Pattern Recognition Letters 000 (2018) 1–7

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

TGLSTM: A time based graph deep learning approach to gait recognition

Francesco Battistone [a,b,*], Alfredo Petrosino [b]

[a] Mer Mec S.p.A., Vicolo Ongarie, 13 31050 Badoere di Morgano, Treviso, Italy
[b] Department of Science and Technology, University of Naples Parthenope, Naples, Italy

ARTICLE INFO

Article history:
Available online xxx

MSC:
41A05
41A10
65D05
65D17

Keywords:
Gait Recognition
Action Recognition
Skeleton
Structured data
Deep learning
LSTM
RNN

ABSTRACT

We face the problem of gait recognition by using a robust deep learning model based on graphs. The proposed graph based learning approach, named Time based Graph Long Short-Term Memory (TGLSTM) network, is able to dynamically learn graphs when they may change during time, like in gait and action recognition. Indeed, the TGLSTM model jointly exploits structured data and temporal information through a deep neural network model able to learn long short-term dependencies together with graph structure. The experiments were made on popular datasets for action and gait recognition, MSR Action 3D, CAD-60, CASIA Gait B, "TUM Gait from Audio, Image and Depth" (TUM-GAID) datasets, investigating the advantages of TGLSTM with respect to state-of-the-art methods.

1. Introduction

Using gait as a biometric is a relatively new area of study. The gait is defined as: "A particular way or manner of moving on foot". It has been receiving growing interest within the computer vision community and a number of gait metrics has been developed. Compared to other biometrics, gait has some unique characteristics [4]. The most attractive feature of gait as a biometric trait is its unobtrusiveness, i.e., the fact that, unlike other biometrics, it can be captured at a distance and without requiring the prior consent of the observed subject. Gait also has the advantage of being difficult to hide, steal, or fake. Anyway, gait recognition is still a very challenging problem: it relies on video sequences taken in controlled or uncontrolled environments; it is not invariant to the capturing viewpoint; it changes over time and is affected by clothes, footwear, walking surface, walking speed, and emotional condition [24].

Surely, among other issues, a critical step in gait recognition is feature extraction, i.e., the extraction, from video sequences that show a walking persons, of signals that can be used for recognition. This step is very important since there are numerous conceivable ways to extract signals from a gait video sequence, e.g., spatial, temporal, spatio-temporal, and frequency-domain feature extraction. Therefore, one must ensure that the feature extraction process compacts as much discriminatory information as possible.

In this paper, we propose to extract information from unstructured data such as video frames, generating structured description in terms of skeletons of persons and basic human actions that are present in the scene.

Skeletons were adopted for object recognition with success in the past [8]; actually, different approaches adopt skeleton features to classify an action. Wang et al. [45] divides the skeleton in five components and uses the spatial and temporal dictionaries of the components to represent actions. Vemulapalli et al. [43] uses rotations and translations to represent the 3D geometric relationships of body components in Lie group, and then employs Dynamic Time Warping (DTW) and Fourier Temporal Pyramid (FTP) to model the temporal dynamics, while GRUNTS [2] uses the structured feature of the skeleton to the temporal segmentation of the human action.

Some approaches use skeletal joints feature to learn a supervised model. For example, Wu and Shao [47] adopts a deep forward neural network to estimate the emission probabilities of the hidden states in HMM. Other approaches include Recurrent Neural Network (RNN) to directly classify sequences without any segmentation: Grushin et al. [19] uses LSTM-RNN for robust action recog-

* Corresponding author.
E-mail address: francesco.battistone@uniparthenope.it (F. Battistone).

https://doi.org/10.1016/j.patrec.2018.05.004
0167-8655/

nition and achieves good results on KTH dataset, Baccouche et al. [1] proposes a LSTM-RNN to recognize actions regarding the histograms of optical flow as inputs. LSTM-RNNs employed in the last two cited approaches are both unidirectional with only one hidden layer, while Lefebvre et al. [25] proposes a bidirectional LSTM-RNN with one forward hidden layer and one backward hidden layer for gesture classification. Du et al. [15], considering that human actions are composed of the motions of human body components, reports a method that uses a RNN in a hierarchical way.

