LAMV: Learning to align and match videos with kernelized temporal layers

Lorenzo Baraldi[1]*, Matthijs Douze[2], Rita Cucchiara[1], Hervé Jégou[2]
[1]University of Modena and Reggio Emilia
[2]Facebook AI Research

Abstract

This paper considers a learnable approach for comparing and aligning videos. Our architecture builds upon and revisits temporal match kernels within neural networks: we propose a new temporal layer that finds temporal alignments by maximizing the scores between two sequences of vectors, according to a time-sensitive similarity metric parametrized in the Fourier domain. We learn this layer with a temporal proposal strategy, in which we minimize a triplet loss that takes into account both the localization accuracy and the recognition rate.

We evaluate our approach on video alignment, copy detection and event retrieval. Our approach outperforms the state on the art on temporal video alignment and video copy detection datasets in comparable setups. It also attains the best reported results for particular event search, while precisely aligning videos.

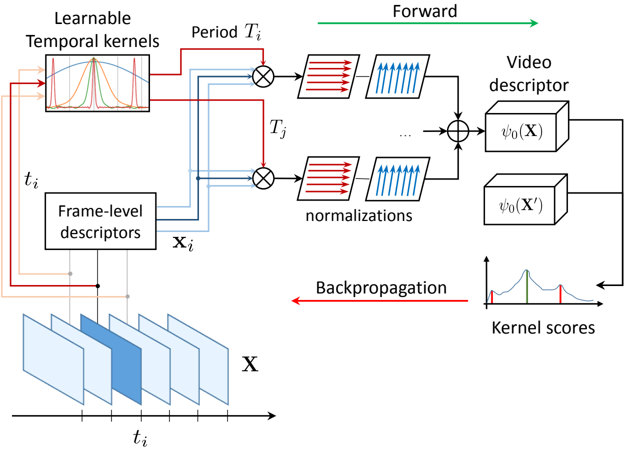


Figure 1: We present a learnable temporal layer that compares and precisely aligns videos by means of multi-period temporal kernels parametrized in the Fourier domain.

1. Introduction

Thanks to the success of neural networks and the availability of large annotated collections of images like Imagenet [4] and COCO [23], we have recently witnessed drastic improvements on many core computer vision problems, such as image classification [22, 14] and segmentation [13]. The analysis of videos has largely benefited from this game-changing adoption of neural networks, in particular by exploiting state-of-the-art image networks. Current methods for tackling video-related tasks mostly rely on the trunk of neural network architectures trained on images [32, 34, 9, 21].

Many attempts to exploit the temporal axis of videos within neural architectures have been proposed. These approaches typically extract information at the frame level and subsequently enforce or mesure the temporal consistency. For instance, Kang *et al.* [21] propose a temporal convolutional network to regularize object detection results. Fernando *et al.* [9] postulate that a method able to temporally re-order the frames of a video would be more suitable to detect the evolution of appearance, and use this supervision signal to improve action recognition. Diba *et al.* [5] investigate different ways of aggregating feature maps from image-level convolutional neural networks to achieve an end-to-end learning of a video representation.

On the contrary, only few works consider learning a joint spatio-temporal representation, like the C3D network [32]. Several difficulties may explain this situation. First, the amount of temporally-labelled data is limited: for large collections the annotation is provided at the video level only, or automatically extracted, or both [1]. Second, the number of parameters to learn a spatio-temporal representation is generally much larger than for still images. Third, depending on the task, it is not obvious that temporality is at all useful. For instance, the recent high-profile leaderbord

[*] This work was carried out while L. Baraldi was in a internship at Facebook AI Research.

This academic article may contain images and/or data from sources that are not affiliated with the article submitter. Inclusion should not be construed as approval, endorsement or sponsorship of the submitter, article or its content by any such party.

competition[1] on video understanding was won by a technique agnostic of temporality [24].

In this paper, we tackle the task of retrieving and aligning similar video instances. This problem arises in different applications such as copy detection, particular event detection, video editing and re-purposing. In the literature, one can distinguish the methods offering temporal alignment and those discarding the time information, typically through temporal pooling operations. According to a comparative study on copy detection conducted in 2014 [19], the best methods were relying on local descriptors and frame-based matching [18], even though temporal alignment is often needed later, for example to manually verify a copyright infringement. In contrast, the state of the art for particular event retrieval [6, 11] exploits a single vector per video.

Similarly, because accurate video alignment requires matching with a frame-level granularity, methods based on temporal pooling [8, 32, 10, 25] inevitably introduce some invariance to small time shifts. They are therefore not appropriate for achieving high localization accuracy.

In order to preserve the capability to align videos while offering a competitive recognition accuracy, another line of research considers Fourier-domain representations, like the circulant temporal encoding (CTE) [28, 7] inspired by prior works on tracking with correlation filters [16, 17]. In our work, we consider the temporal matching kernel (TMK) by Poullot *et al.* [26]. This representation consists of complementary periodic encodings of a sequence of frames into a fixed-sized representation. It provides both an accurate matching and alignment hypothesis, and outperforms CTE [28] in terms of alignement accuracy.

An advantage of TMK is that it disentangles the visual and temporal aspects while keeping the temporal consistency. Our proposal revists temporal match kernels in the context of a neural network. More specifically, we propose a temporal layer inspired by TMK [26]. The design is modified and the parameters are learned with a supervision signal that takes into account both the matching quality and the precision of the alignement. This is in contrast to the original technique, where the parameters are hand-crafted by a choice of a specific kernel (Von Mises). To train our layer, we adopt a temporal proposal strategy providing both positive and negative examples. The learning is performed on both real and synthetic data simulating temporal and visual attacks undergone by videos for our different tasks.

