foreground. The discriminator serves as a pre-trained network for action recognition.

Some generative methods have evaluated their representations on action recognition such as Dual Motion GAN [15] which predicts future images and optical flows. Even if methods are improving for generation (more long-time predictions), they still lag behind in term of pre-training. Main reasons is that prediction in the input space is not really appropriate and does not extract high level information.

2.2.2 Self-supervised learning

Many unsupervised methods apply transformations to the data that a network learns to predict. Listed below are some examples for video representation learning.

The authors of [18] propose to use temporal order to learn representations. Three frames are randomly selected and the network must classify if they are in a correct order or not. O3N [4] improves it by asking the network to select the video in the wrong order against N videos in the correct order. OPN [14] makes the task harder by asking the network to predict the temporal order of the frames and not only its correctness. [33] extends this method to order short clips using 3D CNN.

A spatio-temporal puzzle task is proposed in [1]. Given different crops in an image, the network must be able to replace them in space and time. [11] extends this problem to spatio-temporal volumes.

In AOT [32], segments of optical flows are used to predict if the video is playing forwards or backwards. The network must learn semantics to be able to determine it. 3DRotNet [8] rotates video clips spatially and predicts the rotation angle using a 3D CNN. In [31], global appearance and motion information are extracted and predicted.

2.3. Methods based on mutual information

New methods such as Deep Infomax (DIM) [7] and CPC [27] are based on mutual information estimation and maximization. CPC maximizes mutual information between a context and unseen parts of the data and DIM between the input and the output of a neural network. [3] extends the last

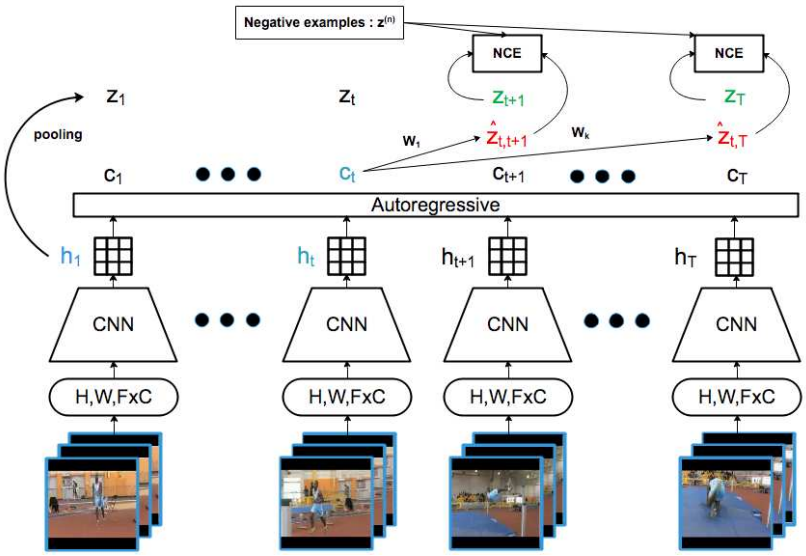

Figure 1. General architecture of our model. A CNN outputs hidden feature maps h_1, ..., h_T from T video segments. These feature maps are processed by an autoregressive model to aggregate past information. The hidden feature maps visible by the autoregressive model at time t are in blue (only past information). From the context c_t, future video segments are predicted. The predictions are in red and the targets in green. The target is a spatial pooling of the hidden feature maps as only temporal predictions are performed. Predictions and target are compared using a loss based on NCE which involves negative examples.

method to maximize mutual information between features of multiple scales, and from different augmented versions of the image. The authors of [25] use different modalities instead of augmented versions. For instance, for video representation, mutual information between images and optical flows is maximized.

These methods have improved state of the art for representation learning for images and other modalities. In this article, we propose to apply them in the context of future prediction that has been studied a lot in the generative setting but not with self-supervised methods.

In this article, we focus on the method CPC [27] as it is suited for sequences. CPC is a model that predicts future high-level representations from previous ones. Parts of the data are encoded into representations through an encoder network. An autoregressive model aggregates the representations up to the current part and returns a context. It contains the information from previous parts of the data. From this context, a prediction network estimates the k future representations.

A regression loss cannot be used because of collapse problems. They chose a loss based on Noise Contrastive Estimation (NCE) which consists of a classification between a positive example and several negative examples. This loss is also related to mutual information. Minimizing it allows to find representations that have the most information in common across the video.

