Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2022) 1–8

Procedia Computer Science

Periodicity Counting in Videos with Unsupervised Learning of Cyclic Embeddings

Nicolas Jacquelin[(1,2)], Romain Vuillemot[(1)], Stefan Duffner[(2)]

(1) École Centrale de Lyon, LIRIS
(2) INSA Lyon, LIRIS
{*nicolas.jacquelin, romain.vuillemot, stefan.duffner*}*@liris.cnrs.fr*

Abstract

We introduce a context-agnostic unsupervised method to count periodicity in videos. Current methods estimate periodicity for a specific type of application (*e.g.* some repetitive human motion). We propose a novel method that provides a powerful generalisation ability since it is not biased towards specific visual features. It is thus applicable to a range of diverse domains that require no adaptation, by relying on a deep neural network that is trained completely unsupervised. More specifically, it is trained to transform the periodic temporal data into some lower-dimensional latent encoding in such a way that it forms a cyclic path in this latent space. We also introduce a novel algorithm that is able to reliably detect and count periods in complex time series. Despite being unsupervised and facing supervised methods with complex architectures, our experimental results demonstrate that our approach is able to reach state-of-the-art performance for periodicity counting on the challenging QUVA video benchmark.

Keywords: unsupervised learning, periodicity, repetition, embedding, triplet loss

1. INTRODUCTION

We define periodicity as any phenomenon that happens multiple times in a similar way over time. Periodicity is ubiquitous in real-world scenes and occurs at multiple scales. In elite sports, the tracking of the athletes' motion is a key issue and is highly repetitive. In swimming, in particular, the stroke pace is one of the most important metrics to determine a race quality and infer other statistics (*e.g.* stroke amplitude, rate etc.). But this task is challenging for many reasons. First, two successive repetitions may significantly differ (*e.g.* swimming strokes rate and amplitude change during the race, as well as the swimmer's position with respect to the camera). Second, the precise beginning and end of a cycle is ambiguous. Finally, there exist other artifacts, such as the different sub-cycles that may be mistakenly detected as cycles. Furthermore, the notion of periodicity is context-dependant: the same event in two different sequences might be periodic or not depending on whether it is repeated or not. Therefore, the signal must be studied globally and not frame-wise.

Estimating periodicity is particularly challenging with videos recorded under unconstrained conditions. Any spacial shift, background noise or viewpoint change result in important variations in the captured signal, which often makes it hard to automatically detect the dominant cycle. Although these problems can be alleviated with recent machine learning methods based on Convolutional Neural Networks (CNN) that are capable of extracting noise-robust abstract representations of an image [1], those deep neural network models often require large amounts of training data [2, 3, 4]. This issue is often circumvented by pre-training such networks on large annotated datasets [5, 6], but then the model may be biased towards specific visual features which may not be relevant for the task at hand and thus lead to a lower performance [7, 8]. To tackle the periodicity counting problem in videos, state-of-the-art methods [9, 10, 11, 12] are trained on Kinetics [13], a videos dataset of persons doing repetitive actions. Such pre-trained models are thus domain specific to human gestures, and their performance are likely to drop when used on less frequent domains, such as astrophysics or medical videos. Thus, a new dataset is required to adapt the model, which is extremely time consuming and costly. Moreover, not all periodicity problems concern regular videos of human activity: there are other types of complex time series, like multi-source sensors monitoring air quality or biophysical activities [14, 15, 16], and 4D MRI videos (*i.e.* 3D images through time) of breathing lungs [17], active brains [18] or beating hearts [19]. For these reasons, it is important to have a domain-agnostic method.

This paper presents such a technique introducing a specific training method adapted for temporal periodic data in general.