As a complementary contribution, we provide guidelines for tuning the hyper-parameters, in particular the design of better complementary elementary kernels. This, by itself, provides a significant boost, leading us to outperform the state of the art for temporal video alignment, copy detection and event retrieval on the public benchmarks Madonna [7], Climbing [7], VCDB [19] and EVVE [28].

The rest of this paper is organized as follows. After reviewing the fundamentals of temporal match kernels in Section 2, we introduce our approach in Section 3 and evaluate it in Section 4.

2. Related work and Temporal kernels

For a given video to describe, we consider a sequence of frame descriptors extracted at distinct timestamps $\mathcal{T} = \{t_1, \dots, t_i, \dots\}$. Each frame f_i is represented as a tuple $(\mathbf{x}_i, t_i)$, where $\mathbf{x}_i$ is a d-dimensional vector and t_i denotes the scalar timestamp of the frame. The frame descriptor $\mathbf{x}_i$ is typically obtained by post-processing hand-crafted or CNN-based representations. We assume that the frame descriptors are ℓ_2-normalized and are compared with inner products, or equivalently with the cosine similarity.

Joint frame and timestamp encoding. We consider a kernel function between frames descriptors such that the similarity between a pair of descriptors takes into account their absolute position in time. This operation is commonly referred to as a *modulation*. Formally, it amounts to defining a kernel between frame descriptors $\mathbf{x}$ and $\mathbf{x}'$ with respective timestamps t and t' as

$$k\left((\mathbf{x}, t), (\mathbf{x}', t')\right) = \langle \mathbf{x}, \mathbf{x}' \rangle k_{\mathrm{t}}(t, t') \tag{1}$$

$$= \langle \mathbf{x}, \mathbf{x}' \rangle \varphi(t)^\top \varphi(t'), \tag{2}$$

where $\varphi(\cdot)$ is a feature map function approximating the kernel k_{t} between timestamps, which lowers the similarity between frames that are distant in time. By convention, we set $k_{\mathrm{t}}(t, t') = 0$ if t or t' are outside the range of the valid timestamps for the two videos. Further algebraic manipulation reveals that this kernel can be expressed as

$$k((\mathbf{x}, t), (\mathbf{x}', t')) = \left(\mathbf{x} \otimes \varphi(t)\right)^\top \left(\mathbf{x}' \otimes \varphi(t')\right), \tag{3}$$

where $\otimes$ is the Kronecker product. Therefore, we describe the tuple $(\mathbf{x}_t, t)$ by a single feature vector, namely $\mathbf{x}_t \otimes \varphi(t)$.

Temporal match kernel. Given two videos represented by the sequences of frame descriptors $\mathbf{X} = \{(\mathbf{x}_i, t_i)\}_i$ and $\mathbf{X}' = \{(\mathbf{x}'_j, t'_j)\}_j$, we consider the temporal kernel

$$\mathcal{K}_\delta(\mathbf{X}, \mathbf{X}') = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} k((\mathbf{x}_i, t_i), (\mathbf{x}'_j, t'_j + \delta)), \tag{4}$$

that compares the videos on a frame-by-frame basis, assuming that the videos are shifted in time by the duration δ. With Eqn. 3, this kernel is subsequently re-written as

$$\mathcal{K}_\delta(\mathbf{X}, \mathbf{X}') = \underbrace{\Bigg(\sum_{i=0}^{\infty} \mathbf{x}_i \otimes \varphi(t_i)\Bigg)}_{\psi_0(\mathbf{X})}^\top \underbrace{\Bigg(\sum_{j=0}^{\infty} \mathbf{x}'_j \otimes \varphi(t'_j + \delta)\Bigg)}_{\psi_\delta(\mathbf{X}')}, \tag{5}$$

[1] https://www.kaggle.com/c/youtube8m/leaderboard

where $\psi_0(\mathbf{X})$ is the descriptor associated with the first video, and $\psi_\delta(\mathbf{X}')$ is the descriptor associated with the second video and re-mapped to the new time origin δ.

In the temporal match kernel from Poullot *et al.* [26], k_{t} is expressed by means of a Fourier approximation with period T and M coefficients. In this case, the feature vector representing a video can be written as

$$\psi_0(\mathbf{X}) = \left[V_0^\top, V_{1,c}^\top, V_{1,s}^\top, ..., V_{M,c}^\top, V_{M,s}^\top\right]^\top, \tag{6}$$

where:

$$V_0^\top = \sqrt{a_0} \sum_{t_i \in \mathcal{T}} \mathbf{x}_i \tag{7}$$

$$V_{m,c}^\top = \sqrt{a_m} \sum_{t_i \in \mathcal{T}} \mathbf{x}_i \cos\left(2m\pi t_i/T\right) \tag{8}$$

$$V_{m,s}^\top = \sqrt{a_m} \sum_{t_i \in \mathcal{T}} \mathbf{x}_i \sin\left(2m\pi t_i/T\right), \tag{9}$$

where a_m are the coefficients of the Fourier series. If $\mathcal{T}$ consists of evenly-spaced timestamps[2], this is equivalent to taking the Fourier transform of the input time series with period T and convolving it with $\varphi(t)$. It leads to a feature vector with dimensionality $d \times (2m+1)$.

Alternative choices for φ exist. For instance, this kind of kernel approximation was first defined with random Fourier features [27]. Vedaldi and Zisserman [33] show that explictly using the Fourier decomposition gives a much better approximation of shift-invariant kernels. By departing from the Fourier basis, Chum [3] shows how to learn sparse feature maps improving the compromise between the number of coefficients and the approximation of a kernel.