CPC has been used to get unsupervised representations of different inputs such as images, sound and text. For instance, [23] predicts text and video to perform video captioning. However, it has never been used to predict future events in videos to learn useful representations for action recognition.

3. Proposed Method

In this section, the proposed model is described as well as the way it is evaluated.

3.1. Model

The goal of this method is to learn good temporal representations for action recognition without annotated videos. The unsupervised task experimented is to predict future high-level frame information given past ones, based on the CPC model [27].

As illustrates Figure 1, the model takes as input a sequence of T segments of F input frames. These input frames can be optical flows, difference of images or images. These segments are noted $x_1, \ldots, x_T$. Optical flows and image differences are interesting for this method as they do not contain background information. Indeed, background information is constant in the video. Therefore, it is easy to predict from past frames but it is not relevant semantically to learn from it. Overlapping segments by half of the length are selected (a stride of $F/2$ is used) as in [27] for images. Random transformations are applied to the different segments so that the network cannot directly rely on global

location for the predictions. This kind of transformations are common in self-supervised methods to avoid trivial solutions and the use of low-level information to perform the task. Further explanations on the pre-processing are provided in experimental details.

The F frames are stacked into the channels to use 2D CNN on spatio-temporal segments as in [21]. Therefore, the input will have the following shape: [B, T, H, W, CxF] [1]. This T video segments are then encoded by a convolutional neural network (CNN) of parameters θ which outputs hidden feature maps h_1 to h_T

$$h_t = f(x_t; \theta) \quad \text{for } t \text{ in } [1, \dots, T].$$

An autoregressive model $\mathbf{a}$ encodes the sequence h_1 to h_T into the contexts c_1 to c_T, which are feature maps of the same size as h_t. Context c_t does not have information about future events. Indeed, this is important that the network has no access to the values it is asked to predict. This can be implemented in different ways (recurrent neural networks for instance). Here temporally masked convolutions are used.

$$c_t = \mathbf{a}(h_1, ..., h_t) \quad \text{for } t \text{ in } [1, \dots, T].$$

To make predictions from the spatial pooling of c_t, T-1 linear functions of weights W_1 to W_{T-1} are used. For each c_t, the network predicts $\hat{z}_{t,t+1}$ to $\hat{z}_{t,T}$ which must correspond to z_{t+1} to z_T where z_t is the spatial mean pooling of the corresponding feature map: $z_t = p(h_t)$ (see Figure 1). All the remaining segments are predicted to enable long term predictions that are more difficult and makes it possible to learn higher level information. The representations z are not pooled before the autoregressive model so as to use powerful spatio-temporal autoregressive models to compute the context. It is a difference with CPC methods that usually predict the variables z_t which are also the input of the autoregressive model.

$$\hat{z}_{t,t+k} = W_k p(c_t) \text{ for } t \text{ in } [1, T-1] \text{ and } k \text{ in } [1, T-t].$$

Predictions $\hat{z}_{t,t+k}$ are then compared to z_{t+k} (the positive example) and N negative examples $z^{(n)}$, using the dot product as a similarity function. The results are used to make a classification with N+1 classes where the right class corresponds to the positive example. The loss is a softmax cross-entropy (see equation 1). The goal is to have a high similarity score between the prediction and the positive example and a low similarity score between the prediction and the negative examples. To this end we used the NCE loss L_N defined as:

$$L_N(z_t, \hat{z}_t) = -\log \frac{\exp(\hat{z}_t^\top z_t)}{\exp(\hat{z}_t^\top z_t) + \sum_{n=1}^{N} \exp(\hat{z}_t^\top z^{(n)})}. \quad (1)$$

[1] B: batch size, H: height, W: width, C: number of channels (3 for images, 2 for optical flows and 1 for image differences)

The final loss L is the sum of the L_N losses for all the predictions for all the videos in the batch, that is:

$$L = \sum_{i=1}^{B} \sum_{t=1}^{T-1} \sum_{k=1}^{T-t} L_N(z_{t+k}^{(i)}, \hat{z}_{t,t+k}^{(i)}). \quad (2)$$

The loss NCE L_N is interesting because it is a lower bound on mutual information as shown in [27]. This framework makes it possible to maximize the mutual information I between the context c_t and the future segment representations z_{t+k} since

$$I(z_{t+k}, c_t) > \log(N) - L_N(z_{t+k}, W_k c_t),$$

where N denotes the number of negative examples. In theory, examples must be taken independently (from the product of the marginal distribution): $(x^{(1)}, \dots, x^{(N)}) \sim \prod_{i=1}^{N} p(x^{(i)})$. Taking a large N is important as it gives a tighter bound on mutual information.

