

PerLNet: Learning to Localize Multiple Periodic Activities in Real-World Videos

Giorgos Karvounas

Thesis submitted in partial fulfillment of the requirements for the
Masters' of Science degree in Computer Science and Engineering

University of Crete
School of Sciences and Engineering
Computer Science Department
Voutes University Campus, 700 13 Heraklion, Crete, Greece

Thesis Advisor: Professor *Antonis A. Argyros*

This work has been performed at the University of Crete, School of Sciences and Engineering, Computer Science Department.

The work has been supported by the Foundation for Research and Technology - Hellas (FORTH), Institute of Computer Science (ICS).

UNIVERSITY OF CRETE
COMPUTER SCIENCE DEPARTMENT

**PerLNet: Learning to Localize Multiple Periodic Activities in
Real-World Videos**

Thesis submitted by
Giorgos Karvounas
in partial fulfillment of the requirements for the
Masters' of Science degree in Computer Science

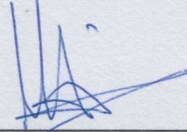
THESIS APPROVAL

Author:

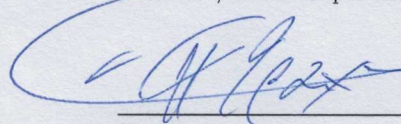


Giorgos Karvounas

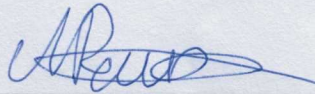
Committee approvals:



Antonis Argyros
Professor, Thesis Supervisor

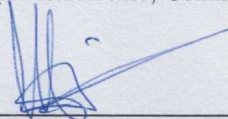


Panos Trahanias
Professor, Committee Member



Anastassios Roussos
Principal Researcher, Committee Member

Departmental approval:



Antonis Argyros
Professor, Director of Graduate Studies

Heraklion, February 2019

PerLNet: Learning to Localize Multiple Periodic Activities in Real-World Videos

Abstract

This thesis addresses the problem of temporal periodicity localization, i.e., the problem of identifying all segments of a video that contain some sort of periodic or repetitive motion. To do so, the proposed method represents a video by the matrix of pairwise frame distances. These distances are computed on frame representations obtained with a convolutional neural network. On top of this representation, we design, implement and evaluate PerLNet, a convolutional neural network that is able to classify a given frame as belonging (or not) to a periodic video segment. An important characteristic of the employed representation is that it permits the handling of videos and periodic segments of arbitrary number and duration. Furthermore, the proposed training process requires a relatively small number of annotated videos. The proposed method drops several of the limiting assumptions of existing approaches regarding the contents of the video and the types of the observed periodic actions. Experimental results on recent, publicly available datasets validate our design choices, verify the generalization potential of PerLNet and demonstrate its superior performance in comparison to the current state of the art.

PerLNet: Εντοπισμός Πολλαπλών Περιοδικών Δραστηριοτήτων σε Πραγματικά Βίντεο με Χρήση Τεχνικών Μάθησης

Περίληψη

Αυτή η εργασία ασχολείται με το πρόβλημα του χρονικού εντοπισμού περιοδικότητας, δηλαδή με το πρόβλημα της αναγνώρισης όλων των τμημάτων ενός βίντεο που περιέχουν κάποιο είδος περιοδικής ή επαναλαμβανόμενης κίνησης. Για να γίνει αυτό, η προτεινόμενη μέθοδος βασίζεται στην επεξεργασία τον πίνακα αποστάσεων όλων των ζευγών των εικόνων του βίντεο. Αυτές οι αποστάσεις υπολογίζονται μεταξύ αναπαραστάσεων εικόνων που υπολογίζονται από ένα συνελικτικό νευρωνικό δίκτυο. Εκτός από αυτήν την παράσταση, σχεδιάζουμε, υλοποιούμε και αξιολογούμε το PerLNet, ένα δεύτερο συνελικτικό νευρωνικό δίκτυο που είναι σε θέση να κατατάξει μια εικόνα ενός βίντεο ως προς το κατά πόσον αυτή ανήκει σε περιοδικό (ή όχι) τμήμα του βίντεο. Ένα σημαντικό χαρακτηριστικό της αναπαράστασης που προτείνεται είναι ότι επιτρέπει τον εντοπισμό περιοδικών τμημάτων ανεξάρτητα από το πλήθος τους και τη διάρκειά τους. Επιπλέον, η προτεινόμενη διαδικασία εκπαίδευσης απαιτεί σχετικά μικρό αριθμό βίντεο εκπαίδευσης. Η προτεινόμενη μέθοδος αίρει αρκετές από τις υποθέσεις των υφιστάμενων προσεγγίσεων σχετικά με το περιεχόμενο του βίντεο και τους τύπους των παρατηρούμενων περιοδικών δραστηριοτήτων. Τα ποσοτικά αποτελέσματα από την πειραματική αποτίμηση της προτεινόμενης μεθόδου επικυρώνουν τις επιλογές σχεδιασμού μας, επιβεβαιώνουν τη δυνατότητα γενίκευσης του PerLNet και επιδεικνύουν την υπεροχή του σε σύγκριση με τις βέλτιστες υφιστάμενες προσεγγίσεις στο πρόβλημα.

Acknowledgements

I would like to thank my supervisor Prof. Antonis Argyros for his guidance throughout all of my studies. His support, the fruitful discussions between us, were instrumental for the completion of my thesis.

I would also like to thank Dr. Iason Oikonomidis for his advice and support for over a decade now.

Finally I would like to thank Dr. Eleftheria Tzamali, Dr. Nikolaos Kyriazis, Dr. Alexandros Makris, Paschalis Paderis, Dennis Bautembach, Kostas Papoutsakis, Dr. Aggeliki Tsoli, Dr. Kostas Panagiotakis, Konstantinos Varsos, Filippou Gouidis, Fotis Koutoulakis, Spuros Chiotakis, Kostas Koukoulakis, Sotiris Avgelis, and Stelios Koutsoudakis for their support.

στους γονείς μου

Contents

Table of Contents	i
List of Tables	iii
List of Figures	v
1 Introduction	1
2 Related Work	7
2.1 Deep Learning and Deep Architectures	7
2.2 Periodicity Localization and Characterization	9
2.3 Exploiting Periodicity	10
2.4 Repetition Counting	11
2.5 Data Mining	11
2.6 Motif Detection	11
2.7 Our Contribution	12
3 Exploiting Periodicity	13
3.1 Data Representation	13
3.1.1 Features based on IDT	14
3.1.2 Features based on Deep Learning	15
3.2 Data Preparation	15
3.3 Ground Truth Annotation	15
3.4 Learnable Periodicity Representations	16
3.5 Data Augmentation	17
3.6 PerLNet: Periodicity Localization Network	18
3.7 Loss Function	19
3.8 Weighted Intermediate Supervision	19
3.9 Testing	19
4 Experimental Evaluation	21
4.1 Training Details	21
4.2 Datasets	21
4.2.1 The PERTUBE Dataset	21

4.2.2	The QUVA Dataset	22
4.3	Monte-Carlo Cross Validation	22
4.4	Evaluation Metrics	22
4.5	Ablative Study	22
4.6	Comparison to SoA & the Impact of Features	23
4.7	Cross-dataset Validation	24
4.7.1	Training on QUVA, Testing on PERTUBE	24
4.7.2	Training on PERTUBE, Testing on QUVA	25
4.8	Qualitative Results	25
5	Discussion	31
5.1	Summary	31
5.2	Future Work	31
	Bibliography	33

List of Tables

4.1	Meta-parameter study of the proposed neural network.	23
4.2	Comparative evaluation of PerLNet with [48] on the PERTUBE dataset with CNN-based (VGG19) and hand-crafted (IDT) features. Training of the proposed method has been performed on the PERTUBE dataset.	23
4.3	Cross-dataset evaluation: The proposed method, PerLNet, was trained on QUVA and evaluated on PERTUBE.	25
4.4	Cross-dataset evaluation: The proposed method was trained on PERTUBE and evaluated on QUVA. The method in [48] requires no training.	25

List of Figures

1.1	A sinusoidal signal (left) and its self-similarity matrix (right). . .	2
1.2	The signal produced by the equation 1.3 is on the left, with the self-similarity matrix on the right.	2
1.3	In the upper left plot we visualize a periodic signal between random noise, with red bars that signify the start and stop of the periodic part. In the upper right we visualize the self similarity of the signal. With the red rectangle we highlight the periodic part. The bottom figure plots the Fast Fourier Transform (FFT) of the signal in the upper right plot. The red circle corresponds to the ground truth regarding the signal.	5
1.4	We address the problem of classifying parts of an input video as periodic or non-periodic. Still frames of an example video are shown, with the time running from left to right and from top to bottom. In the video, a man starts by talking to the camera (non-periodic action), followed by doing crunches (periodic action). Then he stands up (non-periodic), performs some kicks (periodic) and finally again talks to the camera (non-periodic). The output of the proposed method is color-coded as red for non-periodic parts, and green for periodic ones.	6
3.1	Top: The distance matrix M of a video that contains two periodic activities. Warm (cold) colors indicate larger (smaller) distance between frame descriptors. Bottom: 1D ground truth on periodicity localization (red: non-periodic, green: periodic).	14
3.2	A distance matrix M (left) and the corresponding binary ground truth matrix A (right). Two different sub-blocks are also shown in purple color (see text for details).	16
3.3	The building blocks of the architecture of PerLNet, the proposed CNN for periodicity localization. The colors denote layer types. Orange: convolution and non-linearity, red: max-pooling, blue: up-sampling. The count of feature maps is shown under each convolution layer, and the size of the resulting feature maps is shown diagonally to the right of the layer (also visually as the size of the block). Finally, the blue sphere denotes addition.	17

4.1	Video frames from the (a) PERTUBE and (b) QUVA datasets.	26
4.2	In the upper left plot we visualize the distance matrix computed with IDTs. In the upper right we visualize the distance matrix computed using deep features. The bottom plot visualizes the ground truth. With white we denote the periodic region and with black the non-periodic.	27
4.3	The distance matrices, the periodicity localization estimation (1st green/red bar) and the ground truth (2nd green/red bar) for six sequences of the PERTUBE dataset.	28
4.4	The distance matrices, the periodicity localization estimation (1st green/red bar) and the ground truth (2nd green/red bar) of the QUVA dataset.	29

Chapter 1

Introduction

Periodic patterns are ubiquitous in both natural and man-made environments. Common human motions and activities such as walking, running, hand waving, breathing, etc, give rise to periodic patterns. Detecting such patterns is both effortless and useful for humans [30] who use visual, aural and tactile signals as the primary sources of sensory information to solve this task.