Trigonometric polynomial of scores. At this stage, $\psi_0(\mathbf{X})$ is a representation of the video. The first component V_0 is the average frame descriptor and can be used to directly compare two videos, in this case discarding the temporal information. Yet one of the strength of the chosen kernelization is that it keeps a latent variable and allows the maximization of the kernel w.r.t. this variable. This property was first exploited by Tolias *et al.* [30] when aggregating local descriptors. Bursuc *et al.* [2] exploit it to define a kernel local descriptor that automatically adjust the orientation and scale to maximize the matching score when provided with two candidate descriptors.

In our context, the latent variable is the relative time offset between the two videos. Consider a given alignment hypothesis and two videos $\mathbf{X}$ and $\mathbf{X}'$: the similarity between two video sequences is computed, for a given alignment hypothesis, as

$$\begin{aligned}\mathcal{K}_\delta(\mathbf{X}, \mathbf{X}') = {} & V_0^\top V_0' \\ & + \sum_{m=1}^{M} \cos\left(\frac{2m\pi\delta}{T}\right)\left(V_{m,c}^\top V_{m,c}' + V_{m,s}^\top V_{m,s}'\right) \\ & + \sum_{m=1}^{M} \sin\left(\frac{2m\pi\delta}{T}\right)\left(-V_{m,c}^\top V_{m,s}' + V_{m,s}^\top V_{m,c}'\right).\end{aligned} \tag{10}$$

Therefore, the score as a function of δ is a trigonometric polynomial of degree M. Evaluating this polynomial at regular timestamps is efficient and only requires $1+4M$ dot products between vectors of dimension d.

Multiple periods. Poullot *et al.* [26] employ multiple kernels with distinct periods, shorter than the video length, and take the sum of the kernel scores as the final similarity measure. This increases localization accuracy while inducing a large period for the kernel summation.

3. Proposed approach: LAMV

We revisit the temporal match kernel as a global video descriptor to compute the similarity between videos and align them temporally. This approach is referred to as LAMV (Learning to Align and Match Videos). For this, we transform the kernel into a differentiable layer, and learn the coefficients of the feature transform by imposing a triplet loss that jointly takes into account (i) the similarity scores produced when comparing two videos globally, and (ii) the temporal alignment accuracy when processing overlapping videos. The batches on which the loss is evaluated contain hard negative proposals. We also devise a normalization strategy that enhances retrieval and alignment performance.

3.1. Overview: Layerizing temporal match kernels

All the operations involved in the computation of scores produced by the temporal match kernel are differentiable with respect to their parameters a_i, even when using multiple periods. The kernel can be seen as a differentiable layer that can compute the similarity between two videos. The Fourier coefficients of the feature map $\varphi(\cdot)$ are parameters that can be learned by backpropagating a supervision signal built on the similarity scores.

The LAMV layer can aggregate frame-level features to compute a video feature vector, and it can then compare two videos by shifting one of the two descriptors. In this regard, its structure resambles that of Siamese networks, in which the same function is applied to two branches, then compared by a distance function.

Given a set $\mathcal{P}$ of periods for which the kernel is computed, each video segment $\mathbf{X}$ is encoded by taking a Fourier

[2] For typical choices of the kernel φ, one can use unevenly-spaced timestamps, such as those chosen by a frame selection technique. The method can also compare videos with different frame rates (25 Hz vs 30 Hz). In this paper, the timestamps evenly selected at a frequency of 15 Hz in order to be closer to the setup proposed in the literature [28, 7].

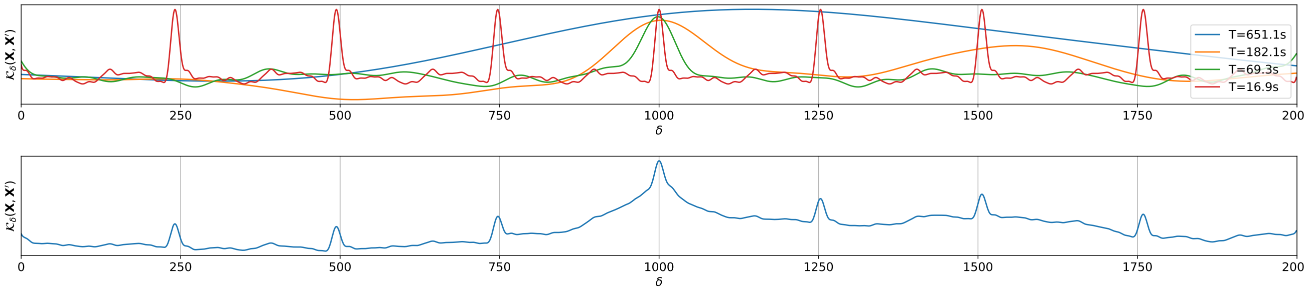


Figure 2: Response of the individual filters (top) when matching a video with a temporally-cropped excerpt of the same video. The bottom figure shows the combination of the response. The ground truth alignment point is $\delta^* = 1000$.

transform for each of the periods in $\mathcal{P}$, and subsequently applying the feature map $\varphi(\cdot)$ according to Eqn. 6. This process results into a tensor $\psi_0(\mathbf{X})$ with dimensionality $d \times (2m+1) \times \ |\mathcal{P}|$, where one of the axes is along the different periods.