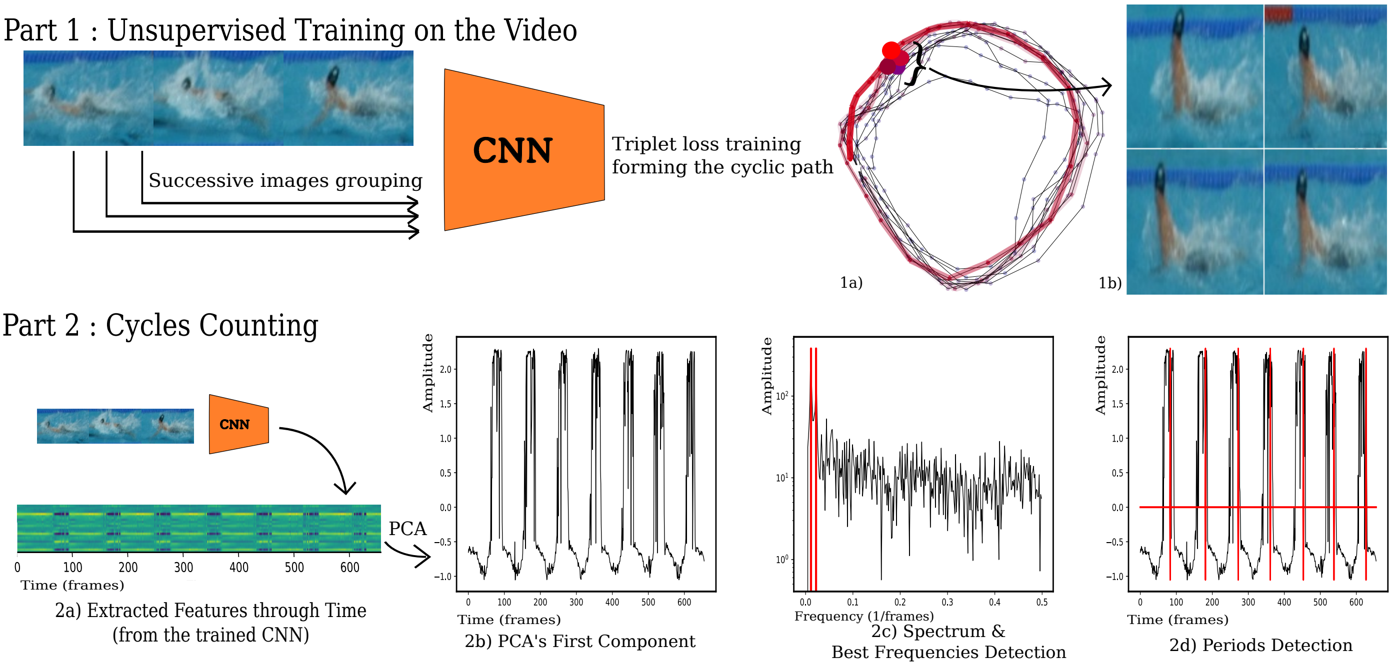

Figure 1. The framework introduced in this paper. In Part I, a CNN is first trained in an unsupervised way on the data to analyze, as described in Section 3.1. Then, it is used to extract an embedding for each image of a video. (1a) shows an example of the 2D PCA projection of these embeddings. The last 50 embeddings are linked chronologically (in red), revealing the cyclic path. (1b) shows the input images whose embeddings correspond to the highlighted points in (1a). All of them belong to different cycles but correspond to the same phase in the cycle, therefore the points are close in the latent space. In Part II, we chronologically concatenate these embeddings to form a multi-variate signal. It is then reduced into a uni-variate temporal signal with PCA keeping only the first component. Finally, our *Max Detector* algorithm is used to count the cycles on the signal, which corresponds to the number of cycles on the video. Best seen in color.

With an adapted neural network architecture, it could even be used outside of the video domain to study other types of multivariate time-series.

Our approach is summarized in Figure 1. It reduces a video into a periodic 1D signal with an original deep learning method and counts its repetitive patterns using a novel peak detection algorithm based on various signal processing techniques. This counting process is performed in a single step. It does not require to test different time-scales, or to use a sliding window through the whole signal to process it completely. The computational cost is therefore greatly reduced compared to other methods based on transformer architectures [9] or multimodal fusion models [11]. Our main contributions are the following:

- An unsupervised method to train a neural network with the triplet loss to encode any kind of video (Section 3.1).
- An algorithm to count the periodic patterns in time-series (Section 3.2).
- A framework combining these algorithms for automatic periodicity counting in videos, based on the analysis of a learnt embedding.

2. RELATED WORK

To analyse videos recorded under unconstrained conditions, recent approaches use CNNs, as they are the current state of the art for image classification [20], action recognition [21], objects tracking in videos [22] and saliency detection [23]. They are also used in periodicity *detection* [24, 9], which is very similar to periodicity *counting*: the first classifies each frame of a video as periodic or not (the PERTUBE dataset [24] typically is used as a benchmark), whereas the latter operates on a periodic video and counts the repetitions.

To specifically address periodicity counting in daily life videos, Levy and Wolf [25] proposed a 3D CNN architecture: the input is composed of 20 chronologically ordered images, each separated by N frames in the timeline. In this way, the temporal information is integrated into the input. They trained the model in a supervised way on synthetic data to separate the sequences on their temporal dimension. This feature-oriented method is robust to colour and lighting variations, but one needs to test several timescales (*i.e.* many different values of N) in order to obtain good results. Also, as for supervised trained models, the performance directly depends on the dataset size and quality.

Similar to our method, other works aim to reduce a video to a one-dimensional signal. Polana and Nelson [26] detected the pixels responsible for motion, and considered them as temporal signals varying throughout the video. They extracted a signal period by detecting the peaks on its Fourier Transform. Yang, Zhang, and Peng [27] used a method based on pixel-wise joint entropy to estimate the similarity between a reference image and the other ones, resulting in a 1D temporal function.

Runia *et al.* [28], introduced another method to convert a video into a 1D signal. They studied the main direction of the foreground's optical flow in order to create multiple 1D signals from its directional gradient components through a wavelet transform. Their paper also introduced the QUVA benchmark dataset for periodicity counting in everyday videos.

More recently, Dwibedi *et al.* [9] proposed a complex architecture mixing CNNs and transformers [29], trained in a fully-supervised fashion on the Countix dataset which they introduced themselves. In their experiments, they also trained their model on a considerable amount of synthetic data obtaining impressive results, but unfortunately they did not publish this dataset. This method achieves good results on public benchmarks, but it is by far the most computationally expensive and data dependant. Using the Countix dataset, Zhang *et al.* [11] proposed a multi-modal approach relying on sound and sight to improve the state-of-the-art on the Countix benchmark. They did no evaluation it on QUVA, however.

The work of Yin *et al.* [12] shares some similarities with our work, as it also extracts periodic features from a video with a learning-based method, reduces it to a 1D signal, and counts the repetitions with an algorithm relying on the Fourier transform. However, their approach is not generalizable to other types of data since it uses a neural network that is pre-trained on a large annotated video dataset (Kinetics [13]) in a supervised way. As such, they can only analyze conventional videos of 2D images and the learnt visual features are domain dependant, which may not give satisfactory results on other types of videos. In addition, the signal processing part of their method is quite different from ours. To detect the dominant frequency, it uses a specific multi-threshold filter in the frequency domain with several empirically determined thresholds, and then detects the peaks in the reconstructed signal with the inverse Fourier transform. Our model is trained unsupervised and end-to-end, and our robust peak detection algorithm operates on the original 1D signal obtained from PCA.