To highlight the inner structure in data, we focus on processing streams of structured data by recursive neural networks [17], where the temporal processing which takes place in recurrent neural networks is extended to the case of graphs by connectionism models and where prior knowledge can be inserted to facilitate the interpretation of learnt structured data [18,33].

Our method, following the ideas proposed in [23,38], is based on a deep RNN that learns not only the skeletal joints features but also the information extracted from the changes in adjacency matrices over time. Indeed, graphs are almost always dynamic - they change shape and size - as time passes. Thus, an important part of the richness and complexity of a graph is how it changes through time; new connections are formed and old ones are broken all the time. It is the first time, at our knowledge, this feature is being considered for the problem of gait analysis and we are confident it increases its accuracy and robustness.

The model we report is named Time based Graph Long Short-Term Memory (TGLSTM); it jointly exploits structured data and temporal information through a deep neural network model able to learn long short-term dependencies together with graph structure. We demonstrate the advantage of the proposed method in both action and gait recognition, investigating the advantages of TGLSTM with respect to state-of-the-art methods on popular datasets like MSR Action 3D, CAD-60, CASIA Gait B, “TUM Gait from Audio, Image and Depth” (TUM-GAID) datasets.

The rest of the paper is structured as follows: Section 2 presents the structured features extraction used in the learning step; in Section 3 we describe the proposed Time based Graph Long Short-Term Memory (TGLSTM) model; in Section 4, we show the robustness of the TGLSTM in action recognition and gait recognition. Lastly, some conclusions are drawn also along with conducted evaluations.

2. Graph representation

The representation of the meaningful features captured at each frame has an important role in our approach. The steps to construct a graph from each frame are the following:

1. Each frame is segmented over time to extract the foreground as the moving object against a learnt background model;
2. A skeleton is constructed for each foreground;
3. Each skeleton is polygonally approximated to extract features and attach them to the graph/skeleton edges

Details about each step will be provided in the following subsections.

2.1. Background subtraction

The preferable walking in gait recognition is in a direction perpendicular to the optical axis of the capturing device since the side view of walking individuals discloses the most information about their gait. Anyway, since gait recognition should be invariant to the capturing viewpoint, the walking subject has to be separated from its background through a selective method for background subtraction producing a *foreground* for each frame. Background subtraction from colour video data is a widely studied problem, as witnessed by several recent surveys [5–7,11,32,39,49].

Among others, the SC-SOBS algorithm [30], extension of the SOBS algorithm [28], is considered one of the most powerful selective method for background subtraction. The SC-SOBS builds a neural background model of the image sequence by learning in a self-organizing manner image sequence variations, seen as trajectories of pixels in time. SC-SOBS can handle scenes containing moving backgrounds, gradual illumination variations, and camouflage, can include into the background model shadows cast by moving objects, and achieves robust detection for different types of videos taken with stationary cameras, and robustness against false detections [29].

SC-SOBS model can be described as follows. Given the colour image sequence $\{I_1, \ldots, I_T\}$, at each time instant t we build and update a neuronal map for each pixel $\mathbf{p}$, consisting of $n \times n$ weight vectors $cm_t^{i,j}(\mathbf{p})$, i, j=0, ..., n-1, which will be called the *colour model* for pixel $\mathbf{p}$ and will be indicated as $CM_t(\mathbf{p})$:

$$CM_t(\mathbf{p}) = \left\{ cm_t^{i,j}(\mathbf{p}),\ i, j = 0, \ldots, n-1 \right\}. \tag{1}$$

At each time step t, *colour background subtraction* is achieved by comparing each pixel $\mathbf{p}$ of the t-th sequence frame I_t with the current pixel colour model $CM_{t-1}(\mathbf{p})$, to determine the weight vector $BM_t^C(\mathbf{p})$ that best matches it:

$$d(BM_t^C(\mathbf{p}), I_t(\mathbf{p})) = \min_{i,j=0,\ldots,n-1} d(cm_{t-1}^{i,j}(\mathbf{p}), I_t(\mathbf{p})), \tag{2}$$