Two video features are compared for a set of time shifts $\{\delta_0, ..., \delta_i, ...\}$ by taking the dot products of Eqn. 10 for each period and then summing, resulting in a scalar score for each shift. Once a loss function L_δ is defined over the score obtained for a time shit δ, its partial derivative with respect to the learnable Fourier coefficients of each of the periods in $\mathcal{P}$ are expressed from the derivatives as

$$\frac{\delta \mathcal{K}_\delta(\mathbf{X},\mathbf{X}')}{\partial a_0} = \tilde{V}_0^\top \tilde{V}_0' \tag{11}$$

and

$$\frac{\delta \mathcal{K}_\delta(\mathbf{X},\mathbf{X}')}{\partial a_m} = \cos\left(\frac{2m\pi\delta}{T}\right)(\tilde{V}_{m,c}^\top \tilde{V}_{m,c}' + \tilde{V}_{m,s}^\top \tilde{V}_{m,s}') + \\ + \sin\left(\frac{2m\pi\delta}{T}\right)(-\tilde{V}_{m,c}^\top \tilde{V}_{m,s}' + \tilde{V}_{m,s}^\top \tilde{V}_{m,c}') \tag{12}$$

where we define $\tilde{V}_0$ and $\tilde{V}_{m,*}$ as $\frac{V_0}{\sqrt{a_0}}$ and $\frac{V_{m,*}}{\sqrt{a_m}}$, respectively.

Normalizations. With the aim of reducing the interferences caused by the strong self-similarity present in videos, we apply two normalization steps which improve the alignment and retrieval performance of the descriptor. First, the $\tilde{V}_0$ and $\tilde{V}_{m,*}$ vectors are ℓ_2-normalized, so that $\psi_0(\mathbf{X})$ becomes a concatenation of normalized vectors, each weighted by its corresponding coefficient. Then, we ℓ_2-normalize $\psi_0(\mathbf{X})$ over its frequency axis. The norms computed in this stage are independent of δ, so the video feature vector $\psi_0(\mathbf{X})$ can be normalized once and then shifted multiple times using trigonometric polynomials to compute the final scores.

Figure 2 reports an example of the scores obtained at different time shifts for two matching videos. As it can be seen, long periods ($T = 651$s) fail to provide enough localization accuracy, while shorter periods ($T = 16.9$s) provide good localization but generate frequent false positives. The sum of the scores obtained with different periods increases localization accuracy while avoiding false positives.

3.2. Loss function

Ideally, kernel scores $\mathcal{K}_\delta(\cdot,\cdot)$ should be higher for overlapping videos and lower for non overlapping videos, so to enhance the retrieval of similar or overlapping videos. At the same time, the layer should perform a precise localization, which corresponds to requiring that the kernel scores for a pair of overlapping videos are higher near to the ground truth alignment point, and lower for incorrect alignment points.

Given a triplet of videos $(\mathbf{X}_0, \mathbf{X}_+, \mathbf{X}_-)$, where $\mathbf{X}_+$ overlaps with $\mathbf{X}_0$ and $\mathbf{X}_-$ does not overlap with $\mathbf{X}_0$, we define a retrieval loss that enforces kernel scores to be globally higher for the overlapping pair than for the non overlapping pair. This is done by placing a margin loss between the maximum of the kernel scores obtained when evaluating $(\mathbf{X}_0, \mathbf{X}_+)$ and $(\mathbf{X}_0, \mathbf{X}_-)$:

$$L_r = \max\left(0, m_r + \mathcal{K}^*(\mathbf{X}_0, \mathbf{X}_-) - \mathcal{K}^*(\mathbf{X}_0, \mathbf{X}_+)\right), \tag{13}$$

where $\mathcal{K}^*(\mathbf{X},\mathbf{X}')$ is the maximum of $\mathcal{K}_\delta(\mathbf{X},\mathbf{X}')$, *i.e.* $\mathcal{K}^*(\mathbf{X},\mathbf{X}') = \max_\delta \mathcal{K}_\delta(\mathbf{X},\mathbf{X}')$, and m_r is the retrieval margin.

To enforce a correct localization inside the overlapping pair, instead, we define a localization loss which imposes a margin between the kernel scores in a neighborhood of the correct alignment point δ^*, and the kernel scores outside the neighborhood:

$$L_l = \max(0, m_l + \mathcal{K}^{\mathcal{N}(\delta^*)}(\mathbf{X}_0, \mathbf{X}_+) - \mathcal{K}^{\mathcal{O}(\delta^*)}(\mathbf{X}_0, \mathbf{X}_+)), \tag{14}$$

where m_l is the localization margin, δ^* is the ground truth alignment point, $\mathcal{K}^{\mathcal{N}(\delta^*)}(\mathbf{X}_0, \mathbf{X}_+)$ is the maximum of kernel scores in a neighborhood $[\delta^* - r, \delta^* + r]$, and $\mathcal{K}^{\mathcal{O}(\delta^*)}(\mathbf{X}_0, \mathbf{X}_+)$ is the maximum of kernel scores outside the neighborhood r.

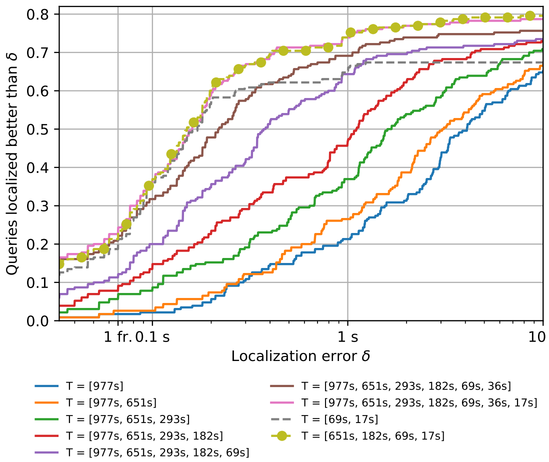


Figure 3: Fraction of correct alignments as a function of the acceptance threshold for several combinations of periods.