Zhang *et al.* [10] proposed an approach based on a context-aware model. However, it is not designed to generalize to unseen domains: the method uses the Kinetics dataset [13], where a separate model is trained for each sports type resulting in excellent overall scores on public benchmarks. Finally, the work of Feirrera *et al.* [30] is also context-specific: it uses human pose classification to count repetitions of workout routines. This approach is suited but limited to the context of human motion repetition counting.

As most of these methods ([9, 11, 10, 12, 30]) are trained on a human motion video dataset (Countix being built on top of Kinetics), they are well adapted to human gestures and actions. However, this makes them (i) specific to videos and not any other type of input data and (ii) biased towards human motion. On the contrary, we designed our method to be applicable to any type of periodic data.

3. UNSUPERVISED PERIOD COUNTING

We introduce a novel unsupervised learning process, illustrated in Figure 1 Part 1, to encode a video in a way that highlights its periodic features. To that purpose, a CNN is trained directly on the video to be analyzed. The resulting video embedding is a periodic signal that is processed by a novel algorithm to count its cycles.This new method does not follow the

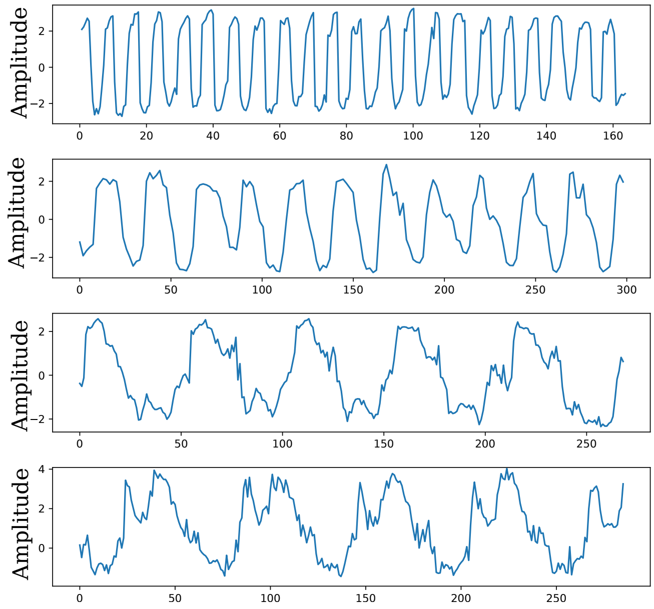

Figure 2. Examples of 1D PCA projections of embeddings. The first three rows show the result for different cycles durations, from 8 to 50 frames per cycle (on average). The last row shows a more complex pattern containing 2 distinct local maxima. In such cases, our *Max Detector* could count 2 cycles per pattern, resulting in a false result, like mentioned in Section 4.3.

classical training/validation/test protocol. The different steps of the pipeline are describes in detail in this section.

3.1. Latent Representation Learning

Before the model can be trained, one needs to group successive frames from the video. The frame at time index t is grouped with the frames $t+1$ and $t-1$ forming a triplet. Each frame belongs to 3 different groups (triplets) where it plays the 3 roles $t-1$, t and $t+1$, except for the first and last frames (because there is respectively no frame before it to be $t-1$ and no frame after it to be $t+1$). With T frames in the video, there are $T-2$ triplets in the end.

The output vector of the image at time index t is called $\phi(t)$. The images need to be embedded by the CNN in such a way that, in chronological order, they form a repetitive pattern in the latent space, *i.e.* a loop. This is achieved by using a continuity criterion and a periodicity criterion. The first forces the images' successive embeddings to be temporally ordered along a pseudo-linear path. The latter forces this path to contain repetitive patterns.

To guarantee the continuity criterion, the triplet loss is used as objective function:

$$L(A, P, N) = \max(0, |\phi(A) - \phi(P)| - |\phi(A) - \phi(N)| + \alpha)\,, \quad (1)$$

where $\alpha \in \mathbb{R}$ is the margin, A is the anchor, P is the positive and N is the negative image. The purpose of the triplet loss is to make the distance between the embeddings of A and N larger than the distance between the ones of A and P up to a minimum distance defined by α. Our approach defines the image at time index $t-1$ as the anchor, t as the positive and $t+1$ as the negative. The overall consequence of applying this training method to

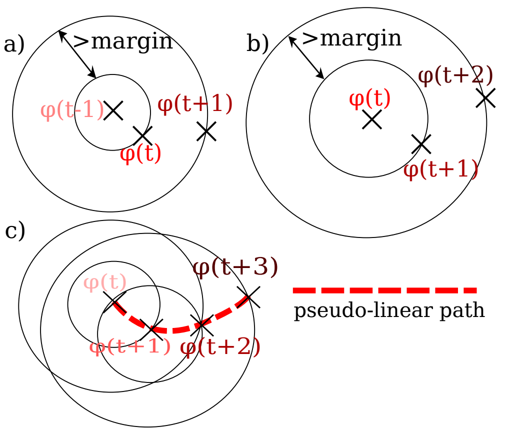

Figure 3. Unsupervised learning of the pseudo-linear path using the Triplet Loss. The anchor is at the center, the positive is on the smaller circle (not necessarily the same size each time), and the negative is outside of the bigger circle. a) The anchor is $\phi(t-1)$: $\phi(t)$ and $\phi(t+1)$ are separated. b) The anchor is $\phi(t)$: $\phi(t)$ and $\phi(t+1)$ are drawn together. When the training starts, the negative can be at the other side of the big circle compared to the positive. But this situation is no longer possible when the constraint is applied to all the successive frames, as shown in c), after convergence: a pseudo-linear path is naturally formed, as it is the only way to respect the constraints imposed by the loss. Best seen in color.

each value of t in the video is that each $\phi(t)$ is "pulled towards" its direct neighbors ($\phi(t-1)$ and $\phi(t+1)$), and "pushed away" from its 2^{nd} degree neighbors ($\phi(t-2)$ and $\phi(t+2)$). Therefore, the positive embedding is "placed" between the anchor and the negative one, with a tolerance of α, as explained in Figure 3. This forces the creation of a pseudo-linear path chronologically aligning the embeddings in the latent space.