The period of a signal is the smallest value T_o for which this signal repeats itself. The mathematical definition of a periodic signal is described by the equation:

$$X(t) = X(t + T_o), \forall t, \quad (1.1)$$

where X is a periodic function over the free variable t . A periodic signal, that is described by the equation:

$$x(t) = \sin(2\pi ft) \quad (1.2)$$

is illustrated in Figure 1.1. For any signal, we also define its self- similarity matrix, that is the square matrix representing the similarity in any pair of samples of the signal. For such a signal, methods such as the Fast Fourier transform can be used to estimate their frequency and, thus, their period. This is because no noise contaminates the amplitude or the phase of the signal. However, in the real world we deal with more complex types of signals where the equation 1.1 does not hold in the strict sense. As an example, Figure 1.2 illustrates the signal

$$x(t) = e^{-t} \sin(2\pi ft). \quad (1.3)$$

In this case, it is not clear where the periodic part fades-out. In this case, methods based on peak detection [60] can still detect the periodic part, but in more complex cases will fail. Recent research has led to the development of algorithms that can tolerate such cases. The answer of localizing the periodicity in such signals usually depends on the application.

A situation that is quite common in practice is when a periodic signal appears between a non periodic prefix and suffix, as illustrated in Figure 1.3. In such a case, peak detection algorithms will fail to detect the periodic segment. Fourier

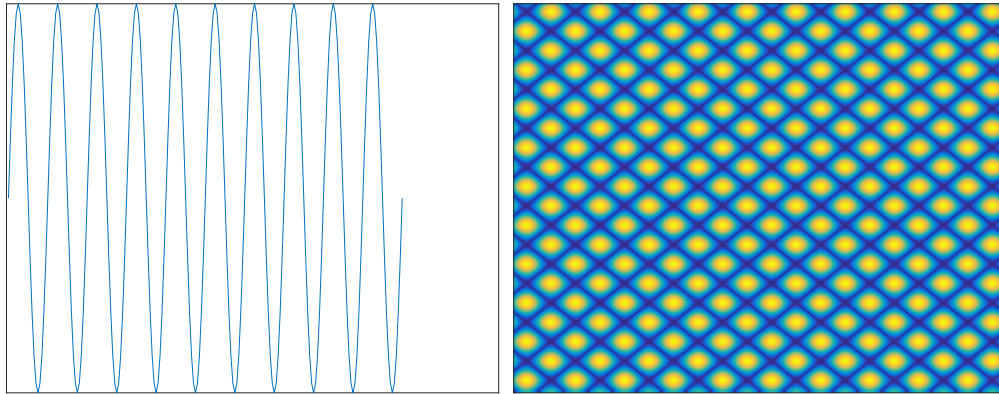


Figure 1.1: A sinusoidal signal (left) and its self-similarity matrix (right).

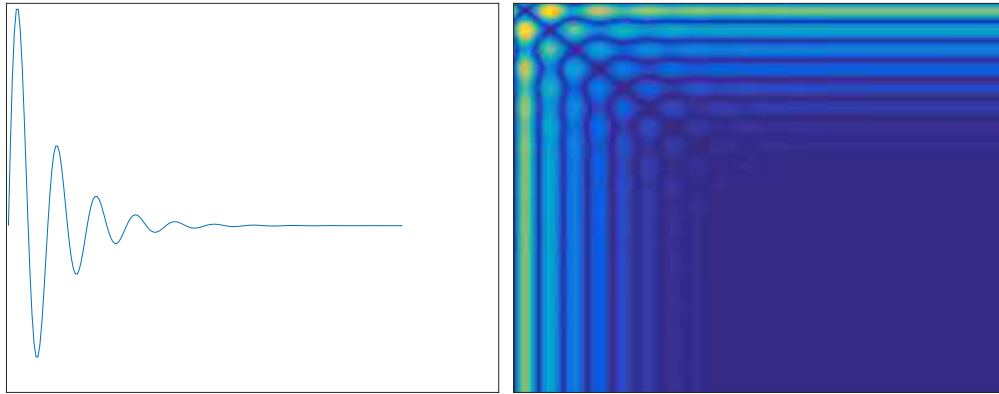


Figure 1.2: The signal produced by the equation 1.3 is on the left, with the self-similarity matrix on the right.

transform also fails to detect the frequency of the periodic part of the signal. This happens because of the noise that surrounds the signal. Such cases are very common in both natural and man-made environments and diverse settings. As an example, imagine the situation of a periodic fitness exercise that occurs between other non-periodic human activities.

We are interested in developing methods for the automatic segmentation of periodic parts of signals. Specifically, this work deals with the problem of *periodicity localization*, that is, the identification of all periodic segments in a video. Given an input video, our goal is to identify all the frames of the video where a repetitive, periodic motion is observed. Consider, as an example, the situation shown in Figure 1.4. In this example, a man starts by talking to the camera (non-periodic motion), followed by doing crunches (periodic motion), then stands up (non-periodic), performs some kicks (periodic), and finally again talks to the camera. The desired output is color-coded as red for non-periodic video segments,

and green for periodic ones. Given this identification of periodic segments, a potential next goal is *periodicity characterization*, i.e., the estimation of the length of the period of the periodic motion in each segment. Periodicity characterization is out of the scope of this work. However, a solution to the periodicity localization problem simplifies the problem of periodicity characterization.

From a practical point of view, many real-world applications like classification [44, 22, 3, 55], 3D reconstruction [5] and camera calibration [27] as well as medical assessment [52] and rehabilitation are applications that can benefit and make use of the periodicity as a useful building block. Work has also been carried out in industrial inspection with minimal prior knowledge [31]. Additionally, algorithms for solving the problem of human action and motion detection can benefit by knowing the location of the periodicity and the period length. In both cases, periodicity, can also give significant cues of the action that is performed. Studies have shown that every human has different gait cycle [11, 68]. Therefore in classification, periodicity detection is useful for human (re-)identification based on their gait cycle. In a broader view, periodicity detection is also useful in the area of chaotic systems [51, 58, 12]. In dynamic systems, detecting unstable periodic orbits can aid to modelling many complex phenomena concerning geophysics, space physics and fluid dynamics.

Periodicity detection and characterization using visual input form a category of problems that has been widely studied [43, 66], and is still of significant interest to the community of computer vision [48, 57]. Many different approaches have been proposed to tackle periodicity localization and related problems. The mathematical tools that have been mainly used is spectral analysis and the Fourier transform. Such approaches exhibit varying levels of computational complexity, leaning towards low computational cost. On the other hand, their main drawback is their strong assumption about an almost perfectly stationary input signal. Formally, stationary signals exhibit constant statistical properties over time as in Figure 1.1. This is the class of signals that are usually termed periodic. Real-world signals deviate from being stationary and truly better resembling the one in Figure 1.2. Repeated actions may differ in their duration (thus, the period of the action is not constant) and/or in their actual execution. These actions produce non-stationary signals. Non-stationarity can be generally considered as another source of noise or contamination to a periodic signal, but it is usually expressed in specific manners. Such a common manner is the repetition of roughly the same pattern at slightly different time intervals. Therefore, successful periodicity localization methods that deal with these types of non-stationarity have to address significant challenges.

Other approaches have been based on heuristics to match repetitive patterns. Such approaches usually have a larger tolerance to input noise, or other kinds of input corruption, according to the target of the adopted heuristic. The relevant methods also need to address a number of challenges. To begin with, the amount of data contained in a regular video stream is big, even for the standards of modern hardware. Further complicating factors include camera motion, sensor noise and the non-stationarity of the observed motion. In a video where the camera is

moving, the observed appearance of the scene can change drastically, making the detection of the periodic pattern extremely challenging. Main research topics in the area include the development of methods that are tolerant to sensor noise and non-stationary input signals.

In this thesis we propose a learning-based method for periodicity localization in videos. Our contributions are many-fold. First, we propose a representation of the input video that makes it possible to train convolutional neural networks to solve the problem. Based on this representation, it is possible to develop CNNs that generalize well based on a relatively small training dataset. Second, within the proposed video representation, we design and propose a deep convolutional neural network to detect and localize multiple periodic patterns. Third, we give a solution to training with unbalanced samples using single model. Finally, the experimental evaluation on existing benchmark datasets showing that the proposed approach achieves state-of-the-art performance.

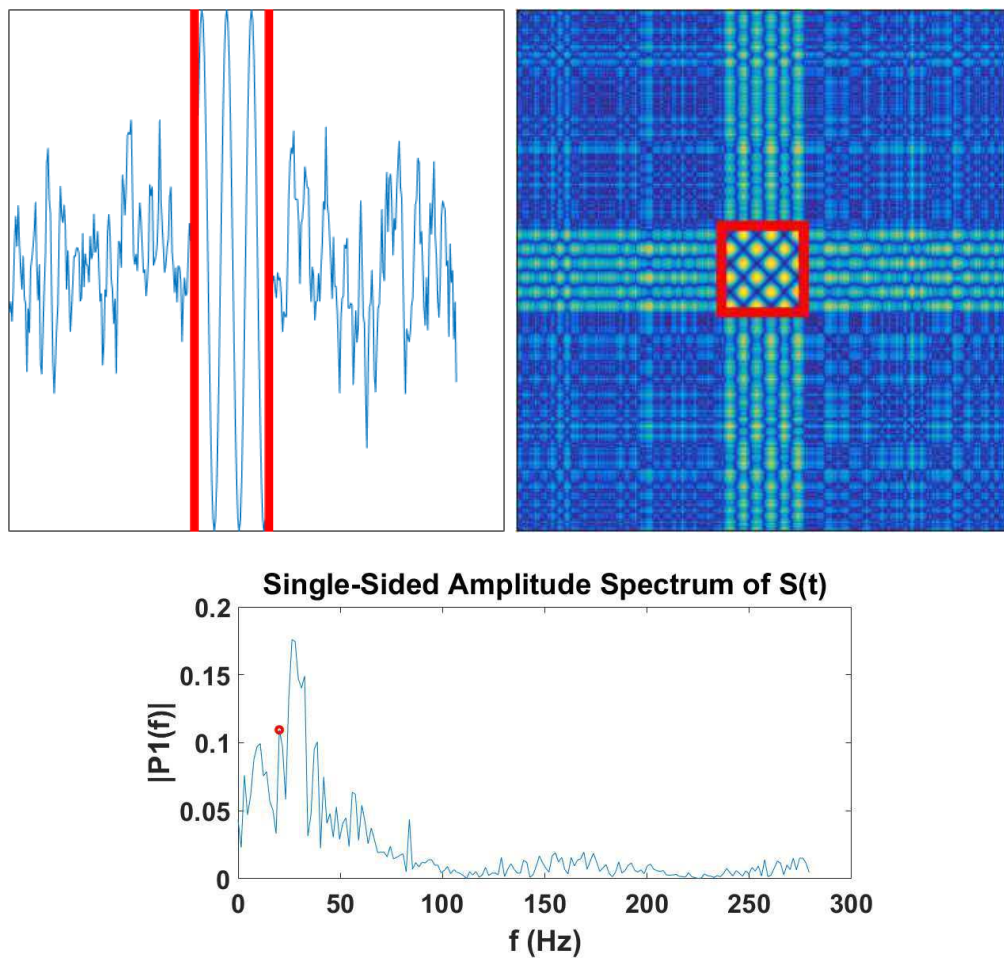


Figure 1.3: In the upper left plot we visualize a periodic signal between random noise, with red bars that signify the start and stop of the periodic part. In the upper right we visualize the self similarity of the signal. With the red rectangle we highlight the periodic part. The bottom figure plots the Fast Fourier Transform (FFT) of the signal in the upper right plot. The red circle corresponds to the ground truth regarding the signal.