3.3. Learning with temporal proposals

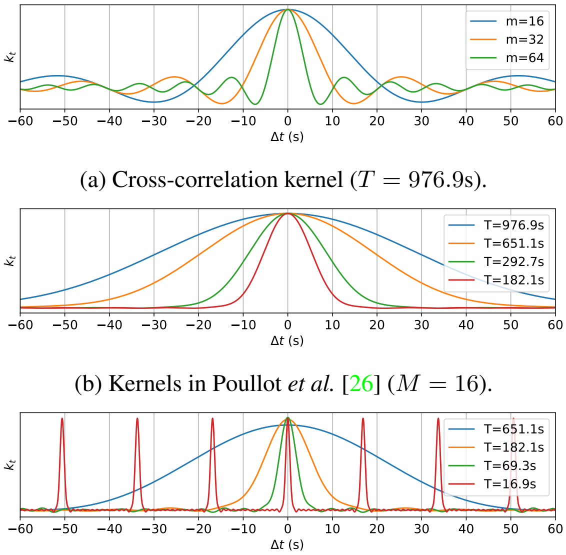
To learn the parameters of the layer, we exploit a dataset of video sequences aligned on a global timeline. In this setting, we know which sequences overlap with which sequences, and we can build suitable training triplets.

Overlapping sequences can be very long and using the entire sequences would result in a reduction of the mini-batch size (because of GPU memory limitations). On the other hand, using very short snippets would downgrade the recognition performance of the layer and create inconsistencies between the train and test phases. The length of training snippets should be related to the longest period in $\mathcal{P}$. In our case, we build training triplets made of 500 frames snippets (which at 15 fps amounts to 33.3 s).

To speed up convergence, we perform negative mining. At each epoch, we build a training triplet for each pair of overlapping videos contained in the dataset. The $\mathbf{X}_0$ snippet is sampled randomly from one of the two videos, while the matching snippet $\mathbf{X}_+$ is obtained by randomly sampling a sequence from the other video, with at least a 75% overlap with $\mathbf{X}_+$. In this way, we guarantee that the ground truth alignment point is random, and that coefficients of long periods can be properly learned. To select a hard negative $\mathbf{X}_-$, we sample a random snippet from 20 videos which do not overlap with $\mathbf{X}_0$ and $\mathbf{X}_+$, and select the one having the highest $\mathcal{K}^*(\mathbf{X}_0, \cdot)$ for the current set of weights.

3.4. Multiple period design

The choice of the periods in $\mathcal{P}$ influences both localization and recognition, as well as the maximum video length the network can process. When summing two periodic signals with periods T_1 and T_2, the resulting signal is periodic with period $T_1 \cdot T_2 / \mathrm{gcd}(T_1, T_2)$, where $\mathrm{gcd}(\cdot, \cdot)$ is the greatest common divisor. To increase the periodicity of $\mathcal{K}_\delta(\cdot, \cdot)$ while preserving a sufficient choice between short and long periods, periods in $\mathcal{P}$ are conveniently selected to be relatively prime. In this case, the period of $\mathcal{K}_\delta(\cdot, \cdot)$ is $\prod_{T_i \in \mathcal{P}} T_i$.



(a) Cross-correlation kernel ($T = 976.9$s).

(b) Kernels in Poullot *et al.* [26] ($M = 16$).

(c) LAMV ($M = 16$).

Figure 4: Comparison between a cross-correlation kernel, the temporal kernels proposed in the paper by Poullot *et al.* [26] and those learned in LAMV.

To design the set of periods, we run a coarse grid search on the Madonna dataset for video alignment. Since no feature learning is involved, findings can be applied to other video alignment datasets. Starting with a single long period ($T = 14653$ frames, equal to 977s) sufficient to cover the longest video in the dataset, we subsequently add shorter and relatively prime periods, by approximately scaling with a factor of 1.5, and test all combinations.

Figure 3 reports the localization accuracy obtained when matching each sequence in the dataset to the rest of the database. Given a query, we use the maximum of kernel scores $\mathcal{K}^*_\delta(\cdot, \cdot)$ as a global similarity score to sort the remaining videos in the database, and then select the offset with the maximum score to compute the localization error.

Starting from the longest period, as shorter periods are added, the localization accuracy increases monotonically (solid lines). On the other hand, this increases the size of the final descriptor, so we investigate the choice of a subset of periods. Using only short periods leads to precise localization and insufficient recognition (an example is reported in dashed line), while a combination of short and medium long periods provides the same performance at a fraction of the size (solid line with markers). In the rest of the experiments, we will use this combination of four periods.

Dataset	# videos	# hours	Task
Madonna	165	14.3	aligning/matching
Climbing	89	6.3	aligning
VCDB	528	27	copy detection
VCDB + 100k	100,528	2,000	copy detection
VCD	1,541	6.3	copy detection
YFCC100M	787,000	8,081	training
EVVE	2,995	166	event instance

Table 1: Characteristics of the datasets.

Discussion Figure 4 compares the temporal kernels learned by our procedure on the Madonna dataset (further details are provided in Sec. 4), with those employed in TMK [26] and with a cross-correlation kernel. We report the cross-correlation kernel using the longest period of TMK, and for an increasing number of frequencies. For $m = 64$ this has the same size as the TMK and our descriptor. While the limited number of frequencies induces of oscillations in the cross-correlation kernel, TMK avoids this phenomenon by using Von Mises kernels which have flat responses out of the target bandwidth. Kernels learned by LAMV, in contrast, have shorter periods and stronger higher-frequency coefficients, which experimentally shows to be beneficial for matching and localization.