In practice, the representations of all the segments in the other videos of the batch and the other segments of the same video are used as negative examples. These latter are called difficult negative examples because they are much more similar with the positive example. There are in total $B \times T - 1$ negative examples where B is the size of the batch.

3.2. Evaluation

To evaluate our model, an action recognition network is initialized using the weights from the unsupervised pretraining. This network is learned in a supervised way. Two main methods are used for evaluation: linear classification and fine-tuning. We chose to use the mean of the representations z_t over different segments as input of the classification part (because it gives better results than using the context c_t in our experiments). A batch normalization bn (to normalize the features), a classification layer of weight W_c and a bias b_c with a softmax activation are applied to the mean of the representations. Cross-entropy on predictions o is used for supervised learning, where o is the output given by:

$$o = softmax(W_c bn(\frac{1}{T} \sum_{t=1}^{T} p(f(x_t, \theta))) + b_c).$$

For linear classification, only the dense layer is optimized (parameters W_c and b_c), while in the fine-tuning phase, all layers are optimized (W_c, b_c and θ).

4. Experiments

In this section, we describe on which dataset our method will be evaluated and we detail its implementation.

4.1. Dataset

In these experiments, three main datasets of action recognition are used: UCF-101, HMDB51 and Kinetics.

UCF-101 [22]: This dataset is composed of 13320 realistic videos coming from Youtube (27 hours in total). Each video shows an action among the 101 classes, for instance, playing a musical instrument, practicing a sport, a hobby or a work. The actions are short (few seconds for most of them).

HMDB51 [13]: It is a similar dataset as UCF-101 (videos from Youtube). It is composed of 6849 clips divided into 51 action categories. These categories represent facial actions, body movements and interactions with objects and persons. It is harder than UCF-101 dataset as there are fewer videos that are most of the time shorter and of lower quality.

Kinetics [10]: This is a much larger scale action recognition dataset than the both previous ones. It contains 495 547 videos of 10 seconds and represents 600 different actions.

Kinetics will be used only for pre-training whereas the other two datasets for pre-training and action recognition. Experiments on transfer between different datasets are conducted in table 2.

4.2. Implementation details

This section describes how the different inputs are obtained, what preprocessing is used and the choice of our networks as well as our training and testing parameters.

4.2.1 Input modalities

To evaluate the importance of the input modalities we test different types of inputs for our model which are: optical flows, image differences and images.

The optical flows are extracted using OpenCV implementations. We use optical flows computed with the TVL1 [24] algorithm for its precision and optical flows computed with Disflow [12] algorithm for its speed. Indeed the computation time is important when we have to deal with large dataset like Kinetics. The optical flows values are thresholded between $[-10, 10]$ and quantized on 256 values to have jpeg images.

For the image differences, images are transformed into grayscale and the difference is made between two consecutive frames.

4.2.2 Preprocessing

For unsupervised training, each segment is cropped independently from the other (different segments have different crops). The frames are cropped from 256×342 to 224×224. The spatial mean is subtracted for optical flow as in [21].

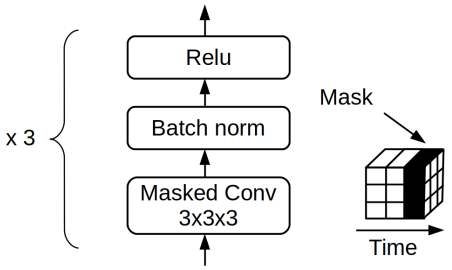

Figure 2. Details of the main autoregressive architecture. It is composed of 3 layers of temporally masked 3D convolutions separated by batchnorm and relu.

For the image modality, color dropping is used to reduce the similarity between two consecutive segments.

For supervised learning (from scratch, linear classification or finetuning), the same random crop is applied and random horizontal flipping is added. No color dropping is used for images.