Figure 1.4: We address the problem of classifying parts of an input video as periodic or non-periodic. Still frames of an example video are shown, with the time running from left to right and from top to bottom. In the video, a man starts by talking to the camera (non-periodic action), followed by doing crunches (periodic action). Then he stands up (non-periodic), performs some kicks (periodic) and finally again talks to the camera (non-periodic). The output of the proposed method is color-coded as red for non-periodic parts, and green for periodic ones.

Chapter 2

Related Work

The tasks of detection and analysis of periodic patterns have received the attention of researchers from various computer science fields [2, 54, 16, 53]. Problems in this category include the detection of repeating patterns within a video [54, 53] or a database [2, 16, 17] as well as the exploitation of the periodicity as a basis for solving a number of other problems [5, 27]. Recent advances in machine learning and in particular deep learning have revolutionized many disciplines, including machine vision. We argue in this work that periodicity localization can benefit from these advances.

2.1 Deep Learning and Deep Architectures

Deep learning is part of machine learning and includes algorithms that are inspired by the structure and function of the brain, called deep neural networks. Deep neural networks(DNNs) are the evolution of the traditional artificial neural networks(ANNs) [23]. The differences between ANNs and DNNs is that the DNNs can have more than three hidden layers and that the DNNs can extract their own features specifically for the training set that is used. In computer vision, the convolutional neural networks (CNNs), are widely used for many tasks like image classification [15]. CNNs also belong the family of DNNs and were introduced by LeCun et al [36]. This first CNN was designed to recognise [36] hand-written ZIP code numbers using back-propagation to learn the convolution kernel coefficients (or weights) directly from images of hand-written numbers.

One disadvantage of CNNs is that they require large computational capability and memory capacity. For this reason algorithms like LeNet-5 proposed by LeCun et al [37] were difficult to implement until 2010. LeNet-5 was introduced to automatically classify hand-written digits on bank cheques in the United States. This network has only 7 layers, among them 2 convolution layers, 2 sub-sampling layers, 2 fully connected layers, and an output layer with Gaussian connection. Krizhevsky et al. [32] proposed a CNN for image classification and won the ImageNet LSVRC-2010 challenge [14]. This network was wider and deeper than

LeNet-5. With AlexNet [32], the Local Response Normalization (LRN) was also introduced. The idea behind LRN is to simulate lateral inhibition and boost the neurons with relatively larger activations. Simonyan et al. [62] show in their work the importance of depth in CNNs. With their three proposed architectures, VGG-11, VGG-16, and VGG-19, they won the ImageNet LSVRC-2014 challenge [14]. The proposed architectures had 11, 16 and 19 layers respectively. The design of the VGG architecture is built around the repeated use of two convolutional layers that both use the ReLu (Rectified Linear Unit) [21] activation function, that are followed by a max pooling layer. At the end, they use several fully connected layers also using as activation function the ReLu [21].

Another type of neural network is the residual neural network architecture. Residual neural networks are using skip connections or short-cuts to skip over one or more layers. He et al. [24] they introduced the residual CNNs (ResNet). ResNet was the winner of the ImageNet LSVRC-2015 challenge [14]. In their work they proposed 5 different networks with 34, 50, 101, 152 and 1202 layers. At the output of every convolution layer they employ Batch Normalization [28]. The intuition behind using skip connections is to address with the problem of vanishing gradient by feature reusing.

Autoencoders are another type of ANN architecture and in contrast to the previous mentioned models, can be used to learn from unlabeled examples. This strategy for learning is called unsupervised learning. Autoencoders consist of two stages, the encoder and the decoder. The encoder stage takes the input data and maps them to feature representations. The decoder does the opposite work, taking the feature representations and translating them to the initial data. The first autoencoder was introduced by Rumelhart et al. [56] and trained using back-propagation. Using the autoencoder they also gave the first solution based on ANNs for the XOR problem. The first deep autoencoder was introduced by Hinton et al. [26] and also featured a weight initialization for tackling high dimensional data. They also prove that their deep autoencoder worked better than Principal Component Analysis. Image denoising is another problem that can be addressed by deep autoencoders. Vincent et al. [65] introduced a stacked architecture with autoencoders to tackle the problem. Recently many computer vision problems are addressed by convolutional autoencoders. Convolutional autoencoders use the same pattern with ANN-based autoencoders with encoding and decoding using convolution and deconvolution or down-sampling and up-sampling. Masci et al. [45] introduced the first convolutional autoencoder for feature extraction. Convolutional autoencoders are performing equally well to other approaches in problems like clustering [20], 3D object recognition [38] and 3D human face reconstruction [63].

2.2 Periodicity Localization and Characterization

The problems of periodicity localization and characterization can be defined in the temporal and spatial domains [13] but also in several others that involve either 1D signals [44, 16, 8] or multidimensional signal representations. Naturally, Fourier transform and, more generally, spectral analysis tools have been employed by many methods [54, 13, 52]. This approach yields satisfactory results in clean, 1D signals, but fails in the presence of large amounts of noise or other sources of signal corruption that invalidate the assumption of signal stationarity. Special care must then be taken to preprocess the input appropriately so that the resulting signal(s) meet the requirements imposed by the selected spectrum analysis tool.

The work by Lu et al. [44] tackles the problem of segmenting and classifying repetitive motion events using visual input. The method decomposes the input video into motion primitives using a multidimensional signal segmentation algorithm and then uses these primitives to segment and classify repetitive motion. Burghouts and Geusebroek [8] propose a method to detect and characterize periodic motion. A spatiotemporal filtering approach is adopted, yielding localization of the observed periodic events. The method of Tong et al. [64] operates on local motion (optical flow) information [6] computed in videos. Periodicity detection and characterization is performed by estimating statistics of the autocorrelation computed over this motion representation. Albu et al. [3] use silhouettes as the main visual cue. They focus on human silhouettes to obtain 1D signals representing motion trajectories of body parts. These signals are then processed to detect their periodic parts, with a final step of the method fusing these detections into a single coherent estimation of the periodic parts of the input. The authors show experimental results of their method on repetitive activities such as aerobic exercises and walking.

Briassouli and Ahuja [7] propose a method to extract multiple periodic motions in a video sequence. They propose the use of Short Term Fourier Transform (STFT) on the volume of the input video. This approach allows for the simultaneous detection and characterization of periodic activities. Pogalin et al. [53] define 10 different classes of periodic motions, and build a system to detect and classify them. To do so, they begin by tracking all the objects in the input stream. This is followed by probabilistic PCA of the resulting tracks, and spectral analysis for detecting and characterizing periodic motion. Azy and Ahuja [4] propose a method to detect and segment objects that move in a periodic manner using a maximum likelihood estimation of the period to characterize the motion. Image segmentation is performed using correlation of image segments over the estimated period. Taking the inverse approach, Gaojian Li et al. [40] first localize the target object in a region of interest and then characterize its periodic motion. Motions with mild non-stationary components are shown to be handled effectively. Karvounas et al. [31] detect and characterize the periodic part of an input video by formulating the task as an optimization problem. This work assumes that only one periodic activity exists per video. In the case that this assumption does not hold,

only one of the periodic activities will be detected, or the method may completely fail to detect and characterize a periodic action. Panagiotakis et al. [48] treat the detection of multiple periodic actions as a problem of video co-segmentation. More specifically, they capitalize on a method [49] that co-segments all common actions of two videos, in an unsupervised manner. By co-segmenting actions in a single video, the proposed work manages to identify periodic motions.

2.3 Exploiting Periodicity

The cue of repetitive motion is very strong and can be used to detect events in a video [44, 34]. In a work demonstrating the power of the periodicity cue using visual input [59], Sarel and Irani show that it is possible to separate two superimposed layers in a video, under the assumption that one of the layers exhibits repetitive motion dynamics. As an example, a man performing jumping jacks is filmed through a glass door that reflects moving people on the other side of the door. The proposed method separates correctly the two scenes by assuming that periodic motion occurs in one of the two layers. Another work where periodicity serves as a strong cue to detect and segment motions is presented by Laptev et al. [34]. In that work, the authors start by observing that corresponding time instances in successive periods of a periodic activity serve as approximate stereo pairs. This observation is then exploited to segment and characterize periodic activities using space-time correspondences across different periods of the observed motion. Goldenberg et al. [22] propose the use of the sequence of silhouettes of a periodically moving object such as a walking dog as a basis for the representation and analysis of the motion towards action classification. The authors propose the eigen-decomposition of this set of silhouettes as a means to extract an intermediate representation. The authors proceed to show how this representation can efficiently solve the task of object and action classification. Xiu Li et al. [42] exploit periodicity to develop a method for non-rigid structure from motion. The input to the method is a monocular video of a non-rigid object undergoing a possibly repetitive dynamic motion. The authors observe that many deforming shapes tend to repeat themselves in time, a property termed “shape recurrency”. Based on this property, the authors apply standard, rigid structure-from-motion tools on this problem. Xiaoxiao Li et al. [41] develop the Repetitive Motion Estimation Network (RMEN) for recovering cardiac and respiratory signals. The input of the network is a video of cardiac activity and the output is a sinusoidal signal. Based on the output signal they perform a peak detection algorithm to finally recover cardiac/respiratory signals. Nevertheless the method is performing well but is only tested in very constrained videos.