4. Experiments

We assess the performance of the proposed method on three settings: temporal video alignment, video copy detection and event retrieval. All can be casted as joint retrieval and localization tasks, in which given a query video we want to retrieve overlapping videos, and precisely localize the query with respect to retrieved videos. In the case of *temporal video alignment*, the same action is recorded from different cameras, while in *video copy detection* the transformation matching videos is limited to 2D geometric and photometric distortions. In *event retrieval*, finally, the same event is captured in different videos which do not necessarily overlap, making this a more high level context.

4.1. Datasets

Table 1 summarizes the datasets we use. The **Madonna** dataset [7] clips are decomposed in segments, and the segments are temporally aligned on a common timeline. The image matching involves challenging viewpoint changes and wildly different frame representations. To build train and test splits, we identify the connected components inside the dataset (*i.e.* sets of sequences that overlap temporally) and build five folds which do not cross different connected components. We then use five-fold evaluation on these, and evaluate the fraction of accurately aligned videos. Similar to Madonna, the **Climbing** dataset [7] contains 89 aligned videos from a rock climbing session. It features only one connected component, therefore we use it only for testing.

The **VCDB dataset** for copy detection [19] consists of clips from sharing sites. They are all copies, possibly partial, of one of 30 source clips (Kennedy assassination, Titanic fly scene, etc.). The manual annotation gives the exact extent of the overlapping part between each pair of the clips. Most clips are quite easy to match automatically, but there are also difficult transforms like large overlays or film-from-screen copies. For evaluation, each clip is matched with all the remaining, and a segment-level version of precision and recall is computed, as defined in [19]. An additional set of 100k distractors is also provided by the same authors.

The **EVVE dataset** [28] contains clips that illustrate one out of 13 "events". The events can be news events (Flood in Thailand), or an event occurring at a specific location (Wedding of Kate and William), or a re-occurring event (eruption of the Stokkur geyser). The depictions can be exactly the same (for example, for the wedding, there is a single official video), or slightly different (different views of the same concert), or just have a common topic (the flood) that is hard to match visually. The evaluation is done with a retrieval protocol: there is a query/database split of the dataset and the result is evaluated in terms of mean average precision.

The **YFCC100M** [29] dataset is a dataset that contains 800,000 videos, whose annotations we ignore. We use it as a background set for unsupervised training.

Finally, **VCD** is a synthetic video copy dataset that we generated for training our layer on vido copy detection and event retrieval. We combined pairs of videos from from YFCC100M [29]. One of the videos is used as foreground and inserted in the other, used as background. The foreground video is clipped to a few seconds, resized and transformed geometrically (rotation, perspective transform, etc.) and photometrically (convert to gray, low-quality encoding, etc.) in various random ways. The ground-truth alignment is recorded. The data and alignment is used to train the alignment quality on an independent dataset. We split the dataset in two equal parts for training and validation.

4.2. Implementation details

The video clips are decoded at a fixed frame rate of 15 fps. As frame descriptors, we employ MultiVLAD whitened descriptors [28] and vanilla RMAC [31]. RMAC is a pooling layer that extracts bounding boxes from an arbitrary activation map in a CNN stack, and pools them into a fixed-size vector. The CNN can be fine-tuned [12], but we found that a pre-trained CNN works just as well in a context where the type of images to match is not known in advance. RMAC requires an unsupervised training phase (to find the PCA matrix), that we train on YFCC100M [29]. In preliminary experiments, we found that extracting RMAC from the 29th activation map of a Resnet-34 [15] gives the best

	@ 0.1s	@ 1s	@ 10s
Frame descriptor is MVLAD			
CTE ($m = 16$)	9.6	14.3	14.8
CTE ($m = 64$)	16.1	35.7	36.5
TMK [26]	11.7	43.5	65.2
LAMV, freq norm.	32.3	67.4	71.3
LAMV, $\tilde{V}$ norm.	40.0	74.8	76.1
LAMV, $\tilde{V}$ + freq norm.	**47.3**	**84.7**	**86.0**
Frame descriptor is RMAC			
CTE ($m = 16$)	14.0	33.8	41.0
CTE ($m = 64$)	22.1	51.4	55.0
TMK [26]	7.7	38.7	73.4
LAMV, freq norm.	28.7	57.8	66.1
LAMV, $\tilde{V}$ norm.	33.0	68.7	73.0
LAMV, $\tilde{V}$ + freq norm.	**39.6**	**76.1**	**82.9**

	@ 0.1s	@ 1s	@ 10s
Frame descriptor is MVLAD			
CTE ($m = 16$)	0.0	18.0	32.6
CTE ($m = 64$)	4.5	37.1	47.2
TMK [26]	2.2	16.9	38.2
LAMV, freq norm.	13.5	32.6	41.6
LAMV, $\tilde{V}$ norm.	19.1	51.7	61.8
LAMV, $\tilde{V}$ + freq norm.	**20.2**	**52.8**	**61.8**
Frame descriptor is RMAC			
CTE ($m = 16$)	4.4	10.1	21.3
CTE ($m = 64$)	7.9	24.7	29.2
TMK [26]	0.0	6.7	32.6
LAMV, freq norm.	7.9	19.1	25.8
LAMV, $\tilde{V}$ norm.	6.7	33.7	40.1
LAMV, $\tilde{V}$ + freq norm.	**6.7**	**34.8**	**42.7**

Table 2: Evaluation on the Madonna (left) and Climbing (right) datasets for temporal video alignment. The evaluation measure the percentage of queries localized better than a threshold (0.1s, 1s, 10s).