The metric $d(\cdot,\ \cdot)$ is chosen as the Euclidean distance in the HSV colour hexcone. The colour background subtraction mask for pixel $\mathbf{p}$ is then computed as

$$M_t^C(\mathbf{p}) = \begin{cases} 1 & \text{if } NCF_t(\mathbf{p}) \le 0.5 \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

where the *Neighborhood Coherence Factor* is defined as $NCF_t(\mathbf{p}) = |\Omega_{\mathbf{p}}|/|N_{\mathbf{p}}|$ [14].

Here $|\cdot|$ refers to the set cardinality, $N_{\mathbf{p}} = \{\mathbf{q} : |\mathbf{p} - \mathbf{q}| \le h\}$ is a 2D spatial neighborhood of $\mathbf{p}$ having width $(2h+1) \in \mathbb{N}$ (in the experiments h=2), and $\Omega_{\mathbf{p}}$ is the set of pixels $\mathbf{q}$ belonging to $N_{\mathbf{p}}$ that either have in their background model a best match that is close enough to their value $I_t(\mathbf{q})$ or are shadows.

An *update of the colour neuronal map* is performed in order to adapt the colour background model to scene modifications.

At each time step t, the weight vectors of CB_{t-1} in a neighborhood of the best matching weight vector $BM_t^C(\mathbf{p})$ are updated according to weighted running average.

In details, if $BM_t^C(\mathbf{p})$ is found at position $\overline{\mathbf{p}}$ in CB_{t-1}, then weight vectors of CB_{t-1} are updated according to

$$CB_t(\mathbf{q}) = (1 - \alpha_t^C(\mathbf{p}))CB_{t-1}(\mathbf{q}) + \alpha_t^C(\mathbf{p}) I_t(\mathbf{p}) \quad \forall \mathbf{q} \in N_{\overline{\mathbf{p}}}, \tag{4}$$

where $N_{\overline{\mathbf{p}}} = \{\mathbf{q} : |\overline{\mathbf{p}} - \mathbf{q}| \le k\}$ is a 2D spatial neighborhood of $\overline{\mathbf{p}}$ having width $(2k+1) \in \mathbb{N}$ (in the reported experiments k=1). Moreover,

$$\alpha_t^C(\mathbf{p}) = \gamma \cdot G(\mathbf{q} - \overline{\mathbf{p}}) \cdot (1 - M_t(\mathbf{p})), \tag{5}$$

where γ represents the learning rate, $G(\cdot) = \mathcal{N}(\cdot; \mathbf{0}, \sigma^2 I)$ is a 2D Gaussian low-pass filter with zero mean and $\sigma^2 I$ variance (in the reported experiments σ^2=0.75). The $\alpha_t^C(\mathbf{p})$ values in Eq. (5) are weights that allow to smoothly take into account the spatial relationship between current pixel $\mathbf{p}$ (through its best matching weight vector found at position $\overline{\mathbf{p}}$) and its neighboring pixels in I_t (through weight vectors at position $\mathbf{q} \in N_{\overline{\mathbf{p}}}$), thus preserving topological properties of the input in the neural network update (close inputs correspond to close outputs). In [30], $M_t(\mathbf{p})$ is the background subtraction mask value $M_t^C(\mathbf{p})$ for pixel $\mathbf{p}$, computed as in Eq. (3).

Please cite this article as: F. Battistone, A. Petrosino, TGLSTM: A time based graph deep learning approach to gait recognition, Pattern Recognition Letters (2018), https://doi.org/10.1016/j.patrec.2018.05.004

In the usual case that a set of K initial sequence frames is available for training, the above described initialization and update procedures on the first K sequence frames are adopted for training the neural network background model, to be used for detection and update in all subsequent sequence frames. For a deeper explanation of the mathematical ground behind the choice of colour model parameters, the interested reader is referred to [30].