2.4 Repetition Counting

Levy and Wolf [39] propose a method that tackles the problem of repetition counting. They use a CNN on windows of the input stream to detect and characterize periodic activities. Based on this output, they proceed to estimate the number of repetitions in the input video stream. The recent method by Runia et al. [57] for repetition counting analyzes the different possible viewpoints of a periodic motion with respect to the dominant orientation of motion in 3D. Then, the method selects the viewpoint with the best signal, estimating the length and period of motion. This, in turn, yields an estimation of the repetition count, the end goal of this work. This method only tackles the problem of counting the repetitions of a single action, so it cannot handle two or more different periodic activities in a video.

2.5 Data Mining

In data mining, detecting periodic patterns can aid in detecting trends in time series. This is currently a very important task since temporal data are increasing constantly [18]. Elfeky et al. [16] proposed the WARP algorithm for the detection of reoccurring, same or similar transactions in databases. Lahiri and Wolf [33] propose a single-pass, polynomial time, algorithm for finding periodic graphs and subgraphs in a social network. It is well known that in social networks it is computationally costly to mine periodic interaction patterns. Periodic patterns could also be mined from social media users. Yuan et al. [69] propose a Bayesian model, called Periodic REgion Detection (PRED). This method discovers periodic mobility patterns by jointly modeling geographical and temporal information. A useful property of PRED is that it is non-parametric.

2.6 Motif Detection

The detection of motifs in time series, is another problem relevant to periodicity detection. Works on motif detection are focused on time series resulting from financial data, biosequences and other areas. Chiu et al. [9] proposed a probabilistic method to detect motifs in the presence of noise. Gao and Lin [19] introduce an approximate algorithm called Hierarchical based Motif Enumeration (HIME) to detect variable-length motifs with a large enumeration range in million-scale time series. Mueen and Keogh [46] introduced an algorithm that monitors and maintains motifs exactly in real time over the most recent history of a stream. The algorithm has an update time linear to the window size in which the algorithm searches for motifs. Serra and Arcos [61] propose the SWARMMOTIF, an algorithm based on Particle Swarm Optimization for finding motifs. The SWARMMOTIF algorithm can find only a predefined number of motifs with a prefix

dissimilarity.

2.7 Our Contribution

The overview of the relevant literature indicates that, with the exception of [48], no method can tackle the problem of detecting and localizing the periodic segments of a real world input video in its full generality. Existing methods make restrictive assumptions regarding the stationarity of the observed signals. For example, they consider strictly periodic signals with almost constant period and very similar execution of actions in the different periods. Others, assume that a single periodic action is executed in a video and, therefore, cannot localize several of them if they exist. Finally, others are tied to specific (and thus, limiting) settings and video content representations. For example, certain methods assume a static camera. Others operate on representations of the human body, thus they may deal only with periodic patterns of human motion. In our work, we present a deep learning method that does not suffer from the above limitations.

We first employ a generic video-content representation on which we show that periodicity localization can be learned. We also propose PerLNet, a specific deep learning architecture to solve this problem. We show that, based on the employed representation, PerLNet learns to localize periodicity on the basis of a relatively small training dataset. Finally, we evaluate PerLNet quantitatively on existing public datasets and in comparison with existing approaches. The experimental results show that the proposed approach outperforms the current state of the art.

Chapter 3

Exploiting Periodicity

The proposed approach operates in two main steps. In the first step the input video is processed to compute an intermediate representation that captures the temporal context of each video frame. This representation is then used in the next step to characterize a frame of the video as periodic or not. We first represent each frame of a video as a vector of deep features that encode high-level information for that frame. Based on this frame representation, we compute a matrix M of the pairwise distances of all video frames. The existence of periodic actions in a video gives rise to distance matrices with a particular block diagonal structure. Based on this observation, we proceed to train a neural network that classifies square blocks on the main diagonal of M into two classes (periodic / non-periodic). Figure 3.1 gives an example of a distance matrix of a video showing two periodic actions among other, non-periodic parts.

3.1 Data Representation

We approach the problem of localizing periodic segments building upon the idea of a distance matrix which has been adopted also by other works in video analysis [13, 50, 48]. Specifically, assuming that the input is a video of N frames, we construct an $N \times N$ matrix M . Each entry m_{ij} of M quantifies the distance between frames i and j . Assuming that some temporally periodic pattern is present in the observed video, the matrix exhibits particular structures. For example, it may contain sub-diagonals that are parallel to the main diagonal with low distance values, capturing the periodic nature of the observations.

The similarity between two frames can be quantified in many ways. Firstly, a convention must be established, regarding the use of metrics that quantify similarity, or alternatively distance. In the first case, higher values denote higher similarity, while in the second case, higher (distance) values denote lower similarity. In our experiments we followed the convention of distance values, since this is exactly what we computed: the Euclidean distance between feature vectors.

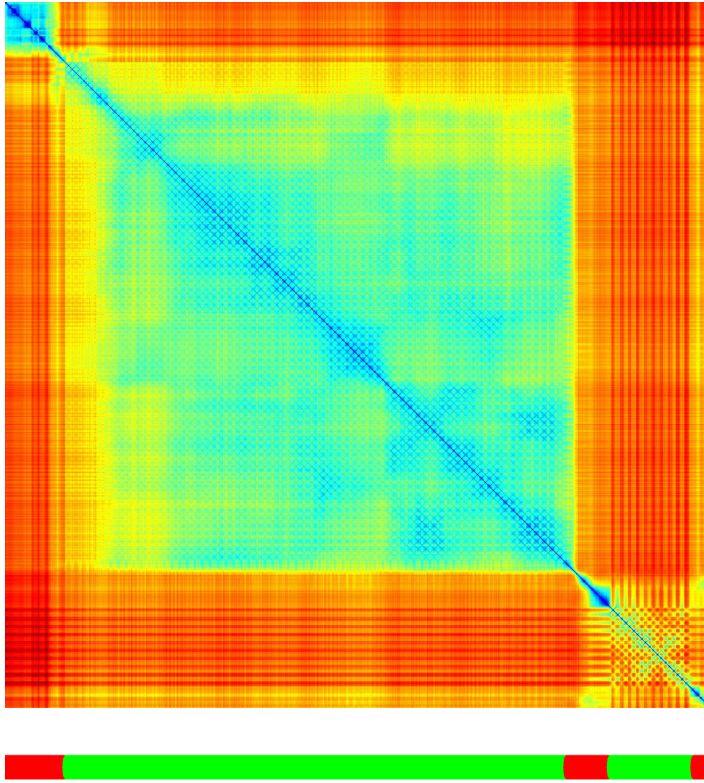


Figure 3.1: Top: The distance matrix M of a video that contains two periodic activities. Warm (cold) colors indicate larger (smaller) distance between frame descriptors. Bottom: 1D ground truth on periodicity localization (red: non-periodic, green: periodic).

This design choice has almost no practical impact on the design of the next steps, apart from the need to appropriately handle the values according to the chosen convention.

An intermediate representation for each frame of the input video is useful for the task of computing the entries of matrix M . The representation of a frame can, in principle, be anything that captures visual content, disentangling it from signal-specific and periodicity-irrelevant details. The description of two such alternative representations follows below. The first is the representation as employed in the work by Panagiotakis et al. [48], and the second is the adopted approach that employs features based on deep learning.

3.1.1 Features based on IDT

Panagiotakis et al. [48] opted for the use of tracklets, short sequences of optical flow trajectories. This method starts with the extraction of IDT(Improved Dense

Trajectories) for a sequence X of N frame. Then for each of the descriptors of Improved Dense Trajectories (IDT, HOF, HOG, MBHx, MBHy) we build a visual vocabulary of K (fixed) number of visual words. We select a temporal window of length $2w$, where w is the number of frames we look before and after the selected frame, and for each frame t_i , $i=[0,N)$ of the sequence X , we center the window on t_i , we calculate a histogram of size k , encoding the distances of the trajectories in the frames of the temporal window to the clusters' centers (visual words). The resulting normalized (L1 or L2) histogram H_i of length k (k bins) serves as a partial feature vector F_i of frame t_i . Finally we concatenate all partial feature vector histograms $H_i(\text{idt})$, $H_i(\text{hof})$, $H_i(\text{hog})$, $H_i(\text{mbh})$ of each frame t_i to form a large feature vector F_i for frame t_i . Having calculated the feature vectors T_i for all N frames, a self-similarity matrix of dimensions $N \times N$ is calculated among all feature vectors using Euclidean distance. This approach can successfully separate appearance from motion, but is prone to tracking failures because the objects might move arbitrarily fast so that IDT fails.

3.1.2 Features based on Deep Learning

In this work we propose the use of features that are computed by a deep convolutional neural network. Specifically, the activation features of the VGG19 network [62] pre-trained on ImageNet [14] at the 15th layer are computed for each frame of the input video. This particular layer has been selected as it strikes a good balance between the required level of feature abstraction and the need to keep features in spatial relation with the input. Then, for each pair i and j of frames of the video, the Euclidean distance of their precomputed feature vectors is computed as the value m_{ij} of M . An example of such a matrix is visualized in Figure 3.1.

3.2 Data Preparation

Since we use self-similarity matrices to encode the videos, two similar frames will have a small pairwise distance. On the other hand, frames that are very different, will have unboundedly high pairwise distances. Training with unnormalized data we observed that the network is very sensitive to the magnitude of the the input, often completely failing to converge. For this reason we normalize our data to the range $[0, 1]$. For the normalization we divide each element of a matrix by its highest value.