To guarantee the periodicity criterion we rely on the property of CNNs that two similar inputs will have similar outputs unless explicitly trained otherwise [31]. With periodic videos, if one cycle has a period T, then the images at time indexes t and $t+T$ will have the same phase in the cycle and look alike. Therefore, the images have an embedding close to the other images corresponding to the same phase in the cycle. This cyclic behavior is illustrated in Figure 1, images 1a) and 1b).

The resulting model closely fits the data it was trained on. Therefore, to get the most adapted latent space representation for a video, a model needs to be specifically trained on it (and no other videos). This requires some training time, but, as explained in Section 4.1, it is not too expensive.

The training process has been presented using frames as a temporal unit, but it can be enriched by other information. In Section 4.2, we show that adding the optical flow to a frame gives better results (*i.e.* frame t is enriched with the optical flow between frames t and $t+1$). In this case, we concatenate the 3 image channels (RGB) to the 2 optical flow channels (direction & magnitude) resulting in $5 \times W \times H$ temporal unit tensors (W and H being the width and height of the video). This section presented a way to fit a latent space to a video, but it also works for other complex time series. Similarly to adding the optical flow, which is the variation of a frame with respect to the next one, one could add the gradient between successive temporal vectors to augment the information encoded by the model.

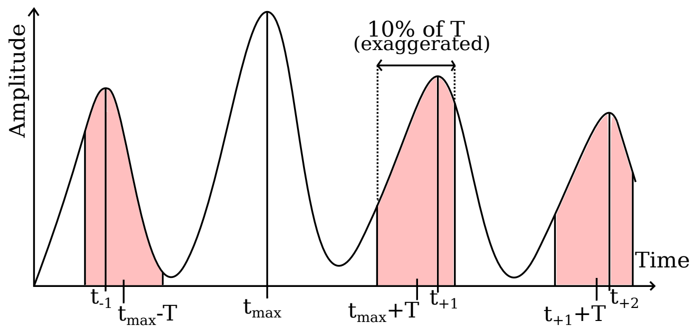

Figure 4. Illustration of *Max Detector*. Starting from the global maximum's index t_{max}, the algorithm shifts by one period T and finds the maximum's index t_{+1} in a window of 10% of T (in red, exaggerated for a better understanding). This window makes *Max Detector* robust to period variations. Starting from t_{+1}, this is repeated to find t_{+2}, t_{+3} and so on until reaching the signal's end. A first iteration goes from t_{max} to the end of the signal and a second from t_{max} to $t = 0$. Best seen in color.

3.2. Cycle Counting

After training, the images in the video are embedded in the latent space in such a way that they form a cyclic pattern. The next step, illustrated in Figure 1, Part 2, is to count these cycles.

In order to effectively work in the frequency domain and apply common signal processing techniques, the model's output vectors have to be transformed into a one-dimension signal. To do so, the embedding vectors of the M images are chronologically "stacked" to form a matrix like in Figure 1, 2a). This is, if the latent space has D dimensions, the resulting matrix is of size $D \times M$. A PCA projection is applied to the matrix in order to keep the features combination with the most importance. By only keeping the 1^{st} element of the PCA, it results in a $1 \times M$ temporal signal S with periodic information, *i.e.* a recurring pattern like in Figure 2, corresponding to a repetition in the video.

The subsequent algorithm uses the Fourier Transform to detect the signal's F main frequencies. These candidate frequencies will all be tested by our proposed algorithm named *Max Detector* explained in the following.

The main goal of *Max Detector* is to detect the maximum of each cycle in S, and to save their time indices in a list named *MaxList*. These maxima will be used to distinguish and count the cycles. We name f_i the current analyzed frequency (one of the F detected by the Fourier transform), *Max List*$_i$ its corresponding maxima list, and T_i its corresponding period. *Max Detector* starts by finding the signal S global maximum's time index, which is added to *Max List*$_i$. We suppose the neighbour cycle maxima are approximately one period away from each other. Therefore, to find the next maximum, one creates a time window by shifting of $T_i \pm 10\%$ from the current maximum. In this window, the local maximum is located and its time index is added to the list *Max List*$_i$. This operation is performed again from this new local maximum, until reaching the signal's edge. This procedure is repeated twice, each time starting from the global maximum: once forward towards the end, and once backward to the beginning of the signal. This is graphically explained in Figure 3.1 and formally explained in Algorithm 1.

Algorithm 1 Max Detector: creation of candidate lists $MaxList_m$

Require: signal S, $f_m, m \in (1, ..., F)$
  $MaxList = \emptyset$
  for m in (1, ..., F) **do**
    $MaxList_m = \emptyset$
    $T_m = 1/f_m$
    $t_0^{max} = \text{argmax}\, S(t)$
    $MaxList_m \leftarrow MaxList_m \cup t_i^{max}$
    $t_i^{max} = t_0^{max}$
    while $t_i^{max} - T_m \geq 0$ **do**
      $t_i = t_i^{max} - T_m$
      $W_i = (t_i - 0.1 \cdot T_m\ ,\ t_i + 0.1 \cdot T_m)$
      $t_i^{max} = \text{argmax}_{t \in W_i}\, S(t)$
      $MaxList_m \leftarrow MaxList_m \cup t_i^{max}$
    end while
    $t_i^{max} = t_0^{max}$
    while $t_i^{max} + T_m < length(S)$ **do**
      $t_i = t_i^{max} + T_m$
      $W_i = (t_i - 0.1 \cdot T_m\ ,\ t_i + 0.1 \cdot T_m)$
      $t_i^{max} = \text{argmax}_{t \in W_i}\, S(t)$
      $MaxList_m \leftarrow MaxList_m \cup t_i^{max}$
    end while
    $MaxList \leftarrow MaxList \cup MaxList_m$
  end for
  return $MaxList$