2.2. Skeleton

Each frame-based detected foreground is skeletonized to produce a unit width skeleton. We adopt the skeletonization method [37] that does not require the iterated application of topology preserving removal operations, and does not need checking a condition specifically tailored to end point detection. Indeed, skeletonization is accomplished on the distance transform (DT) of the object, computed according to the (3,4) distance [3]. Thus, end points are automatically identified when the so called centers of maximal discs are found in DT. The skeletal pixels are all found in one raster scan inspection of DT. The set of the skeletal pixels detected in DT has all the properties expected to characterize the skeleton of the object except for unit thickness. Indeed, the set of the skeletal pixels is 2-pixel thick in correspondence of regions of the object with thickness given by an even number of pixels. Thus, a reduction to unit width is obtained by using templates able to erase the marker from suitable skeletal pixels [34]. Finally, a pruning step is also taken into account to simplify the structure of the resulting skeleton by removing some peripheral branches corresponding to scarcely elongated regions. The elongatedness of each object region can be measured by analyzing the skeleton branch mapped into it and a threshold on elongatedness can be set depending on the specific application.

2.3. Polygonal approximation and building the graph for each frame

Several approaches exist in the literature to compute the polygonal approximation of a digital curve. We use the split type approach [35] because it is convenient when working with open curves, like the individual skeleton branches. This type of algorithm can be described as follows. The two extremes of the input open curve are taken by all means as vertices of the polygonal approximation. The Euclidean distance of all points of the curve from the straight line joining the two vertices is computed. In particular, each point of the skeleton at position (x, y) and with distance value k can be interpreted as a point in the $3D$ space with coordinates (x, y, k). Then, the Euclidean distance d of a point C in the $3D$ space from the straight line joining two points A and B, is calculated by using the following expression:

$$d^2 = \|AC\|^2 - P_{ABC} * \frac{P_{ABC}}{\|AB\|^2} \tag{6}$$

where $\|AC\|$ is the norm of the vector AC, and P_{ABC} is the scalar product between vectors AB and AC.

The point with the largest distance is taken as a new vertex, provided that such a distance overcomes an a priori fixed threshold θ (for this work, $\theta = 1.5$, as we aim at a faithful approximation [36]). If a new vertex is detected, such a vertex divides the curve into two sub-curves, to each of which the above split type algorithm is applied. The splitting process is repeated as long as points are detected having distance larger than θ from the straight lines joining the extremes of the sub-curves to which the points belong.

2.4. Graph construction

When the recursion is completed, a graph with n vertexes and m undirected edges is represented as a pair $G^g = \{J^g, A^g\}$, $g = 1, \ldots, F$ and F number of frames, with $J^g \in \mathbb{R}^{n\times 3}$ and $A^g \in \mathbb{R}^{n\times n}$ (m elements greater than zero) where nodes are the points detected as vertexes and edges are the segments that best approximates the curves in the skeleton (see Fig. 1). Note that Fig. 1(b) has been cropped to the smallest area safely including the foreground once the binarized version of Fig. 1(a) has been obtained, so as to reduce the amount of data to process.

Fig. 1. A frame of the sequence (a); binary image where the foreground is the silhouette (b); the graph obtained by skeletonization and polygonal approximation (c).

Specifically, $\boldsymbol{J^g}$ in $\mathbb{R}^{3\times n}$ represents the set of vertexes that have been detected by polygonal approximation. Each vertex has three coordinates, $(x, y, DT_{(3,4)}(x, y))$, where x and y are the spatial coordinates of the vertex while $DT_{(3,4)}(x, y)$ is the value of DT in position (x, y). The third coordinate describes the thickness of the object region in correspondence to the skeleton point; indeed, graphs with similar structure may correspond to objects with different shapes.

While $\boldsymbol{A^g}$ in $\mathbb{R}^{n\times m}$ represents the topology of the graph, where $A^g_{ic} = A^g_{jc} = 1$ if the c_{th} edge starts from the i_{th} node and ends at the j_{th} node.