3.3 Ground Truth Annotation

Having computed the distance matrix M based on the input video, our next task is to denote each frame of the video as part of a periodic pattern or not. We assume

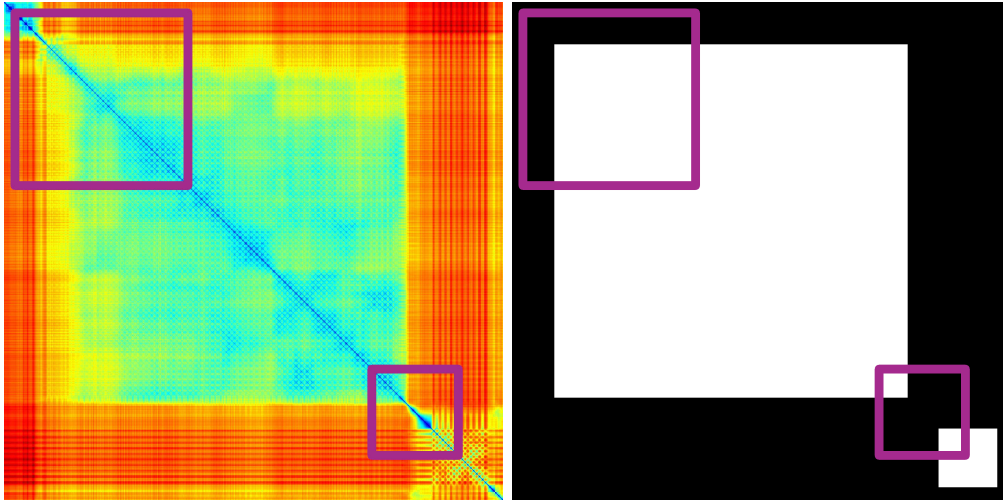


Figure 3.2: A distance matrix M (left) and the corresponding binary ground truth matrix A (right). Two different sub-blocks are also shown in purple color (see text for details).

that there is ground truth annotation for a number of input videos. Specifically, the annotation for a video is a list of pairs of frames of the form (s_i, e_i) , where s_i is the frame count of the start of the i -th periodic segment, and e_i is the ending frame of that segment.

It is not practical to train a neural network directly on the ground truth represented in the form of periodic segments (s_i, e_i) . There are several reasons why this is the case. Firstly, the input size of the image sequence (or periodic part of it) can be arbitrarily large. This would make the resulting network prohibitively large for practical use on current hardware. A solution to the arbitrary input size is the use of recurrent neural networks. We do not adopt this design choice because of the additional complexity to fine-tune a recurrent neural network. Another reason is that this strategy has no clear way of representing multiple periodic segments that may exist in a video. Experience has also shown that, whenever possible, tasks to be solved using machine learning should be formulated as classification problems instead of regression ones [35]. In order to address these issues, we adopt a representation that is efficient with respect to input size, and represents the target output in a way that is compatible with a fully convolutional architecture.

3.4 Learnable Periodicity Representations

To tackle the issues above, we resort to a binary, square matrix A with the same dimensions as M . We adopt the convention that a value a_{ij} of A is equal to 1 if both frames i and j belong to the same periodic segment. Otherwise, $a_{ij} = 0$. Figure 3.2 shows a distance matrix M (left) and the corresponding binary matrix

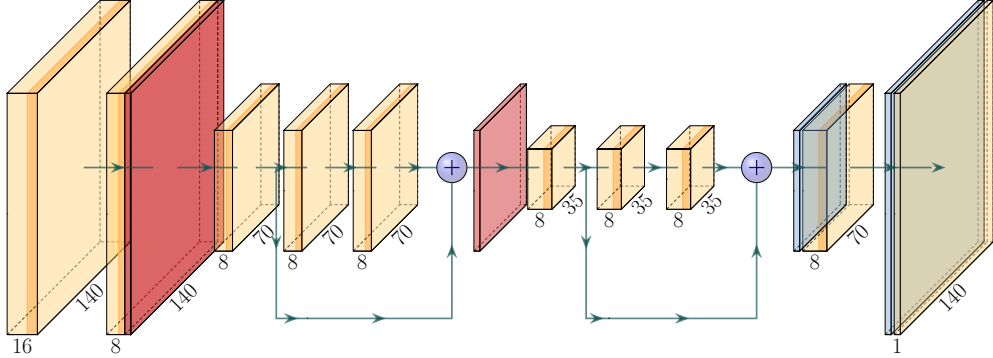


Figure 3.3: The building blocks of the architecture of PerLNet, the proposed CNN for periodicity localization. The colors denote layer types. Orange: convolution and non-linearity, red: max-pooling, blue: upsampling. The count of feature maps is shown under each convolution layer, and the size of the resulting feature maps is shown diagonally to the right of the layer (also visually as the size of the block). Finally, the blue sphere denotes addition.

A (right). Given the annotation information as described in Section 3.3, it is straightforward to compute the respective matrix A .

Representing the input video content using the matrix M , and the ground truth annotation as the matrix A , we propose to train a network on fixed-size sub-blocks of these matrices. Since the information regarding the periodicity is encoded around the main diagonal of the matrix M , any square block of the matrix M that is centered on the main diagonal of M can be used as a training sample. The corresponding block of the matrix A can then serve as the ground truth annotation for this training sample.

Given this information, we proceed to train a deep neural network on such training samples. Overall, the training samples are corresponding pairs of sub-blocks of the matrix M and A . It is fine to select sub-blocks of M that have large overlap since this will essentially result in data augmentation that trains the network to be translation invariant. This process is repeated for all matrices M_i coming from input videos V_i in the training set, resulting in thousands of training sub-blocks given a few input videos.

Figure 3.2 shows two example training sub-blocks (purple squares) superimposed on a distance matrix M (left) and on the respective ground truth annotation matrix A (right).

3.5 Data Augmentation

In the case of a network that accepts visual input such as a regular color image, it is common practice to apply data augmentation in the form of geometric and

intensity transformations [32] so as to force the neural network to become invariant to such transformations. Our case is different, since the goal is to learn the patterns generated from repetitive, periodic motion. The only useful form of data augmentation applicable to our input data is one that achieves temporal scale invariance. This is effectively achieved by varying the size of the sub-blocks used during training. Extreme cases where the scale is twice the original size network lead to misclassification of negative examples. Therefore we limit the range of augmentation to 50% of the length of small videos (less than 250 frames). The highlighted sub-blocks in Figure 3.2 are of different size, following this data augmentation approach.

3.6 PerLNet: Periodicity Localization Network

PerLNet Architecture: Given the defined input and target output, we train a convolutional neural network to learn this mapping. The employed architecture is a stack of hourglass modules [47], that are essentially autoencoders [25] with skip connections [24]. The architecture consists of three stacked autoencoders. The reason we adopted an autoencoder-like network is because we are targeting an output with the same size as the input. Furthermore, the autoencoder architecture offers good generalization by squeezing the information through the (spatially) low-resolution intermediate layers.

The proposed neural network is built using copies of two basic building blocks, an encoder part, and a decoder part. This architecture is shown in Figure 3.3. The encoder first applies 16 convolution filters of dimension 11×11 . This is a rather large spatial dimension, however it is justified because the patterns we are looking for are rather large-scale and noisy. The encoder then contains a ReLU activation layer, followed by a batch normalization layer [28]. Another set of convolution, activation, and batch-norm layers follow in the original input dimension, and then a max pooling layer halves the resolution in each of the two spatial dimensions of the input. Two more sets of convolution, activation, and batch normalization follow, and in parallel to these layers, an identity residual connection [24] is also used. The skip connection is added to the result of the other branch, and a last max-pooling layer halves again the spatial resolution. In total, the encoder applies 5 different convolution layers and two max pooling operations, resulting in a output spatial resolution that is $1/4$ of the input resolution in each spatial dimension. The decoder follows essentially the mirrored connectivity pattern of the encoder, upsampling its input by a factor of 4 for each input spatial dimension. There are again 5 layers applying a convolution operation, and a skip connection runs parallel to a group of two such convolutions.

3.7 Loss Function

We build a classifier with two output classes so it is natural to use binary cross entropy (BCE) [29] as the loss function for training:

$$E(y, p) = -(y \log p + (1 - y) \log(1 - p)). \quad (3.1)$$

In Eq.(3.1), $y \in \{0, 1\}$ is the ground truth class annotation, and $p \in [0, 1]$ is the real-valued prediction of the network for this frame. In practice, we noticed that there is an imbalance between positive and negative examples in our training data. Specifically, our training set contains more periodic segments than non-periodic ones. This, in turn, had an impact on the discriminative power of the trained models. Specifically, the trained models exhibited a bias, tending to predict most input as positive samples. To alleviate this problem, we used the weighted binary cross entropy, a straightforward extension of the BCE loss:

$$E(y, p) = -(wy \log p + (1 - w)(1 - y) \log(1 - p)). \quad (3.2)$$

In Eq.(3.2), y and p are as in Eq.(3.1), and w is a weight term balancing the two classes.

3.8 Weighted Intermediate Supervision

Supervising the network end-to-end, leads to poor true positive predictions. To deal with this problem we supervised the network at the end of every autoencoder. Using the WBCE we had to select the best weight scheme to suit our needs. For this reason we tune the weight for the first autoencoder to be more sensitive to positive examples, the second to treat the positive and negative examples equally, and the last to be more sensitive to negative examples. By using this training scheme we achieve a balanced response in positive and negative test samples. We tried to avoid more complex architectures like siamese networks. In that case we would have to define a loss function for matching the unbalanced nature of the dataset.

3.9 Testing

At run-time, we adopt a sliding window approach. More specifically, given an input video, the distance matrix M is formed. Then, a square window is centered around each point of its diagonal. The corresponding information is fed to PerLNet which returns a square window of the same size with real-valued predictions. This sliding window approach results in multiple (n_f) predictions p_i , $1 \leq i \leq n_f$ per frame f , each in the range $[0, 1]$, for the overlapping frames of consecutive windows.

A frame f is declared as being periodic if the sum of all the different predictions for exceed a predefined threshold T :

$$P_f = \sum_{i=1}^{n_f} \frac{p_i}{n_f} > T. \quad (3.3)$$

In Eq.(3.3), T was set to 0.5 in all experiments.

Chapter 4

Experimental Evaluation

The experimental evaluation of the proposed periodicity localization method was performed on two recent relevant datasets. A first category of experiments assessed the adopted design choices in an ablation study. Furthermore, the localization accuracy of PerLNet was compared to the current state of the art, for different choices of employed features. We also assessed the cross-dataset generalization of PerLNet by training our network on a dataset and testing it on another.

4.1 Training Details

We implemented PerLNet using the Keras framework [10] on top of tensorflow [1]. The Adam optimizer was used to train it for 3 epochs, with a learning rate value of 0.002. For training, we employed an Nvidia GTX 1070 Ti GPU. On that machine, each epoch took 70 seconds. The chosen range of sub-block sizes for data augmentation (see Section 3.5) was between 100 and 200, with all the sub-blocks resized to 140×140 using Lanczos resampling.

4.2 Datasets

For the evaluation of the proposed methodology, we use the PERTUBE [48] and the QUVA [57] datasets (see Fig. 4.1).