Once the F different frequencies have been processed, there are F different candidate lists *Max List*$_i$. Each list is evaluated individually and the best solution is retained. To evaluate a *Max List*$_i$, each of its local maxima will be compared to their local region accordingly to equation 2. This score computes the proportion of elements in *Max List*$_i$ that correspond to the local maximum in half a period centered on them.

$$Score\_i = \frac{1}{L_i} \sum_{k}^{Max\_List\_i} \left[S[k] = \max\left(S[k - \frac{T}{4} : k + \frac{T}{4}] \right) \right] , \tag{2}$$

L_i being the number of elements in *Max List*$_i$ (*i.e.* its length), k representing the different local maxima indices. As a result, a list that contains each and every local maxima of the signal separated by approximately T has a score of 1. On the contrary, the more incorrect maxima a list contains, the lower its score is.

The list with the highest score is kept , whose number of elements represent the number of cycles in the signal and therefore the number of repetitions on the video.

4. EXPERIMENTS AND RESULTS

To compare our method with the current state of the art, we used the QUVA [28] and Countix [9] benchmarks. QUVA is composed of 100 videos showing between 4 and 63 repetitions. The videos are very diverse and recorded in real-life situations, often with camera motion and background variation. Countix contains a similar visual variety. It is the first large video repetition dataset, containing more than 8000 clips showing 2 to 73 repetitions. The metrics used for performance comparison are the Mean Absolute Error (MAE) and the Off-By-One Accuracy (OBOA), defined as:

$$MAE = \frac{1}{N} \sum_{i}^{N} \frac{|c_i - \hat{c}_i|}{c_i} \qquad OBOA = \frac{1}{N} \sum_{i}^{N} [|c_i - \hat{c}_i| \leq 1] ,$$

where c_i is the true count and $\hat{c}_i$ is our model estimation on the same video i and N is the number of videos in the dataset. The OBOA, introduced in [28], counts the proportion of correct predictions with a tolerance of 1. This margin serves to reduce the importance of rounding mistakes, as ambiguous cycle cutoffs can happen at both ends of the video.

Each model was trained independently on one video at a time. This means that for a dataset of 100 videos like QUVA, 100 different models have been trained and evaluated for each experiment (except said otherwise). The following sections describe the experiments performed on the two benchmarks and the results obtained with the two metrics.

4.1. CNN Architecture

During our test phase, we did not notice a significant difference of performances using different CNN architectures (we tried VGG19 and VGG11 [32], results shown in Table 2). We also designed a straightforward CNN model with fewer layers than VGG11 as it would train better on the few images of the video clips. Our custom model is composed of 6 layers of 3×3 convolutions with ReLU activation [33], each layer doubling the number of filters (starting at 4, finishing at 128) and 2×2 max pooling [34] after each layer, and a final global average pooling giving a 32 dimensions output vector.

For each study, we trained a model for 30 epochs with a batch size of 16, a learning rate of 10^{-3} and the Adam optimizer [35]. Under these conditions, the training took about 1.1 times the total duration of a video using a NVIDIA GTX 1080 GPU.

4.2. Ablation Study

Our initial baseline CNN model just takes one image as input (Variation "1 img" of Table 2). To improve performances, we enriched the input with the optical flow between two consecutive frames, similar to Zhou *et al.* [37], as mentioned in Section 3.1. The new input is therefore made of an image concatenated with the optical flow from this image to the next one. This variation is named "flow" in Table 2.

To show the importance of our training policy, we used common CNN models trained on Imagenet [2] to do the embedding, with only one image as an input, as required by these architectures (they were not retrained on the cyclic videos images). The obtained embeddings did not give easily exploitable cyclic curves, resulting in bad performance. With our training policy, however, the different CNN architectures all reached comparable results, our shallow model being better than the deeper ones. For all lines in Table 2 not stating a specific architecture, we used our custom shallow CNN.

In the *Max Detector* algorithm, we compare F different frequencies. As shown in Table 2, we studied the performance obtained for different values of F. The QUVA benchmark does not

Table 2. Results of different variations of our approach on the QUVA dataset. Pretrained models did not perform well at embedding the images in a cyclic manner. The same architectures, trained using our method, give much better results. Different architectures do not change the results.

Variations	MAE$\pm\sigma\downarrow$	OBOA$\uparrow$
1 img + F=4	0.388± 0.512	0.43
VGG19 (pretrained) + F=4	0.758± 0.812	0.21
VGG11 (pretrained) + F=4	0.783± 0.761	0.17
VGG19 + flow + F=4	0.252± 0.400	0.60
VGG11 + flow + F=4	0.241± 0.367	0.62
flow + F=2	0.291± 0.445	0.59
flow + F=5	0.239± 0.335	0.62
flow + F=7	0.244± 0.328	0.61
flow + F=10	0.378± 0.710	0.57
flow + Scholkmann *et al.* [38]	0.307± 0.408	0.51
flow + F=4	**0.231± 0.326**	**0.64**

provide a specific evaluation protocol, so we used cross validation on QUVA with 50/30/20 splits (*i.e.* random splits with said sizes were created to evaluate different values of the parameter F without changing anything else, in particular the temporal input signal). The results were the same for the different splits: between 4 an 7, F seems to have little impact on the result, $F = 4$ being the optimum. On the other hand, Countix has a training dataset, which we used to compute the best value for F. The results were similar between 2 and 7 again, obtaining an optimum for F=2.