4.2.3 Networks and hyperparameters

We chose Resnet18 [6] as an encoder because it is efficient and low time and memory consuming. The autoregressive model is composed of 3 layers of temporally masked 3D convolutions as shown in figure 2. The output of the Resnet18 is a feature map of size $7 \times 7 \times 512$. It is globally mean pooled for supervised learning. Batch normalization is used to reduce training time.

For the unsupervised pre-training, we chose to use 10 segments of 10 frames ($T = F = 10$). The batch size is set to 25 which leads to 249 negative examples.

4.2.4 Training

The networks are trained using Stochastic Gradient Descent with a momentum of 0.9 and regularized using a weight decay of 0.0001.

Unsupervised training: For unsupervised learning, the training lasts 150000 iterations. The initial learning rate is set to 0.01 and is decreased at iterations 20000, 40000 and 60000 when training on UCF-101. Batch size and learning rate is multiplied by 2 when learning on Kinetics (2 GPU are used).

Linear classification training: 8 segments are selected during training and 15 for testing. The batch size is set to 16. The network is trained for 20000 iterations. The initial learning rate is 0.005 and is decreased to 0.001 after 10000 iterations. A dropout of 0.5 is used. The weights of the backbone are fixed during this training step (only the weights of the classification layer are learned).

Training from scratch: As training from scratch requires to pass through a lot more of examples, the batch

PRETRAINING	FINETUNE	SPLIT1	SPLIT2	SPLIT3	MEAN	HMDB51
RANDOM	NO	21.1	23.6	22.4	22.4	10.3
OURS	NO	**78.4**	**80.1**	**80.6**	**77.9**	**45.3**
AOT [32]	NO	58.6	X	X	X	X
RANDOM	YES	80.5	83.5	82.5	82.2	52.0
OURS	YES	84.6	88.1	88.4	87.0	**58.9**
AOT [32]	YES	**86.3**	**88.6**	**88.7**	**87.9**	55.4
3D ST-PUZZLE [11]	YES	X	X	X	65.8	33.7
DUAL MOTION GAN[15]	YES	55.1	X	X	X	X
VGAN [29]	YES	X	X	X	52.1	X
SHUFFLE AND LEARN [18]	YES	50.9	X	X	50.2	18.1

Table 1. Pretraining performance on the different splits of UCF-101 with optical flows. First section shows linear classification results, the second displays finetuning results (sections are delimited by double lines). The two first sections are separated between our results and state of the art. Results on the three different splits of UCF-101 as well as results on HMDB51 are detailed as in [32]. Last section shows fine-tuning of methods using images as input.

size is increased to 64 for training but only one segment is selected (as in [21]). The number of iterations is increased to 50000. The initial learning rate is 0.01 and is decreased to 0.005 after 15000 iterations and 0.001 after 30000 iterations. A dropout of 0.9 is used to regularize the network. 15 segments are used for evaluation.

Finetuning: UCF-101 and HMDB51 datasets are quite small. Finetuning on such datasets is challenging because of overfitting and therefore losing all the benefit of the pretraining. For this problem, we employed several techniques. A stronger weight decay with origin the pretrained weights is used as well as early stopping. The same parameters (batch size and number of segments) than for linear classification are used. The network is trained for 30000 iterations. The initial learning rate is 0.01 and is decreased to 0.005 after 15000 iterations. A dropout of 0.9 is used to regularize the network.

5. Results

In this section, we first analyze the learning of the unsupervised task. Then, the results of finetuning and linear classification are studied on UCF-101, the transfer between two databases (from unsupervised pre-training to supervised finetuning) is examined too. We also explored different modalities than optical flow such as images and image differences. Finally, an ablation study is conducted and qualitative results are shown.

5.1. Unsupervised task learning

The unsupervised task consists in predicting the future segments of the video against negative ones. As shown in figure 3, the prediction accuracy decreases as we predict further away, a lot more when using difficult negative examples and random transformations on the input. Indeed, it makes the task really hard as consecutive frames are very similar which is beneficial as shown in table 3.