4.2.1 The PERTUBE Dataset

This dataset¹ [48] contains a set of videos depicting repetitive motions (human activities, object motions, etc) that were obtained from YouTube. There is a total of 50 annotated videos in the dataset. The videos are selected in a way that camera motion and non stationary motion is present. Each frame of every video is annotated with information that denotes whether it belongs to a periodic segment

¹Available online at <https://www.ics.forth.gr/cvrl/pd/>

or not. Each video consists of 143 to 2307 frames. The videos contain a total of 200 segments, 75 of which are periodic. Each video has from 1 to 4 periodic segments and each segment consists of 30 to 1037 frames. In total PERTUBE has 40307 frames.

4.2.2 The QUVA Dataset

This dataset² [57] is compiled for the related problem of repetition counting. It contains 100 videos depicting human activities. Each video is appropriately annotated for repetition detection. Additionally to this information, specific frames are denoted as the starting frames of a new repetition, making the annotation useful for the task of repetition counting. We disregard this information, retaining only the periodic/non-periodic annotation per frame.

4.3 Monte-Carlo Cross Validation

Neither dataset has a train/test split. Moreover the PERTUBE dataset has 40307 frames, rendering it a rather a small dataset in comparison to datasets that contain millions of frames. For this reason we adopt the Monte-Carlo Cross Validation. In Monte-Carlo Cross Validation we randomly split the dataset in half, 50% for training and 50% for testing and we cross validate the model. This process is repeated for 5 times. The performance of the model is computed as the average of these 5 cross validations.

4.4 Evaluation Metrics

We view periodicity localization as a classification problem. Thus, to evaluate the obtained results, we use the standard metrics of recall \mathcal{R} , precision \mathcal{P} , F_1 score and Overlap \mathcal{O} . In our problem, precision quantifies how many of the frames that were classified as belonging to a periodic segment are truly such. Recall quantifies the percentage of frames that actually belong to periodic segments and were correctly classified by a method.

4.5 Ablative Study

In a first set of experiments, we justified specific design choices of PerLNet based on the PERTUBE dataset. The results of these experiments are presented in Table 4.1. In each column, the best result is highlighted with bold. The results were obtained as described above in Section 4.3 by performing five randomized repetitions with half the videos of the dataset used as training and the other half as test. Then, the reported value is the average of the performance on the test set in the

²Available online at <http://tomrunia.github.io/projects/repetition/>

Table 4.1: Meta-parameter study of the proposed neural network.

Configurations	$\mathcal{R}(\%)$	$\mathcal{P}(\%)$	$F_1(\%)$	$\mathcal{O}(\%)$
PerLNet (proposed)	87.6	85.1	85.8	75.5
Init. filter size 5×5	86.6	85.2	85.3	75.6
Single autoencoder	80.7	84.0	81.6	69.5
Without skip conn.	84.6	83.7	83.6	73.0
Without interm. supervision	84.2	86.5	85.3	75.0

Table 4.2: Comparative evaluation of PerLNet with [48] on the PERTUBE dataset with CNN-based (VGG19) and hand-crafted (IDT) features. Training of the proposed method has been performed on the PERTUBE dataset.

Method	Features	$\mathcal{R}(\%)$	$\mathcal{P}(\%)$	$F_1(\%)$	$\mathcal{O}(\%)$
[48]	IDT	84.1	75.7	77.0	67.7
PerLNet	IDT	87.1	78.6	82.0	71.9
[48]	VGG19	79.2	68.1	71.1	61.8
PerLNet	VGG19	87.6	85.1	85.8	75.0

five randomized repetitions. The first line of Table 4.1 refers to the architecture as this has been described in Section 3.6. The experiment “Initial filter size 5×5 ” tests a smaller size of the first convolutional layer of the network, compared to the 11×11 of the proposed method. The “Single autoencoder” experiment uses a single autoencoder instead of the three stacked autoencoders of the proposed method. The experiment “Without skip connection” disables the ResNet-like skip connections of the proposed method. Finally, the “Without intermediate supervision” experiment explores the idea of removing the two extra terms in the loss function of the network, at the end of each autoencoder in the stack. The target is the same for all three loss terms, the ground truth classification of the sub-block. Evidently, the proposed method outperforms all other network variants. In the following, unless otherwise stated, PerLNet and the term “proposed method” refer to the architecture presented in Section 3.6 and evaluated in the first row of Table 4.1.

4.6 Comparison to SoA & the Impact of Features

The recent method by Panagiotakis et al. [48] solves the problem of periodicity localization. It does so, using a distance matrix computed on an Improved Dense Trajectories (IDT) [67] representation of each frame of the sequence. This distance

matrix is then appropriately manipulated to extract the periodic parts of the video. We note that both the method in [48] and the proposed method consist of a first phase that builds a distance matrix based on some feature representation of the frames of a video (IDT features for [48], VGG19 features for us) and a second phase that localizes periodicity on this distance matrix (hand-crafted filtering process followed by Discrete Time Warping for [48], PerLNet for us). It is therefore possible to interchange parts of the two methodologies, assessing the impact of deep neural networks on the proposed approach in a total of 4 different experiments.

In Figure 4.2, we give an example of the difference between the two feature extraction methods. It is clear that the IDT fails to detect the movement of the scene. On the other hand the matrix that has been computed by the deep features, exhibits rectangular patterns. Training the network with distance matrices that contain uninformative features with respect to the depicted periodic motions leads to poor network performance.

Table 4.2 presents the results we obtained with the 4 different variants on the PERTUBE dataset. The first column regards the employed method ([48] and PerLNet). The second column regards the employed features (Improved Dense Trajectories [67] suggested by [48], VGG19 features [32] suggested in this work).

For the approach in [48], the results were obtained by a single run over the whole dataset, using the publicly available implementation provided by the authors of that work³. For our approach, the values were estimated with the same methodology described above, using five repetitions with randomized training and test sets. For each of the employed metrics, the best result is highlighted in bold. It can be verified that the proposed approach achieves the best results compared to all the other possible configurations. Interestingly, the worst of our two variants performs better than the best of the variants of [48], that is, regardless of whether we use IDT or VGG19 features. Thus, the quality of the obtained results is attributed mostly to the method itself and to a lesser extend to the employed features.

4.7 Cross-dataset Validation

We performed a set of experiments to investigate how PerLNet generalizes across datasets. In a first experiment we trained the proposed method on the PERTUBE dataset, and assessed its performance on the QUVA dataset. In a second experiment, we swapped the training and test sets.

4.7.1 Training on QUVA, Testing on PERTUBE

Table 4.3 shows the performance of PerLNet when trained on QUVA and tested on PERTUBE. The obtained results demonstrate that there is a small performance

³The implementation of the method in [48] is available at <https://sites.google.com/site/costaspanagiotakis/research/pd>

Table 4.3: Cross-dataset evaluation: The proposed method, PerLNet, was trained on QUVA and evaluated on PERTUBE.

Methods	Features	$\mathcal{R}(\%)$	$\mathcal{P}(\%)$	$F_1(\%)$	$\mathcal{O}(\%)$
PerLNet	VGG19	89.8	74.0	80.8	67.4

Table 4.4: Cross-dataset evaluation: The proposed method was trained on PERTUBE and evaluated on QUVA. The method in [48] requires no training.

Methods	Features	$\mathcal{R}(\%)$	$\mathcal{P}(\%)$	$F_1(\%)$	$\mathcal{O}(\%)$
PerLNet	VGG19	92.1	98.5	95.1	90.9
[48]	VGG19	83.8	80.6	83.1	72.8

drop compared to training on PERTUBE and testing on PERTUBE (comparison with the fourth line of Table 4.2). The performance drop is attributed to the smaller diversity of QUVA compared to PERTUBE. Still, the proposed approach generalizes well.

4.7.2 Training on PERTUBE, Testing on QUVA

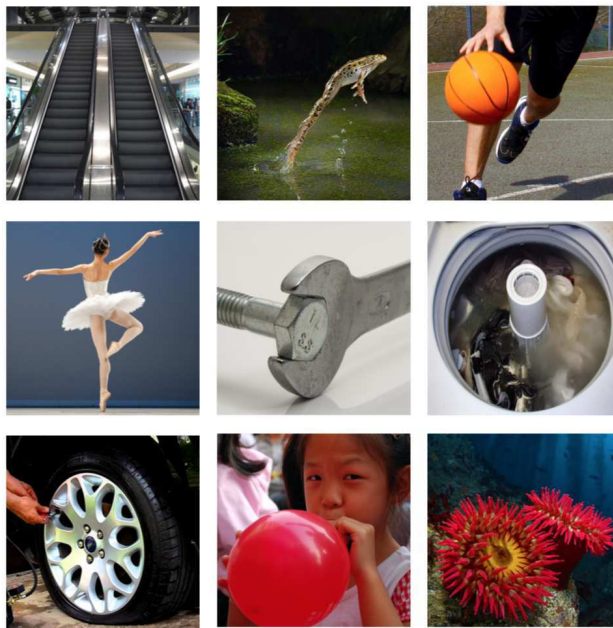
The results of this experiment are shown in Table 4.4. When training on PERTUBE, the results of testing on QUVA are better than those of testing on PERTUBE (Table 4.2, fourth row). Given that training involved the same subset of PERTUBE in both cases, this serves as a further indication that PERTUBE is more diverse than QUVA (see previous paragraph). For comparison, the second row of Table 4.4 shows the performance of the method by Panagiotakis et al. [48] on the QUVA dataset using our proposed deep features. It can be verified that the proposed approach outperforms [48] in all performance metrics and that there is a 18% increase in overlap \mathcal{O} .

4.8 Qualitative Results

Figure 4.3 and 4.4 shows the distance matrices, the periodicity localization estimation (1st green/red bar) and the ground truth (2nd green/red bar) for six sequences of the PERTUBE (top three rows) and the QUVA (bottom row) datasets. More sequences have been selected from the PERTUBE dataset because it contains more complex scenarios of multiple periodic segments per video.



(a)



(b)

Figure 4.1: Video frames from the (a) PERTUBE and (b) QUVA datasets.