Finally, we measured the importance of *Max Detector*, so we used another automatic peak detection algorithm, described in [38] by Scholkmann *et al.* It counts the cycles of the same signal as our *Max Detector*, but performs significantly worse. This shows the effectiveness of our algorithm and the importance of a more specialized algorithm for periodicity counting.

4.3. Results and Discussions

Table 1 shows the results compared to other supervised and unsupervised methods. On QUVA, our model has the best MAE and OBOA of all the unsupervised methods. This is achieved with no prior bias or complex model, which demonstrates the efficiency of our framework. Moreover, even compared to supervised models, it is outperformed by only one model with a small margin.

Regarding Countix, we would like to highlight a few major weaknesses of the dataset. First, many clips with only 2 repetitions are cutting out parts of the periodic actions (at the start or the end of the video), resulting in no fully repeated movement. Moreover, the shortest video is 0.2s, which corresponds to 6 frames at 30 fps. In our opinion, such video clips are too short to contain distinct repetitions. In addition, the choice to keep the same train/validation/test splits as originally in Kinetics seems questionable, each action category being represented in both the train/validation set and test sets. To create a more context agnostic dataset, it would be preferable to have specific test categories missing from the train/validation split to challenge the generalisation of the method. On Countix, our unsupervised method gives an OBOA better than Zhang *et al.* [11] and is only outperformed by Dwibedi *et al.* [9]. The MAE is slightly worse than the supervised methods, but not by a big margin. In fact, the difference between our score and Dwibedi *et al.*'s equals the difference between them and Zhang *et al.*

In addition, we observed a behavior in most of the “OBOA failure” cases (*i.e.* where $|c_i - \hat{c}_i| \geq 2$). Our *Max Detector* sometimes counts 2 repetitions instead of 1 for each cyclic pattern, therefore doubling the prediction compared to the ground truth. Indeed, a lot of ambiguity in the cycles count exist, the most usual being the “double action” that can be counted as either one or two periods. For instance, on a freestyle swimming clip, the annotated ground truth cycle can either be one “left and right arm movement” or only one “arm movement” depending on the labeller. Such ambiguity can often not be managed by context-agnostic methods, which will “guess” the answer between N and $2 \times N$ cycles when it occurs. This partly explains the difference between our score and supervised method's score, which are specifically trained to correctly choose in these ambiguous contexts. This problem artificially increases the MAE in an “unsymmetrical” way. If the truth is 10 repetitions, but our model gives 5, $MAE = 0.5$. If it is the opposite, $MAE = 2$. We could use the Normed MAE (NMAE) as a new metric, as it does not cause this “unsymmetrical” issue:

$$NMAE = \frac{1}{N} \sum_{i}^{N} \frac{|c_i - \hat{c}_i|}{max(c_i, \hat{c}_i)}$$

On QUVA and Countix, the NMAE of our method is respectively 0.158 and 0.345.

Table 1. Results for different methods of periodicity counting methods. Bold: the best result of a category. Underlined: the second best. Our unsupervised method reaches comparable performances to the best fully-supervised models. This proves the overall interest of our method.

Method	Unsupervised	QUVA : MAE$\pm\sigma\downarrow$	QUVA : OBOA$\uparrow$	Countix : MAE$\pm\sigma\downarrow$	Countix : OBOA$\uparrow$
Levy and Wolf [25]		0.482± 0.615	0.45	-	-
Yin *et al.* [12]		**0.199** ± <u>0.335</u>	-	-	-
Dwibedi *et al.* [9]		0.322	**0.66**	<u>0.364</u>	**0.697**
Zhang *et al.* [11]		-	-	**0.307**	0.511
Pogalin *et al* [36]	✓	0.389± 0.376	0.49	-	-
Runia *et al* [28]	✓	0.232± 0.344	0.62	-	-
Our method, F=4	✓	<u>0.231</u>± **0.326**	<u>0.64</u>	0.495 ± 0.769	0.517
Our method, F=2	✓	0.291± 0.445	0.59	0.419 ± 0.496	<u>0.545</u>

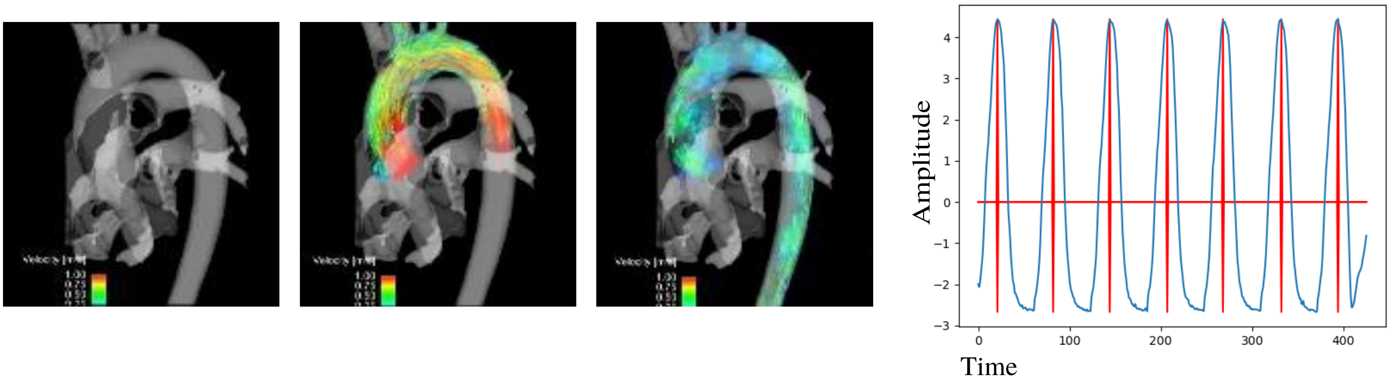

Figure 5. 4D MRI video analyzed by our method. This is a proof of concept of the method's generalisation to different input types. Left: 2D slices of 3D input images (for display purpose) at different moments. The blood pulses through the artery. Right: the 1D PCA (blue) and peak detection of our model (red). As MRI contain very little noise, the periodic pattern is perfectly smooth. Better seen in color.

