# Contrastive predictive coding for video representation learning

Guillaume Lorre, Jaonary Rabarisoa, Astrid Orcesi, Samia Ainouz, Stéphane Canu

# Contrastive Predictive Coding for Video Representation Learning

**Guillaume LORRE** [1] **Jaonary RABARISOA** [1] **Astrid ORCESI** [1] **Samia AINOUZ** [2] **Stéphane CANU** [2]

## Abstract

Contrastive Predictive Coding (CPC) (van den Oord et al., 2018) has been successfully used to learn representations for different signals (audio, text, images). It uses an autoregressive modeling and contrastive estimation to learn long-term temporal relation inside the raw signal while remaining robust to local noise. The result is a higher level signal representation useful to solve downstream tasks. Using CPC to learn representations for videos remains challenging due to the structure and the high dimensionality of the signal. In this work, we propose different implementations of CPC for video signal. The learned representation increases the performance of an action recognition classifier.

## 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 bypass 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 (Mathieu et al., 2016) or (Finn et al., 2016). To predict future events, the model should understand the movement involved and action performed in the video and therefore learn useful representations for downstream tasks, such as video classification. As in (van den Oord et al., 2018), we chose not to predict raw data but representations instead. This unable the model to focus on high-level information. In (Vondrick et al., 2015), future frame representations are also predicted. However, a pre-trained CNN is used for the representations whereas in our work, the representation is learned in the same time. Our main results use optical flow as input as in (Wei et al., 2018) and predicts future optical flow representations. (Luo et al., 2017) also predicts future optical flow but using a codebook instead of a learned representation. The main contributions are to show that it is possible to learn good video representations thanks to CPC and to evaluate its usefulness on the task of action recognition. In this paper, different formulations, architectures and parameters are evaluated to get more insight on the model.

[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. Correspondence to: Guillaume LORRE <guillaume.lorre@cea.fr>.

*Proceedings of the $36^{th}$ International Conference on Machine Learning*, Long Beach, California, PMLR 97, 2019. 

## 2. Related work

In this section, we review two main categories of prior work, unsupervised video representation learning and CPC that will be applied to videos in our method.

### 2.1. Unsupervised video representation learning

Video representation learning methods split in two main families, generative methods and self-supervised methods. On one hand, generative methods output 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 so that the network cannot exploit trivial solutions. In the next sections, different methods of these two families are briefly described.

#### 2.1.1. Generative methods

Two main tasks are presented in the state of the art: future frames prediction and video generation.

In (Liang et al., 2017), a variational autoencoder predicts the next frame and optical flow of the video. To improve the results, it takes advantage of the duality between two consecutive images and optical flow. It uses the pre-trained encoder to improve action recognition results.

In (Vondrick et al., 2016), videos are generated from noise using a Generative Adversarial Network. Two streams are used, one for the background and one for the foreground. The discriminator serves as a pre-trained network for action

recognition.

#### 2.1.2. Self-supervised learning

The authors of (Misra et al., 2016) 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. (Fernando et al., 2016) improves it by asking the network to select the video in the wrong order against N videos in the correct order.

A spatio-temporal puzzle task is proposed in (Ahsan et al., 2018). Given different crops in an image, the network must be able to replace them in space and time. (Kim et al., 2018) extends this problem to spatio-temporal volumes.

In (Wei et al., 2018), 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.

### 2.2. Contrastive Predictive Coding (CPC)

The authors of (van den Oord et al., 2018) propose a model that predicts future high-level representations. Parts of the data are encoded into representations $(z_1, \ldots, z_T)$ through an encoder network. An autoregressive model aggregates the representations up to the current part and returns a context $c_t$. It contains the information from previous parts of the data. From this context, a prediction network estimates the k future representations $z_{t+1}, \ldots, z_{t+k}$.

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 (more detail is given in appendix A). Minimizing it allows to find representations that have the most information in common across the video.