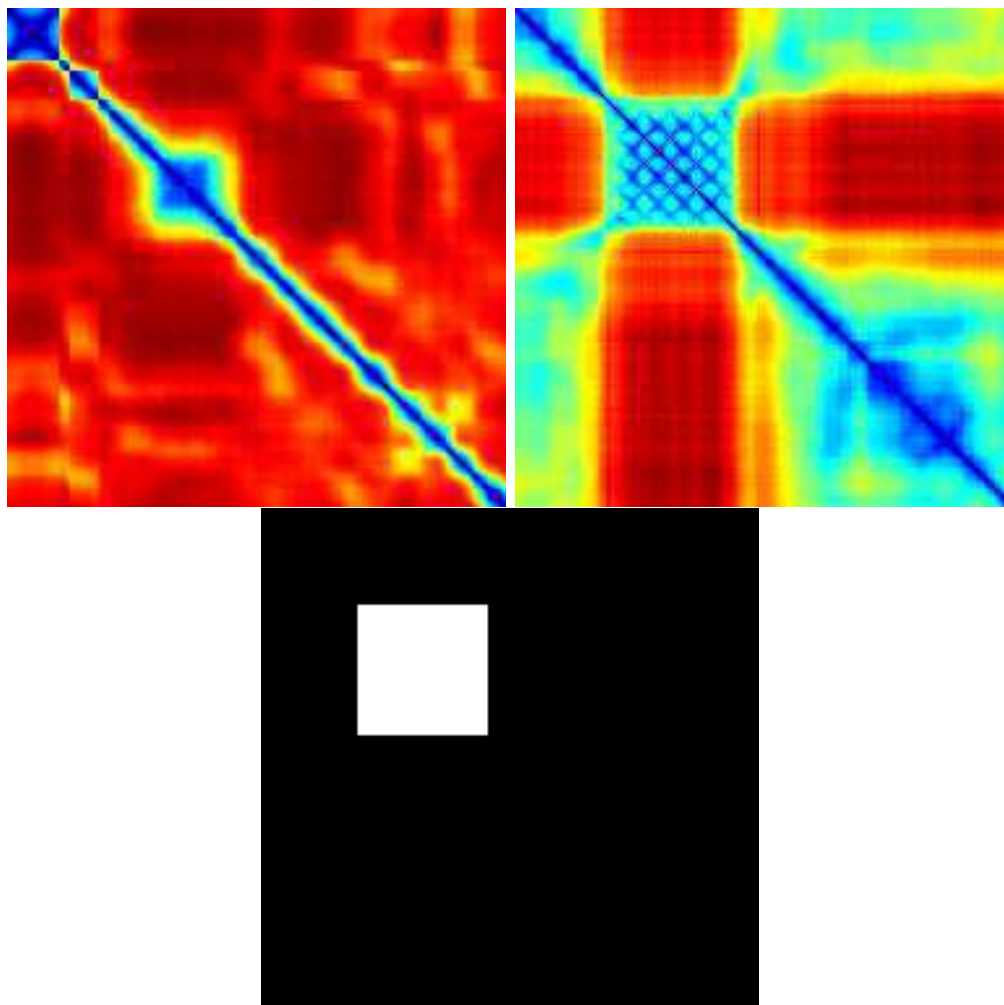


Figure 4.2: In the upper left plot we visualize the distance matrix computed with IDTs. In the upper right we visualize the distance matrix computed using deep features. The bottom plot visualizes the ground truth. With white we denote the periodic region and with black the non-periodic.

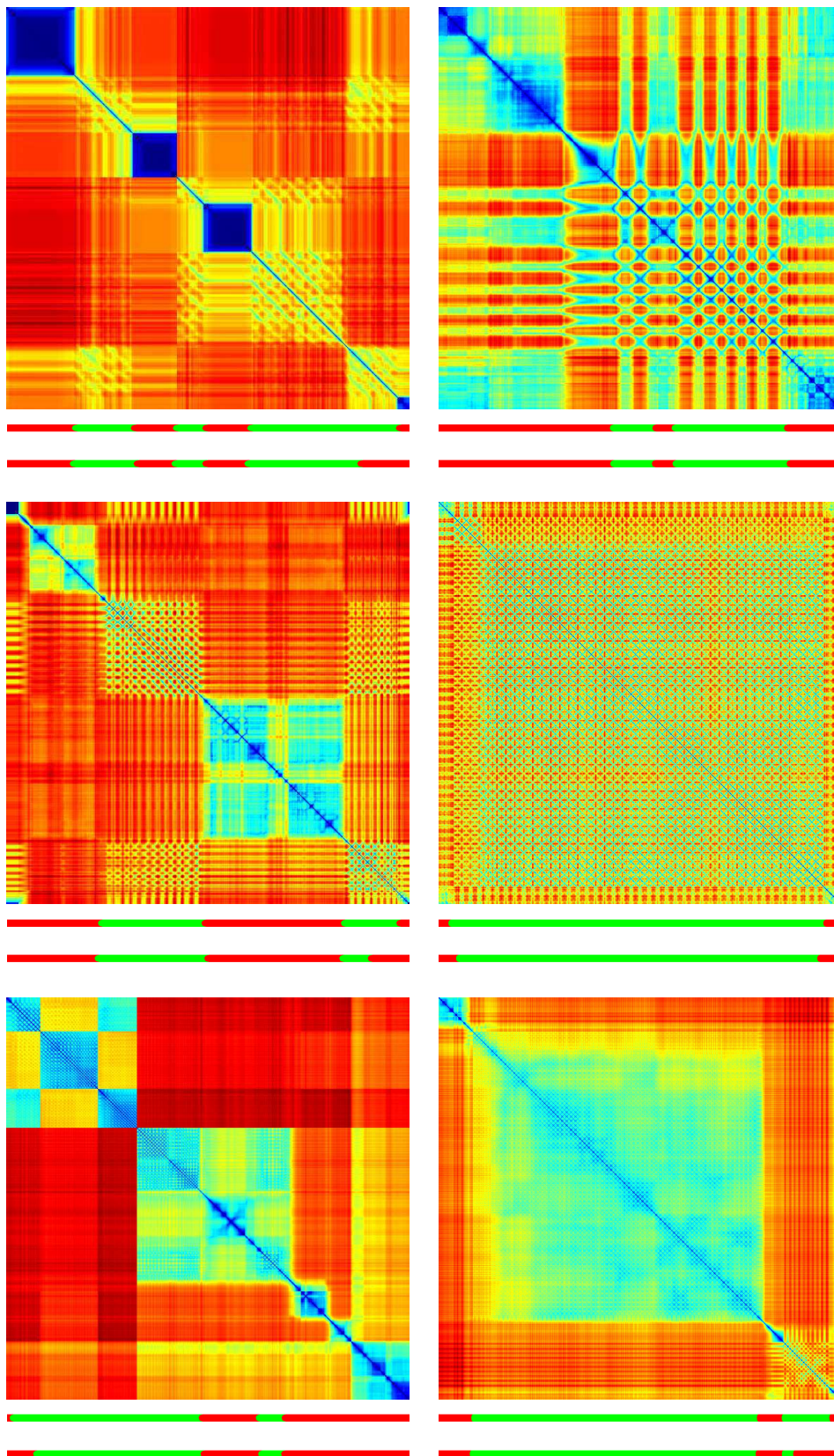


Figure 4.3: The distance matrices, the periodicity localization estimation (1st green/red bar) and the ground truth (2nd green/red bar) for six sequences of the PERTUBE dataset.

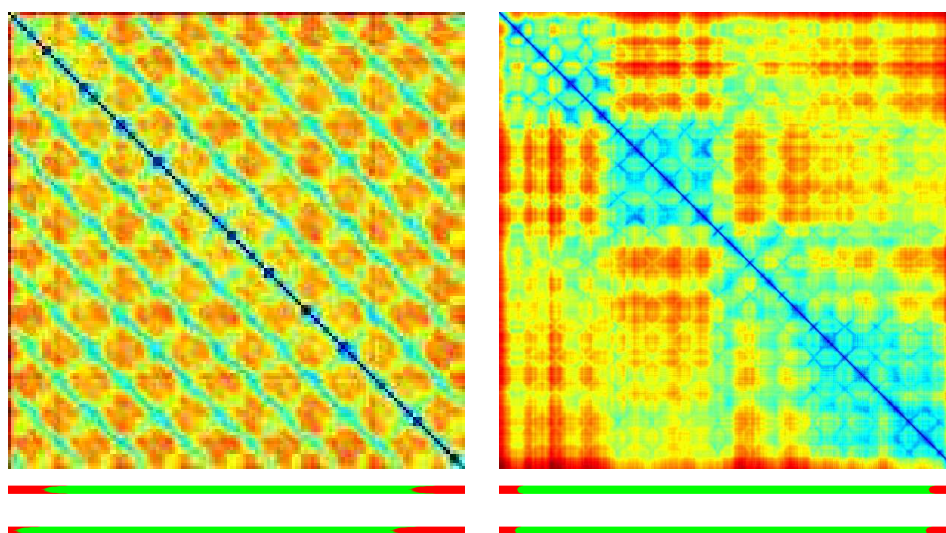


Figure 4.4: The distance matrices, the periodicity localization estimation (1st green/red bar) and the ground truth (2nd green/red bar) of the QUVA dataset.

Chapter 5

Discussion

5.1 Summary

We presented a novel method for the temporal localization of periodic segments within a video. In each frame of the video, the method extracts features using a deep neural network and then estimates a matrix of pairwise distances between video frames. Then, PerLNet, a specially designed convolutional neural network classifies the frames of the video as belonging to periodic segments or not. Extensive experiments backed the design choices behind the proposed method, verified the use of the employed features, compared the approach to a state of the art method and assessed the generalization potential across datasets. We also present a method for training the network for unbalanced datasets, using different weights for every autoencoder. This training scheme was formulated to avoid introducing a bias for the most represented class. In conclusion, using (a) an abstract representation of a video that reduced the camera effects and other sources of noise, and (b) the stacked autoencoders architecture that is trained with the proposed training scheme, we outperformed the state-of-the-art in periodicity detection and we achieved generalization, as shown by the results of the performed cross-dataset experiments.

5.2 Future Work

Although the proposed approach has several advantages, it also has a few shortcomings. The first issue regards the training of the CNN estimator on square sub-blocks. This approach leaves room for failure in the case of very small sub-block size compared to the period length, that can potentially lead to insufficient representation of the periodic features. For this reason we use a sufficiently large block size at prediction time and multiscale data augmentation during training. From a computational point of view, a shortcoming of the proposed method is that the current feature extraction method prevents its use in real-time or online applications. The current feature extraction method requires the entire video to

compute the distance matrix. Future plans include the investigation of alternatives that may allow online or even real-time operation. Tasks like industrial inspection will benefit from an extension like this.

Another extension, given the existing framework, is to tackle the problem of periodicity characterization. Then we can use it as a cue for non model based tracking of complex scenes that exhibit periodic activities, human or otherwise.