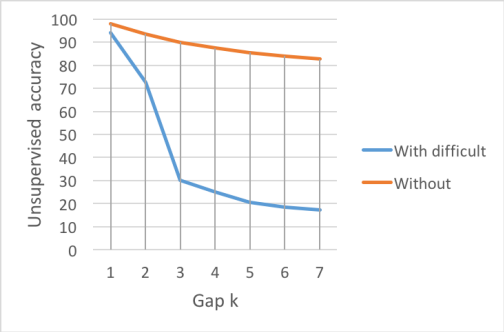

Figure 3. Unsupervised accuracy (classification between the positive and negative examples) in function of the gap between the current segment and the segment to predict (k when predicting z_{t+k} from c_t). Blue is when using difficult negative examples and random transformations and orange without

5.2. Linear classification and finetuning results on UCF-101

Main experiments on optical flows are presented in table 1. Pre-training is done on UCF-101 for those experiments. First, our model is evaluated on linear classification to measure the quality of our representation. Indeed, a powerful representation should not need finetuning to predict actions as proposed in [27]. With our pre-training, linear classification results are much higher than with a random initialized network. It also outperforms state of the art proposed by [32] by 19.8 p.p. (percentage points) with 78.4% on UCF-

101 and is really close to supervised training (only 1.9 p.p. less). It means that given our representation, actions are almost linearly separable.

Finetuning on our unsupervised pretraining improves the results in comparison to learning from scratch on UCF-101 or HMDB51. Performances are way better than state of the art results reported in [32] on HMDB51 (+ 3.5 p.p.) but are slightly lower on UCF-101. It could be explained by the fact that our unsupervised pre-training transfers better to other datasets or is better when labeled data are scarce. This is studied in next sections.

5.3. Supervised training with less data

Precedent results do not highlight the fact that less data can be used for training. In figure 4, results in linear classification and finetuning when using less data are presented. Accuracy with linear classification overcomes learning from scratch when using only 18% and 5% of the labels for the supervised training. Furthermore, the gap in accuracy between finetuning and learning from scratch increases as less labeled data are available, reaching 19.1 p.p. with only 5% of the data. An interesting point is that we achieve 80% of the result given with all the data by using only 5% of them.

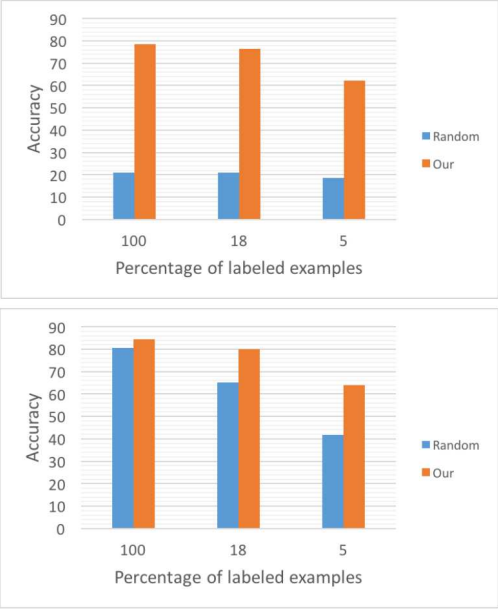

Figure 4. Pretraining performance with optical flows and different numbers of labeled examples on UCF-101. First diagram shows linear classification results and next one shows finetuning results

5.4. Results on different modalities and transfer

Experiments on other modalities and transfer from the unsupervised representation to a different dataset for the supervised task are conducted in table 2. Optical flows and image differences give better results than images. The reason is that optical flows and image differences focus on movement whereas images contain irrelevant background information. TVL1 method for optical flow estimation gives better results as more precise than Disflow but takes longer to compute. Our pre-training improves accuracies compared to learning from scratch on the two datasets UCF-101 and HMDB51 with the three different inputs. It shows that our algorithm is quite general and is not restricted to any kind of input. Pre-training on Kinetics gives better results than pre-training on UCF-101 which means that our representations get better as more unlabeled data are used which is a desirable property. Finally, our method outperforms other state of the art methods on most of the modalities. For instance, it gains 4.3 p.p. with image differences and 4.0 p.p. with images compared to [8] on HMDB51.

MODALITY	PRETRAIN	UCF-101	HMDB51
TVL1	RANDOM	80.5	52.0
	UCF-101	**84.6**	**58.9**
DISFLOW	RANDOM	76.7	46.7
	UCF-101	**79.2**	**55.6**
IMAGE DIFF	RANDOM	73.5	37.5
	KINETICS	**78.6**	**46.8**
	UCF-101	77.7	44.6
	ROTATION [8]	74.3	42.5
	OPN [14]	71.4	37.5
IMAGE	RANDOM	48.6	17.9
	KINETICS	**70.5**	**41.1**
	UCF-101	64.8	34.7
	ROTATION [8]	66.0	37.1
	PUZZLE [11]	65.8	33.7
	OPN [14]	59.8	23.8

Table 2. Evaluation of the transfer between different datasets from the unsupervised pretraining to the target task for different modalities in input. First part of the table shows our results and second one state of the art.