## 3. Video CPC

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 Contrastive Predictive Coding model (van den Oord et al., 2018).

The model takes as input a sequence of T segments of N frames (images or optical flows). The N frames are stacked into the channels as in (Simonyan & Zisserman, 2014). Therefore, the tensor will have the following shape : [B, T, H, W, CxN] [1]. This T video segments are then encoded by a convolutional neural network which outputs representations $z_1$ to $z_T$.

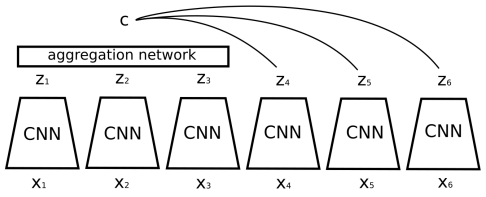


*Figure 1.* Schema of the CPC Video model

In this work, two different models are investigated. The first one takes the first representations $z_1$ to $z_m$ and aggregates them through a network. The output is a context value $c$ from which $z_{m+1}$ to $z_T$ representations are predicted. A schema of the model is shown in figure 1. The second model replaces the aggregation network by an autoregressive network to create context values $c_t$ with $t$ going from 1 to $T-1$, depending only on the past representations. From each $c_t$, $z_{t+1}$ to $z_T$ values are predicted. They will be referred as *model 1* and *model 2* in the rest of the text.

The first model is easier to interpret and has the advantage of not relying on an autoregressive model. The second model has the advantage of making multiple predictions and therefore being faster to train.

For both models, the predictions $\hat{z}_i$ are then compared to $z_i$ (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: $L = -\log \frac{\exp(\hat{z}_i . z_i)}{\exp(\hat{z}_i . z_i) + \sum_{n=1}^{N} \exp(\hat{z}_i . z^{(n)})}$

The representations of all the other videos in the batch are selected as negative examples. It is also possible to select examples from the same video but at a different time, they are called difficult negative examples. As they are more similar with the true example, they are harder to classify as a negative example.

## 4. Experiments

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

### 4.1. Evaluation

To evaluate the efficiency of our unsupervised learning algorithm, we propose two main methods : linear classification and finetuning. Eight segments of the video are randomly selected and fed into the pre-trained CNN encoder. A dense layer is added on top of it with a softmax activation. Clas-

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

sical cross-entropy loss is used for learning. For linear classification, only the dense layer is optimized, all layers are optimized for finetuning.

### 4.2. Dataset

UCF-101 dataset (Soomro et al.) is used for the unsupervised representation learning and the supervised evaluation on the task of action recognition. It 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, practising a sport, a hobby or a work. The actions are short (few seconds for most of them).

### 4.3. Implementation details

For *model 1*, the frames are cropped from $256 \times 342$ to $224 \times 224$ as a pre-processing (frame segments in input have the same crop and frame segments to predict different ones). For *model 2*, the frames are resized to $224 \times 224$. The spatial mean is subtracted for optical flow as in (Simonyan & Zisserman, 2014). We chose Resnet18 (He et al., 2015) as an encoder because it is efficient and low time and memory consuming. The 7x7 feature maps at the end of the CNN encoder are global mean-pooled if the aggregation or autoregressive model is 1D (LSTM) and not if it is 3D (TCAM and Conv3D). Precisions about the networks used can be found in appendix B.

We chose to use 8 time-steps. For the first model, 4 are used in the aggregation model and 4 are predicted. We use a batch size of 16 which allows to use $15 \times 8$ negative examples, 7 more if the difficult ones are included.

We investigated different segments selection as well. They can be selected with an overlap, in a consecutive manner or spaced. We also proposed a selection method with a variable spacing which allows to model multiple time scales.

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 60 000 iterations. The initial learning rate is set to 0.01 and is decreased at iterations 3000, 10000 and 20000.

**Linear classification training:** The network is trained for 20000 iterations. The initial learning rate is 0.005 and is decreased to 0.001 after 10000 iterations.