	F1 score
Temporal Hough voting (SIFT+BoV) [19]	55.0
Temporal network (SIFT+BoV) [19]	60.0
Temporal network (AlexNet) [20]	65.0
TMK (RMAC) [26]	67.4
LAMV, freq norm.	62.8
LAMV, $\tilde{V}$ norm.	60.0
LAMV, $\tilde{V}$ + freq norm.	**68.7**

Table 3: Evaluation on the VCDB dataset for video copy detection. The evaluation measure is the maximum F1 score on segment-level precision and recall measures [20].

matching results, so we keep this setting throughout. We also tested with C3D features [32]. The localization and retrieval accuracy was not satisfactory with these techniques.

We build mini-batches with 128 triplets. We combine the retrieval loss L_r and the localization loss L_l, respectively, with weights 1/4 and 3/4. The retrieval margin m_r is set to 0.01, and the localization margin m_l to 0.001. The radius r is set to 1s. We train the network using SGD with Nesterov momentum 0.9 and a learning rate of 0.001.

The set of periods $\mathcal{P}$ is set to $\{9767, 2731, 1039, 253\}$, which, in seconds, correspond to $\{651\text{s}, 182\text{s}, 69\text{s}, 17\text{s}\}$. When computing the TMK and the LAMV descriptor, the number of frequencies M is always set to 16, so to have comparable descriptor sizes.

4.3. Experimental results

Video alignment. We assess the localization and retrieval performances of our model on temporal video alignment by learning on Madonna with five-folds evaluation, and using MVLAD and RMAC descriptors. For each fold we use each sequence in the test set as query against the remaining sequences in the same set. As in Section 3.4, we use the maximum of kernel scores to sort the set, and then select the offset with maximum score from the first retrieved sequence.

We compare LAMV against our reimplementations of TMK [26] and CTE [28] with 16 and 64 frequencies. The size of our descriptor is equal to that of TMK and of CTE with 64 frequencies. Table 2 reports the localization errors: our model attains the best localization accuracy using both descriptors, both for low and high localization errors in comparable settings. To validate the two stage normalization proposed in Section 3.1, we also show the performance of LAMV when applying both or only one of the two normalizations. Using the combined normalization helps to localize videos with greater accuracy, and to enhance the retrieval capabilities of the layer, as testified by the increased localization at higher thresholds.

We investigate the generality of the models learned for temporal video alignment by testing each of them on the Climbing dataset, which contains a different scenario. Averaged results are reported in Table 2 (right). Our method obtains a higher localization accuracy and retrieval performances when compared to the same baselines, and the effectiveness of the two-step normalization is confirmed also in this setting. In Figure 5 we report a sample of challenging sequences taken from different point of views that are correctly aligned by our method.

Video copy detection. For video copy detection, we train on VCD using RMAC features, which show good invariance to copy detection transformations, and test on the recent VCDB dataset. Results are reported in Table 3. We compare with our reimplementation of TMK, and with three state of the art proposals for copy detection: the temporal Hough voting and the temporal network proposed in [19] on local SIFT descriptors, and temporal network us-

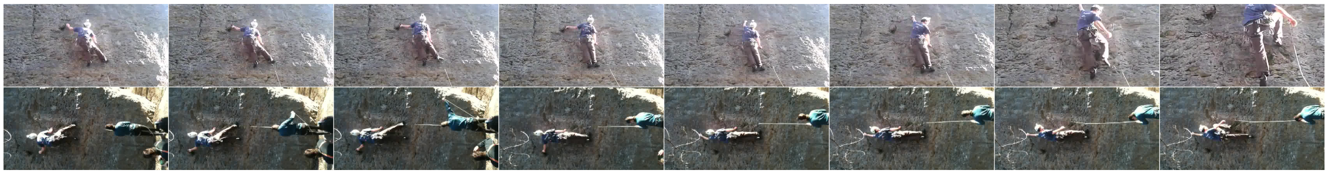


Figure 5: Examples of a sequence correctly aligned by LAMV on the Climbing dataset. Each column corresponds to temporally aligned frames (2 frames per second are represented).

Method	mean mAP	per category												
TMK [26]	51.6	65.9	37.5	13.2	43.9	36.3	28.7	22.6	14.4	16.8	29.7	23.6	86.2	65.9
LAMV	53.6	71.5	38.3	15.8	46.1	38.7	27.7	24.7	13.8	22.2	27.3	27.3	90.8	69.1
LAMV + QE	58.7	83.7	50.0	12.6	58.8	45.5	34.3	26.7	14.2	23.0	29.3	21.6	95.0	77.6

Table 4: Evaluation for event retrieval (mAP on EVVE). The ordering of categories is the same as in the EVVE paper [28].

Method	Localization	mAP
Frame descriptor is MVLAD		
MMV [28]		33.4
CTE [28]	✓	35.2
Stable hyperpooling [6]		36.3
TMK [26]	✓	33.5
Frame descriptor is RMAC		
Mean RMAC		52.9
TMK [26]	✓	51.6
LAMV, freq norm.	✓	53.5
LAMV, $\tilde{V}$ norm.	✓	51.9
LAMV, $\tilde{V}$ + freq norm.	✓	**53.6**
CGA [11] (AlexNet+ResNet)		52.3
Average query expansion ($N_1 = 10$)		
Stable hyperpooling [6] (MVLAD)		38.9
CGA [11] (AlexNet+ResNet)		58.5
LAMV (RMAC)	✓	**58.7**

Table 5: Comparison with the state of the art for event retrieval (mAP on EVVE).

ing AlexNet features [20]. Temporal Hough voting aligns matched frames by means of a temporal Hough transform, while the temporal network uses a network flow optimization strategy. They both require to store frame-level descriptors for matching videos. LAMV attains the best F-Score reported on this dataset, and features a fixed-size video descriptor, independent on the video length. When testing with the large number of distractors from the VCDB+100K set, however, we observed that the performance of the temporal network [20] is still higher (58.9 vs 49.3 F1), even though LAMV outperforms TMK also in this setting (49.3 vs 35.5 F1).