5.5. Ablation studies

The effect of critical elements of our method is analyzed in table 3. First, we focus on the number of future segments predicted. Accuracy improves quickly as the model predicts further away. Three future predictions already give good results whereas the reference model predicts at maximum 9 future segments. The autoregressive architecture is also an important component. Table 3 shows that using a powerful autoregressive model improves the results a lot. Conv3D (used in main model) gives better results than Conv(2+1)D and ConvLSTM. LSTM model applied on the spatial pool-

	Linear
Main model	80.1
1 step	50.3
2 step	69.9
3 step	77.6
Conv(2+1)D	77.0
ConvLSTM	73.5
LSTM	69.7
LSTM $_{w/o}$ transformation	50.0
LSTM $_{w/o}$ transformation $_{w/o}$ difficult negatives	37.9

Table 3. Ablation studies of the model: First section corresponds to the results of the reference model. Second section shows results when predicting fewer future segments (for one step, only z_{t+1} is predicted from c_t). Third section compares results between different autoregressive models. The last two lines present results when no random transformation on the input are used (a simple resize is used) and when difficult negative examples are not included. Accuracy of linear classification using the unsupervised representation learned with optical flows on the split 2 of UCF-101 are shown.

ing of the h_t gives bad results as it takes into account only temporal information. Finally, it is really important to make the unsupervised task hard. As experiments conducted on LSTM show, applying different transformations to the segments and the use of difficult negative examples greatly improves the results.

5.6. Qualitative results

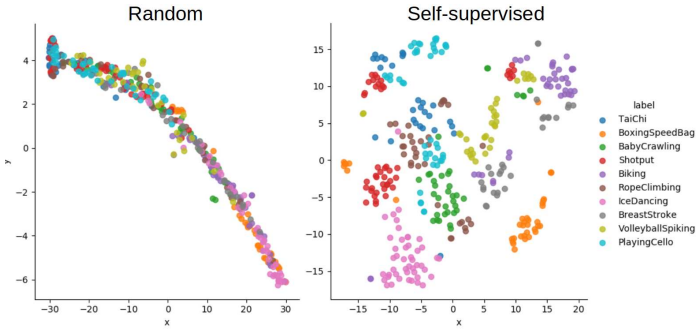

Figure 5. t-SNE of the unsupervised representations using optical flows on the test set of UCF-101 (only 10 classes among the 101 for visualization purposes). The representations are the mean of z_t for 10 segments selected across the video. Left figure shows the results when using a network initialized randomly and right one when initializing with our pre-training method.

Representations obtained by our method are visualized using t-SNE in figure 5. The representations are quite in line with the different classes. For instance, Ice Dancing, Boxing Speed Bag, Biking and Baby Crawling are well separated from the other classes. In contrast, the t-SNE visualization of the representations extracted with random initialization is not able to separate any classes.

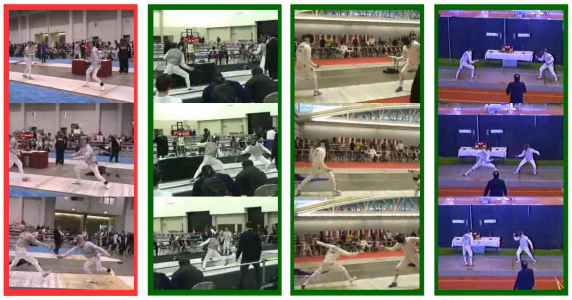

Figure 6. Most similar videos using cosine distance on the unsupervised representation using optical flows on UCF-101. Unsupervised video representations are computed by mean pooling the z_t values on 10 segments. Red represents query and green retrieved results.

In figure 6, a video is chosen and the three most similar videos are displayed. It shows that the similarity in the embedding space is quite good as retrieval results are in agreement with the human judgement and retrieved videos belong to the same class as the query.