**From scratch and finetuning training:** 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 0.8 is used to regularize the network.

*Table 1.* **Comparison of the accuracy for images and optical flows.** *Model 1* with Conv3D aggregation network and 10 frames per segment and a variable spacing selection are used.

| MODALITY | UNSUPERVISED | SUPERVISED |
|---|---|---|
| IMAGE | 90.4 | 14.5 |
| OPTICAL FLOW | 81.7 | **55.4** |

*Table 2.* **Influence of the selection of segments and the number of flows in each segment** *Model 1* with TCAM aggregation are used. First part uses 1 flow per segment and second part variable spacing selection

| SPACING | SUPERVISED | N FLOWS | SUPERVISED |
|---|---|---|---|
| 0 | 40.9 | 1 | 44.6 |
| 3 | 43.2 | 3 | 48.8 |
| 5 | 43.5 | 5 | 48.6 |
| VARIABLE | **44.6** | 10 | **49.6** |

## 5. Results

First, we evaluate the influence of different parameters on the performances of our method thanks to linear classification. Unsupervised accuracy, which considers the classification of the true example against the negative ones is also given. Then, we compared the results of our method with random initializations and state of the art methods. In table 5, the method is evaluated using linear classification and finetuning. The supervised training is done with different number of labeled examples per class. All results given in the tables are accuracies, for supervised or unsupervised training.

We investigated the gain of using images or optical flows. As shown in table 1, with images, linear classification accuracy is largely low whereas unsupervised accuracy is higher than for optical flows. The main reason is that images in a same video are too similar which makes the unsupervised task too easy and does not enable high-level features learning. Optical flows are used for the following experiences.

The selection of the optical flows is also a very important parameter. Table 2 shows that with a higher number of optical flows in the stacks, better accuracies in linear classification are obtained. Indeed, it gives more information to the network during the unsupervised learning, information that will be also used for linear classification. If the network is asked to predict further away representations in time, it has lower unsupervised accuracies but better linear classification accuracies, as shown in table 2. It is explainable by the fact that long way predictions are harder but force the network to learn higher semantic information. The better results are obtained with the variable spacing. Segments of 10 flows selected with a variable spacing are used for the

*Table 3.* **Comparison between *model 1* and *model 2*** 10 flows per segment with variable spacing selection is used. Temporally masked Conv3D is used instead of Conv3D in the dense model

| MODEL | AGGREGATION | SUPERVISED |
|---|---|---|
| 1 | LSTM | 32.8 |
| 1 | CONV3D | **55.4** |
| 2 | LSTM | 43.6 |
| 2 | CONV3D M | 45.2 |

*Table 4.* **Influence of difficult negative examples** *Model 2* with 10 segments of 10 flows and a selection with an overlap of 5. LSTM is used as an autoregressive model

| DIFFICULT | UNSUPERVISED | SUPERVISED |
|---|---|---|
| NO | 90.3 | 37.9 |
| YES | 33.7 | **44.9** |

final results.

We compared both models described previously for different aggregation architectures. As shown in table 3, the *model 2* has better results when using a LSTM, but using masked 3D convolutions does not improve much the results, compared to *model 1* with 3D convolutions. It can be explained by the fact that 3D convolutions have more expressivity than masked ones.

We studied the effects of difficult negative examples as well. Difficult negative examples make the task harder, the unsupervised accuracy decreases significantly as shown in table 4. However, it allows to have better linear classification results as the network is forced to learn the temporal order.

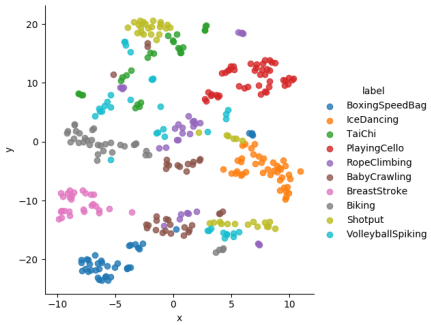


*Figure 2.* t-SNE visualization of the obtained representations

Representations obtained by our method are visualized using t-SNE in figure 2 (only 10 classes among the 101 for visualization purposes). The representations are quite in line with the different classes. For instance, Boxing Speed Bag, Breast Stroke and Playing Cello are well separated from the other classes.