Event retrieval. Finally, we apply our approach on event retrieval. We compare against the Mean-MultiVLAD (MMV), obtained by averaging and ℓ_2-normalizing Multi-VLAD frame descriptors, CTE [28], Stable hyperpooling [6] and the recent Counting Grid Aggregation (CGA) [11]. LAMV, CTE and TMK are able to provide a good localization in addition to retrieval, the others can not. To factor out the impact of the raw frame descriptor, we also report the values obtained by using the ℓ_2-normalized mean RMAC descriptor, and run our reimplementation of TMK on RMAC features. As shown in Table 5, our method outperforms all the baselines it has been compared to, including CGA and TMK. We also evaluate the performance of LAMV when using average query expansion (AQE) [6]. In this setting, the top N_1 results are averaged and then to produce an augmented query, which is then used for retrieval. Overall, our methods attains the best result reported on this dataset without query expansion and with AQE.

End-to-end training and performance. We tested end-to-end training of the architecture. In practice it did not give a significant improvement. This observation is common with videos, and can be explained by (a) the lack of real data for these tasks (feature learning is limited with artificially copied sequences), and (b) by the structure of TMK and RMAC which creates complex path of gradients, as also observed in prior works [12]. The matching, for each δ hypothesis, requires the computation of an inner product between frame-level features, which is comparable to CTE. In terms of memory consumption, LAMV is $|\mathcal{P}| = 4$ times larger than CTE if using the same number of frequencies, but provides a significant localization accuracy boost.

5. Conclusion

We presented a learnable descriptor based on temporal match kernels. It can be learned with a triplet loss function designed to improve its performance when comparing and temporally aligning videos. Experimental results, conducted on temporal video alignment, video copy detection and event retrieval, show that our approach beats the state of the art on all three tasks with a significant margin.

References

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. YouTube-8M: a large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] A. Bursuc, G. Tolias, and H. Jégou. Kernel local descriptors with implicit rotation matching. In *Proceedings of the ACM International Conference on Multimedia Retrieval*, 2015.

[3] O. Chum. Low dimensional explicit feature maps. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[5] A. Diba, V. Sharma, and L. Van Gool. Deep temporal linear encoding networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2017.

[6] M. Douze, J. Revaud, C. Schmid, and H. Jégou. Stable hyper-pooling and query expansion for event detection. In *Proceedings of the IEEE International Conference on Computer Vision*, December 2013.

[7] M. Douze, J. Revaud, J. Verbeek, H. Jégou, and C. Schmid. Circulant temporal encoding for video retrieval and temporal alignment. *International Journal of Computer Vision*, 119(3):291–306, 2016.

[8] B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, and T. Tuytelaars. Rank pooling for action recognition. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[9] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5378–5387, 2015.

[10] B. Fernando, S. Shirazi, and S. Gould. Unsupervised human action detection by action matching. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017.

[11] Z. Gao, G. Hua, D. Zhang, N. Jojic, L. Wang, J. Xue, and N. Zheng. Er3: A unified framework for event retrieval, recognition and recounting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2253–2262, 2017.

[12] A. Gordo, J. Almazan, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. In *Proceedings of the European Conference on Computer Vision*, 2016.

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[16] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *Proceedings of the European Conference on Computer Vision*, October 2012.

[17] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 2015. to appear.

[18] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of the European Conference on Computer Vision*, October 2008.

[19] Y.-G. Jiang, Y. Jiang, and J. Wang. VCDB: a large-scale database for partial copy detection in videos. In *Proceedings of the European Conference on Computer Vision*. Springer, 2014.

[20] Y.-G. Jiang and J. Wang. Partial copy detection in videos: A benchmark and an evaluation of popular methods. *IEEE Transactions on Big Data*, 2(1):32–42, 2016.

[21] K. Kang, W. Ouyang, H. Li, and X. Wang. Object detection from video tubelets with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 817–825, 2016.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, December 2012.

[23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[24] A. Miech, I. Laptev, and J. Sivic. Learnable pooling with context gating for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2017.

[25] K. Papoutsakis, C. Panagiotakis, and A. A. Argyros. Temporal action co-segmentation in 3d motion capture data and videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[26] S. Poullot, S. Tsukatani, A. P. Nguyen, H. Jégou, and S. Satoh. Temporal matching kernel with explicit feature maps. In *ACM International conference on Multimedia*, 2015.

[27] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, 2008.

[28] J. Revaud, M. Douze, C. Schmid, and H. Jégou. Event retrieval in large video collections with circulant temporal encoding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2013.

[29] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

[30] G. Tolias, T. Furon, and H. Jégou. Orientation covariant aggregation of local descriptors with embeddings. In *Proceedings of the European Conference on Computer Vision*, September 2014.

[31] G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of CNN activations. In *Proceedings of the International Conference on Learning Representations*, 2016.

[32] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497, 2015.

[33] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 34:480–492, March 2012.

[34] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4507–4515, 2015.