6. Conclusion

We have proposed a new method based on Contrastive Predictive Coding to learn video representations. We showed that it learns useful representations for a downstream action recognition task, especially when labelled data are scarce. This method outperforms the state of the art for linear classification using optical flows on UCF-101. For finetuning, it achieves higher accuracies than previous methods on HMDB51 and is competitive on UCF-101. Our results prove that our method can be used with diverse inputs (optical flow, images and image differences). Even if accuracies are lower using image differences and images, results are greatly improved by our method which achieves state of the art results on those modalities. Experiments also prove that our unsupervised representations can be transfered to similar datasets (from UCF-101 to HMDB51 for instance). Finally, we highlighted the importance of long term predictions and difficult negative examples. An improvement of the method could be to calculate a motion representation directly inside the network instead of using precalculated optical flows as the algorithm used has a huge impact on the results.

Acknowledgements: Computations have been performed on the computer facilities FactoryIA of the cea List.

References

[1] U. Ahsan, R. Madhok, and I. A. Essa. Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. *CoRR*, abs/1808.07507, 2018.

[2] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine. Stochastic variational video prediction. *CoRR*, abs/1710.11252, 2017.

[3] P. Bachman, R. D. Hjelm, and W. Buchwalter. Learning representations by maximizing mutual information across views. *CoRR*, abs/1906.00910, 2019.

[4] B. Fernando, H. Bilen, E. Gavves, and S. Gould. Self-supervised video representation learning with odd-one-out networks. *CoRR*, abs/1611.06646, 2016.

[5] C. Finn, I. J. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. *CoRR*, abs/1605.07157, 2016.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[7] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019.

[8] L. Jing and Y. Tian. Self-supervised spatiotemporal feature learning by video geometric transformations. *CoRR*, abs/1811.11387, 2018.

[9] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[10] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.

[11] D. Kim, D. Cho, and I. S. Kweon. Self-supervised video representation learning with space-time cubic puzzles. *CoRR*, abs/1811.09795, 2018.

[12] T. Kroeger, R. Timofte, D. Dai, and L. V. Gool. Fast optical flow using dense inverse search. *CoRR*, abs/1603.03590, 2016.

[13] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[14] H. Lee, J. Huang, M. Singh, and M. Yang. Unsupervised representation learning by sorting sequences. *CoRR*, abs/1708.01246, 2017.

[15] X. Liang, L. Lee, W. Dai, and E. P. Xing. Dual motion GAN for future-flow embedded video prediction. *CoRR*, abs/1708.00284, 2017.

[16] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *CoRR*, abs/1511.05440, 2015.

[17] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *CoRR*, abs/1511.05440, 2016.

[18] I. Misra, C. L. Zitnick, and M. Hebert. Unsupervised learning using sequential verification for action recognition. *CoRR*, abs/1603.08561, 2016.

[19] V. Patraucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. *CoRR*, abs/1511.06309, 2015.

[20] F. A. Reda, G. Liu, K. J. Shih, R. Kirby, J. Barker, D. Tarjan, A. Tao, and B. Catanzaro. Sdcnet: Video prediction using spatially-displaced convolution. *CoRR*, abs/1811.00684, 2018.

[21] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. *CoRR*, abs/1406.2199, 2014.

[22] K. Soomro, A. R. Zamir, M. Shah, K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, page 2012.

[23] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid. Videobert: A joint model for video and language representation learning. *CoRR*, abs/1904.01766, 2019.

[24] J. Sánchez Pérez, E. Meinhardt-Llopis, and G. Facciolo. TV-L1 Optical Flow Estimation. *Image Processing On Line*, 3:137–150, 2013.

[25] Y. Tian, D. Krishnan, and P. Isola. Contrastive multiview coding. *CoRR*, abs/1906.05849, 2019.

[26] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. C3D: generic features for video analysis. *CoRR*, abs/1412.0767, 2014.

[27] A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

[28] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating the future by watching unlabeled video. *CoRR*, abs/1504.08023, 2015.

[29] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. *CoRR*, abs/1609.02612, 2016.

[30] C. Vondrick and A. Torralba. Generating the future with adversarial transformers. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2992–3000, July 2017.

[31] J. Wang, J. Jiao, L. Bao, S. He, Y. Liu, and W. Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. *CoRR*, abs/1904.03597, 2019.

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

[33] D. Xu, J. Xiao, Z. Zhao, J. Shao, D. Xie, and Y. Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[34] T. Xue, J. Wu, K. L. Bouman, and W. T. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. *CoRR*, abs/1607.02586, 2016.