4.4. Application to 4D videos

Many applications in medical imaging deeply rely on 4D videos (*i.e.* 3D images through time), acquired with Magnetic Resonance Imaging (MRI) for instance. However, state-of-the-art periodicity counting methods cannot analyze them as their model can only input regular videos with 2D images. They could circumvent the problem by individually processing each 2D slice of the 3D images, but doing so contextual data is lost and many model inferences would be required. In the end, one count per slice would be obtained and further post-processing methods would be needed to determine the final result.

On the other hand, our method can perform 4D video analysis with no loss of context, as the model is created with the data itself. Adapting the CNN architecture is straightforward in this case: the 2D convolutions are replaced by 3D convolutions. The remaining training method is unchanged and the results obtained by our approach are as good as for conventional videos. Figure 5 gives an example of a 4D MRI video, from the results of [39], showing a beating heart. The 1D signal obtained by our method is extremely smooth and easy to interpret. Although further quantitative evaluation would need to be done, these promising results represent a proof of concept that the method is able to generalize well to other types of data.

5. CONCLUSION

We introduced a framework to count repetitions in periodic videos. This method is outside of the usual training set - validation set - testing set paradigm, as the training is unsupervised and directly done on the test data. We believe that such an unsupervised approach may be of increasing importance in the future for different applications, in order to reduce the need for big datasets and complex architectures.

Despite being unsupervised and based on a shallow model, our method gives results comparable to state-of-the-art supervised techniques with complex architectures. Due to its nature, it can work on any kind of video, even the ones that differ considerably from daily life (aeronautics, medical, astrophysics etc.). Moreover, with an appropriate neural network architecture, it can also perform well on other temporal data, such as 4D videos, biological sensors, and audio.

6. ACKNOWLEDGMENTS

This work has been funded by a PhD scholarship from CNRS. We want to thank Méghane Decroocq from the LIRIS lab who provided us with feedback on this paper, and Pierre Ripoll who helped with the visualisations.

References

[1] A. Fawzi, S.-M. Moosavi-Dezfooli, P. Frossard, The robustness of deep networks: A geometrical perspective, IEEE Signal Processing Magazine 34 (6) (2017) 50–62. doi:10.1109/MSP.2017.2740965.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.

[3] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, P. Dollár, Microsoft coco: Common objects in context (2015). arXiv:1405.0312.

[4] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M. M. A. Patwary, Y. Yang, Y. Zhou, Deep learning scaling is predictable, empirically (2017). arXiv:1712.00409.

[5] D. Hendrycks, K. Lee, M. Mazeika, Using pre-training can improve model robustness and uncertainty, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, Vol. 97 of Proceedings of Machine Learning Research, PMLR, 2019, pp. 2712–2721.
URL `https://proceedings.mlr.press/v97/hendrycks19a.html`

[6] D. Erhan, A. Courville, Y. Bengio, P. Vincent, Why does unsupervised pre-training help deep learning?, in: Y. W. Teh, M. Titterington (Eds.), Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Vol. 9 of Proceedings of Machine Learning Research, PMLR, Chia Laguna Resort, Sardinia, Italy, 2010, pp. 201–208.
URL `https://proceedings.mlr.press/v9/erhan10a.html`

[7] T. Tommasi, N. Patricia, B. Caputo, T. Tuytelaars, A deeper look at dataset bias (2015). arXiv:1505.01257.

[8] T. L. Paine, P. Khorrami, W. Han, T. S. Huang, An analysis of unsupervised pre-training in light of recent advances (2015). arXiv:1412.6597.

[9] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, A. Zisserman, Counting out time: Class agnostic video repetition counting in the wild, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[10] H. Zhang, X. Xu, G. Han, S. He, Context-aware and scale-insensitive temporal repetition counting (2020). arXiv:2005.08465.