Another goal is to investigate other feature extraction algorithms that will allow for better representation of complex periodic actions. In this work we used deep features for tackling the problem of drastic scene changes. The problem with this method is that the employed VGG19 network is trained in 1000 classes, setting a limit to how well a scene can be represented. Further research is needed to explore other, potentially more powerful scene representations.

Bibliography

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [2] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns, 1995.
- [3] A. Branzan Albu, R. Bergevin, and S. Quirion. Generic temporal segmentation of cyclic human motion. *Pattern Recognition*, 41(1):6–21, 2008.
- [4] Ousman Azy and Narendra Ahuja. Segmentation of Periodically Moving Objects. *Pattern Recognition*, pages 6–9, 2008.
- [5] Serge Belongie and Josh Wills. Structure from periodic motion. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3667 LNCS:16–24, 2006.
- [6] M J Black and Paul Anandan. The robust estimation of multiple motions: Parametric and piece-wise smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996.
- [7] Alexia Briassouli and Narendra Ahuja. Extraction and analysis of multiple periodic motions in video sequences. *IEEE Trans. on PAMI*, 29(7):1244–1261, 2007.
- [8] Gertjan J. Burghouts and Jan Mark Geusebroek. Quasi-periodic spatiotemporal filtering. *IEEE Transactions on Image Processing*, 15(6):1572–1582, 2006.
- [9] Bill Chiu, Eamonn Keogh, and Stefano Lonardi. Probabilistic discovery of time series motifs. In *Proceedings of the ninth ACM SIGKDD international*

- conference on Knowledge discovery and data mining*, pages 493–498. ACM, 2003.
- [10] François Chollet et al. Keras. <https://keras.io>, 2015.
- [11] Robert T Collins, Ralph Gross, and Jianbo Shi. Silhouette-based human identification from body shape and gait. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 366–371. IEEE, 2002.
- [12] Jonathan J Crofts. Efficient method for detection of periodic orbits in chaotic maps and flows. *arXiv preprint arXiv:0706.1940*, 2007.
- [13] Ross Cutler and Larry S. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Trans. on PAMI*, 22(8):781–796, 2000.
- [14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.
- [15] PN Druzhkov and VD Kustikova. A survey of deep learning methods and software tools for image classification and object detection. *Pattern Recognition and Image Analysis*, 26(1):9–15, 2016.
- [16] Mohamed G. Elfeky, Walid G. Aref, and Ahmed K. Elmagarmid. WARP: Time warping for periodicity detection. *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 138–145, 2005.
- [17] Themis P. Exarchos, Costas Papaloukas, Christos Lampros, and Dimitrios I. Fotiadis. Mining sequential patterns for protein fold recognition. *Journal of Biomedical Informatics*, 41(1):165–179, 2008.
- [18] Tak-chung Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181, 2011.
- [19] Yifeng Gao and Jessica Lin. Efficient discovery of variable-length time series motifs with large length range in million scale time series. *arXiv preprint arXiv:1802.04883*, 2018.
- [20] Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5736–5745, 2017.
- [21] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011.

- [22] Roman Goldenberg, Ron Kimmel, Ehud Rivlin, and Michael Rudzsky. Behavior classification by eigendecomposition of periodic motions. *Pattern Recognition*, 38(7):1033–1043, 2005.
- [23] Simon Haykin. *Neural networks*, volume 2. Prentice hall New York, 1994.
- [24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.
- [25] G E Hinton, A Krizhevsky, and S D Wang. Transforming Auto-encoders. In *International Conference on Artificial Neural Networks*, 2011.
- [26] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.
- [27] Shiyao Huang, Xianghua Ying, Jiangpeng Rong, Zeyu Shang, and Hongbin Zha. Camera Calibration from Periodic Motion of a Pedestrian. *Computer Vision and Pattern Recognition*, pages 3025–3033, 2016.
- [28] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.
- [29] Katarzyna Janocha and Wojciech Marian Czarnecki. On Loss Functions for Deep Neural Networks in Classification. 25(December):49–59, 2017.
- [30] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics*, 14(2):201–211, 1973.
- [31] Giorgos Karvounas, Iason Oikonomidis, and Antonis A Argyros. Localizing Periodicity in Time Series and Videos. In *British Machine Vision Conference (BMVC 2016)*, pages 1–12, 2016.
- [32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. pages 1–9.
- [33] Mayank Lahiri and Tanya Y Berger-Wolf. Mining periodic behavior in dynamic social networks. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 373–382. IEEE, 2008.
- [34] Ivan Laptev, Serge J. Belongie, Patrick Pérez, and Josh Wills. Periodic motion detection and segmentation via approximate sequence alignment. *ICCV*, I:816–823, 2005.
- [35] Stéphane Lathuilière, Pablo Mesejo, Xavier Alameda-Pineda, and Radu Horaud. A comprehensive analysis of deep regression. *CoRR*, 2018.
- [36] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

- [37] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [38] Biao Leng, Shuang Guo, Xiangyang Zhang, and Zhang Xiong. 3d object retrieval with stacked local convolutional autoencoder. *Signal Processing*, 112:119–128, 2015.
- [39] Ofir Levy and Lior Wolf. Live repetition counting. *ICCV*, 2015 Inter:3020–3028, 2015.
- [40] Gaojian Li, Xintong Han, Weiyao Lin, and Hui Wei. Periodic Motion Detection with ROI-based Similarity Measure and Extrema-based Reference Selection. pages 947–954, 2012.
- [41] Xiaoxiao Li, Vivek Singh, Yifan Wu, Klaus Kirchberg, James Duncan, and Ankur Kapoor. Repetitive motion estimation network: Recover cardiac and respiratory signal from thoracic imaging. *arXiv preprint arXiv:1811.03343*, 2018.
- [42] Xiu Li, Hongdong Li, Hanbyul Joo, Yebin Liu, and Yaser Sheikh. Structure from Recurrent Motion: From Rigidity to Recurrency.
- [43] Fang Liu and Rosalind Picard W. Finding Periodicity in Space and Time. *ICCV*, 1998.
- [44] Chun Mei Lu and Nicola J. Ferrier. Repetitive Motion Analysis: Segmentation and Event Classification. *IEEE Trans. on PAMI*, 26(2):258–263, 2004.
- [45] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks*, pages 52–59. Springer, 2011.
- [46] Abdullah Mueen and Eamonn Keogh. Online discovery and maintenance of time series motifs. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 1089–1098, New York, NY, USA, 2010. ACM.
- [47] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked Hourglass Networks for Human Pose Estimation. In *European Conference on Computer Vision*, 2016.
- [48] Costas Panagiotakis, Giorgos Karvounas, and Antonis Argyros. Unsupervised Detection of Periodic Segments in Videos. In *International Conference on Image Processing*, 2018.
- [49] Costas Panagiotakis, Konstantinos Papoutsakis, and Antonis A Argyros. A graph-based approach for detecting common actions in motion capture data and videos. *Pattern Recognition*, 79:1–11, 2018.

- [50] Konstantinos Papoutsakis, Costas Panagiotakis, and Antonis A. Argyros. Temporal Action Co-Segmentation in 3D Motion Capture Data and Videos. In *CVPR*, pages 2146–2155, 2017.
- [51] Ulrich Parlitz and L Junge. Synchronization of chaotic systems. In *Control Conference (ECC), 1999 European*, pages 4637–4642. IEEE, 1999.
- [52] Silvia L Pinteá, Jian Zheng, Xilin Li, Paulina JM Bank, Jacobus J van Hilten, and Jan C van Gemert. Hand-tremor frequency estimation in videos. In *4th International Workshop on Observing and Understanding Hands in Action*, 2018.
- [53] E Pogalin, A W M Smeulders, and A H C Thean. Title Visual quasi-periodicity Visual Quasi-Periodicity. In *Computer Vision and Pattern Recognition*, 2008.
- [54] Ramprasad Polana and Randal C Nelson. Detection and Recognition of Periodic, Nonrigid Motion BT - Int. J. Comput. Vision. *IJCV*, 23(3):261–282, 1997.
- [55] Yang Ran, Isaac Weiss, Qinfen Zheng, and Larry S. Davis. Pedestrian detection via periodic motion analysis. *IJCV*, 71(2):143–160, 2007.
- [56] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Learning Internal Representations by Error Propagation, pages 318–362. MIT Press, Cambridge, MA, USA, 1986.
- [57] Tom F. H. Runia, Cees G. M. Snoek, and Arnold W. M. Smeulders. Real-World Repetition Estimation by Div, Grad and Curl. In *CVPR*, 2018.
- [58] Y Saiki. Numerical detection of unstable periodic orbits in continuous-time dynamical systems with chaotic behaviors. *Nonlinear Processes in Geophysics*, 14(5):615–620, 2007.
- [59] Bernard Sarel and Michal Irani. Separating transparent layers of repetitive dynamic behaviors. *ICCV*, I:26–32, 2005.
- [60] Felix Scholkmann, Jens Boss, and Martin Wolf. An efficient algorithm for automatic peak detection in noisy periodic and quasi-periodic signals. *Algorithms*, 5(4):588–603, 2012.
- [61] Joan Serrà and Josep Lluís Arcos. Particle swarm optimization for time series motif discovery. *Knowledge-Based Systems*, 92:127–137, Jan 2016.
- [62] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

- [63] Ayush Tewari, Michael Zollhofer, Hyeonwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3715–3724, 2017.
- [64] Xiaofeng Tong, Lingyu Duan, Changsheng Xu, Qi Tian, Hanqing Lu, Jinjun Wang, and Jesse S. Jin. Periodicity detection of local motion. *IEEE International Conference on Multimedia and Expo, ICME 2005*, 2005:650–653, 2005.
- [65] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408, 2010.
- [66] Michail Vlachos, Philip Yu, and Vittorio Castelli. On Periodicity Detection and Structural Periodic Similarity. In *Proceedings of the 2005 SIAM International Conference on Data Mining*. 2005.
- [67] Heng Wang and Cordelia Schmid. Action Recognition with Improved Trajectories. In *ICCV*, volume December, pages 3551–3558, 2013.
- [68] Liang Wang, Tieniu Tan, Huazhong Ning, and Weiming Hu. Silhouette analysis-based gait recognition for human identification. *IEEE transactions on pattern analysis and machine intelligence*, 25(12):1505–1518, 2003.
- [69] Quan Yuan, Wei Zhang, Chao Zhang, Xinhe Geng, Gao Cong, and Jiawei Han. Pred: periodic region detection for mobility modeling of social media users. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 263–272. ACM, 2017.