*Table 5.* **Pretraining performance with different numbers of labeled examples.** 10 flows with a variable spacing selection are used. *Model 1* is used with TCAM and Conv3D aggregation networks. First 2 sections shows linear classification results, the 2 next finetuning results. Last section shows finetuing of methods using images as input

| PRETRAINING | FINETUNE | ALL | 18 | 5 |
|---|---|---|---|---|
| RANDOM | NO | 21.1 | 20.9 | 18.5 |
| AGGREG TCAM | NO | 49.6 | 47.8 | 34.7 |
| AGGREG CONV3D | NO | 55.4 | **51.6** | **37.5** |
| (WEI ET AL., 2018) | NO | **58.6** | X | X |
| RANDOM | YES | 76.4 | 65.3 | 41.8 |
| AGGREG TCAM | YES | **80.5** | 70.8 | **50.0** |
| AGGREG CONV3D | YES | 80 | **71.4** | 49.5 |
| (WEI ET AL., 2018) | YES | **86.3** | X | X |
| (KIM ET AL., 2018) | YES | 65.8 | X | X |
| (LIANG ET AL., 2017) | YES | 55.1 | X | X |
| (VONDRICK ET AL., 2016) | YES | 52.1 | X | X |
| (MISRA ET AL., 2016) | YES | 50.2 | X | X |

Table 5 shows that without finetuning, our pre-trained model has better results than random initialization, but does not reach the state of the art accuracy (Wei et al., 2018). The accuracy also drops when using less labels per class. When finetuning, our method has better results than learning from scratch. Using all the labels, our pre-training method gains 4.1% and 8.2% when using only 5 labels per class. The gap between finetuning and learning from scratch performances increases with label scarcity. We can observe that using Conv3D as an aggregation network gives better results in linear classification than TCAM but similar results when finetuning. Last methods of table 5 have worst results as they use images as input.

## 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 labeled data is scarce and even if the neural network used is rather small. We found that the use of optical flow and long term predictions are essential for this method. Furthermore, we highlighted the benefits to use a dense model and difficult negative examples. Scaling up the model and the dataset used for unsupervised training could improve the performances and is let to further experiments.

**Acknowledgements:** This work was partly supported by BPI France through the ETS project.

## References


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

Fernando, B., Bilen, H., Gavves, E., and Gould, S. Self-supervised video representation learning with odd-one-out networks. *CoRR*, abs/1611.06646, 2016. URL http://arxiv.org/abs/1611.06646.

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

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

Kim, D., Cho, D., and Kweon, I. S. Self-supervised video representation learning with space-time cubic puzzles. *CoRR*, abs/1811.09795, 2018. URL http://arxiv.org/abs/1811.09795.

Liang, X., Lee, L., Dai, W., and Xing, E. P. Dual motion GAN for future-flow embedded video prediction. *CoRR*, abs/1708.00284, 2017. URL http://arxiv.org/abs/1708.00284.

Luo, Z., Peng, B., Huang, D., Alahi, A., and Fei-Fei, L. Unsupervised learning of long-term motion dynamics for videos. *CoRR*, abs/1701.01821, 2017.

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

Misra, I., Zitnick, C. L., and Hebert, M. Unsupervised learning using sequential verification for action recognition. *CoRR*, abs/1603.08561, 2016. URL http://arxiv.org/abs/1603.08561.

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

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

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018. URL http://arxiv.org/abs/1807.03748.

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

Vondrick, C., Pirsiavash, H., and Torralba, A. Generating videos with scene dynamics. *CoRR*, abs/1609.02612, 2016. URL http://arxiv.org/abs/1609.02612.

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