[11] Y. Zhang, L. Shao, C. G. M. Snoek, Repetitive activity counting by sight and sound, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 14070–14079.
[12] J. Yin, Y. Wu, C. Zhu, Z. Yin, H. Liu, Y. Dang, Z. Liu, J. Liu, Energy-based periodicity mining with deep features for action repetition counting in unconstrained videos, IEEE Transactions on Circuits and Systems for Video Technology 31 (12) (2021) 4812–4825. doi:10.1109/TCSVT.2021.3055220.
[13] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, A. Zisserman, The kinetics human action video dataset (2017). arXiv:1705.06950.
[14] J. U. van Baardewijk, S. Agarwal, A. S. Cornelissen, M. J. A. Joosen, J. Kentrop, C. Varon, A.-M. Brouwer, Early detection of exposure to toxic chemicals using continuously recorded multi-sensor physiology, Sensors 21 (11) (2021). doi:10.3390/s21113616.
URL https://www.mdpi.com/1424-8220/21/11/3616
[15] E. Vavrinsky, J. Subjak, M. Donoval, A. Wagner, T. Zavodnik, H. Svobodova, Application of modern multi-sensor holter in diagnosis and treatment, Sensors 20 (9) (2020). doi:10.3390/s20092663.
URL https://www.mdpi.com/1424-8220/20/9/2663
[16] G. Kolumban-Antal, V. Lasak, R. Bogdan, B. Groza, A secure and portable multi-sensor module for distributed air pollution monitoring, Sensors 20 (2) (2020). doi:10.3390/s20020403.
URL https://www.mdpi.com/1424-8220/20/2/403
[17] G. D. Hugo, E. Weiss, W. C. Sleeman, S. Balik, P. J. Keall, J. Lu, J. F. Williamson, Data from 4d lung imaging of nsclc patients (2016). doi:10.7937/K9/TCIA.2016.ELN8YGLE.
[18] S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, A. Crimi, R. Shinohara, C. Berger, S. Ha, M. Rozycki, M. Prastawa, E. Alberts, J. Lipkova, J. Freymann, J. Kirby, M. Bilello, H. Fathallah-Shaykh, R. Wiest, J. Kirschke, B. Menze, Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge (2019) 38.
[19] C. Tobon-Gomez, A. J. Geers, J. Peters, J. Weese, K. Pinto, R. Karim, M. Ammar, A. Daoudi, J. Margeta, Z. Sandoval, B. Stender, Y. Zheng, M. A. Zuluaga, J. Betancur, N. Ayache, M. A. Chikh, J.-L. Dillenseger, B. M. Kelm, S. Mahmoudi, S. Ourselin, A. Schlaefer, T. Schaeffter, R. Razavi, K. S. Rhode, Benchmark for algorithms segmenting the left atrium from 3d ct and mri datasets, IEEE Transactions on Medical Imaging 34 (7) (2015) 1460–1473. doi:10.1109/TMI.2015.2398818.
[20] M. Tan, Q. V. Le, Efficientnet: Rethinking model scaling for convolutional neural networks (2019). arXiv:1905.11946.
[21] Z. Qiu, T. Yao, C.-W. Ngo, X. Tian, T. Mei, Learning spatio-temporal representation with local and global diffusion, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[22] G. Ciaparrone, F. Luque Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, F. Herrera, Deep learning in video multi-object tracking: A survey, Neurocomputing 381 (2020) 61 – 88. doi:10.1016/j.neucom.2019.11.023.
[23] G. Bellitto, F. P. Salanitri, S. Palazzo, F. Rundo, D. Giordano, C. Spampinato, Video saliency detection with domain adaptation using hierarchical gradient reversal layers (2020). arXiv:2010.01220.
[24] C. Panagiotakis, G. Karvounas, A. Argyros, Unsupervised detection of periodic segments in videos, in: 2018 25th IEEE International Conference on Image Processing (ICIP), 2018, pp. 923–927. doi:10.1109/ICIP.2018.8451336.
[25] O. Levy, L. Wolf, Live repetition counting, in: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), ICCV '15, IEEE Computer Society, USA, 2015, p. 3020–3028. doi:10.1109/ICCV.2015.346.
URL 10.1109/ICCV.2015.346
[26] R. Polana, R. Nelson, Detection and recognition of periodic, nonrigid motion, International Journal of Computer Vision 23 (1997) 261–282. doi:10.1023/A:1007975200487.
[27] J. Yang, H. Zhang, G. Peng, Time-domain period detection in short-duration videos, Signal, Image and Video Processing 10 (2016) 695–702.
[28] T. F. H. Runia, C. G. M. Snoek, A. W. M. Smeulders, Real-world repetition estimation by div, grad and curl, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 9009–9017.
[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need (2017). arXiv:1706.03762.
[30] B. Ferreira, P. M. Ferreira, G. Pinheiro, N. Figueiredo, F. Carvalho, P. Menezes, J. Batista, Deep learning approaches for workout repetition counting and validation, Pattern Recognition Letters 151 (2021) 259–266. doi:https://doi.org/10.1016/j.patrec.2021.09.006.
[31] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, F. A. Wichmann, Shortcut learning in deep neural networks, Nature Machine Intelligence 2 (11) (2020) 665–673. doi:10.1038/s42256-020-00257-z.
URL http://dx.doi.org/10.1038/s42256-020-00257-z
[32] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition (2015). arXiv:1409.1556.
[33] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10, Omnipress, Madison, WI, USA, 2010, p. 807–814.
[34] D. Scherer, A. Müller, S. Behnke, Evaluation of pooling operations in convolutional architectures for object recognition, in: K. Diamantaras, W. Duch, L. S. Iliadis (Eds.), Artificial Neural Networks – ICANN 2010, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 92–101.
[35] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization (2017). arXiv:1412.6980.
[36] E. Pogalin, A. W. M. Smeulders, A. H. C. Thean, Visual quasi-periodicity, in: 2008 IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[37] B. Zhou, P. Krähenbühl, V. Koltun, Does computer vision matter for action?, Science Robotics 4 (30) (2019) eaaw6661. doi:10.1126/scirobotics.aaw6661.
URL http://dx.doi.org/10.1126/scirobotics.aaw6661
[38] F. Scholkmann, J. Boss, M. Wolf, An efficient algorithm for automatic peak detection in noisy periodic and quasi-periodic signals, Algorithms 5 (2012) 588–603. doi:10.3390/a5040588.
[39] S. Schnell, P. Entezari, S. C. Mahadewia, Riti J.and Malaisrie, P. M. McCarthy, J. D. Collins, J. Carr, M. Markl, Improved semiautomated 4d flow mri analysis in the aorta in patients with congenital aortic valve anomalies versus tricuspid aortic valves, Journal of Computer Assisted Tomography 40 (January/February 2016). doi:10.1097/RCT.0000000000000312.