3. The TGLSTM model

The Time based Graph LSTM (TGLSTM) model has been designed on the basis of recurrent neural networks concepts (Fig. 2). The network is composed of some LSTM nodes alternating to Fully Connected layers such that in input a frame based skeleton graph is fed and in output the action represented by the graph is provided.

Differently from other RNN models working on action sequences, the model works in a frame-by-frame manner.

3.1. LSTMs layer

Long Short Term Memory (LSTM) networks [22] are a special kind of RNN, capable of learning long-term dependencies. It gets four neural layers (unlike the basic RNN, that has only a layer), that is interacting in a very special way. Indeed, LSTMs have special units called memory blocks. The memory blocks contain memory cells with self-connections storing the temporal cell state of the network in addition to special multiplicative units called gates to control the flow of information. A LSTM has three sigmoid gates to protect and control the cell state: *forget gate, input gate* and *output gate*.

At the first step the LSTM decides which information to throw away from the cell state. This decision is made by a sigmoid layer called the *forget gate*. It checks the previous output h_{t-1} and the input x_t to return a value in [0,1] for each sample in the cell state C_{t-1}. This output is described by:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{7}$$

where σ is a sigmoidal activation function ($\sigma(x) = 1/(1 + e^{-x})$), W_f and b_f are the weights and bias of the forget gate.

In the next step the LSTM decides what new information are storing in the cell state. It has two layers: a sigmoid layer called

Please cite this article as: F. Battistone, A. Petrosino, TGLSTM: A time based graph deep learning approach to gait recognition, Pattern Recognition Letters (2018), https://doi.org/10.1016/j.patrec.2018.05.004

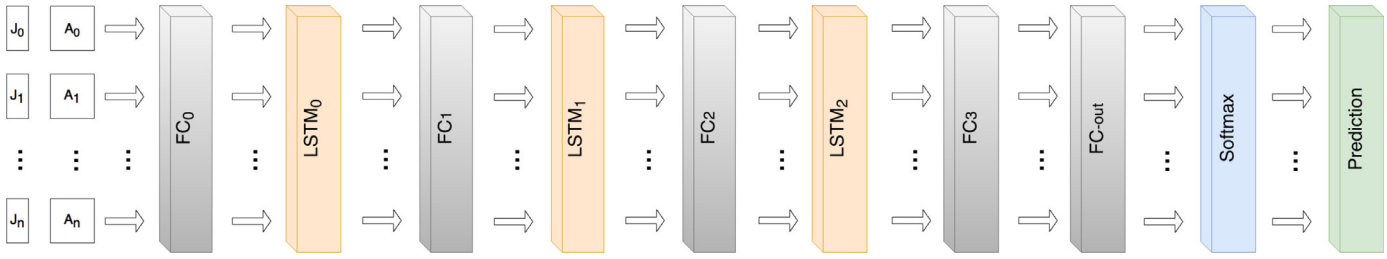

Fig. 2. The TGLSTM model.

the *input gate* that decides which values going to update,

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i); \tag{8}$$

and a tanh layer $(\tanh(x) = (exp(-x) - exp(x))/(exp(-x) + exp(x)))$ that creates a vector of new candidate values, C'_t, that could be added to the state.

$$C'_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C). \tag{9}$$

The state is updated according to the combination of both layers as follow:

$$C_t = f_t * C_{t-1} + i_t * C'_t \tag{10}$$

where f_t and i_t are respectively the weights of the old and new states at time t. Finally, the LSTM decides what it is going to output. This output will be based on its cell state, but will be a filtered version.

To resolve this task a LSTM adopts two layer: the first layer is the *output gate* that through a sigmoid activation function decides which cell states should be going to output; the second layer is a tanh layer that produces some values in $[-1, 1]$ over the new cell state and modulates the output of the sigmoid gates. These layers are described by:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{11}$$

$$h_t = o_t * tanh(C_t) \tag{12}$$

3.2. Weight regularization with LSTM

An issue with LSTMs is that they can easily overfit training data, reducing their predictive skill over testing dataset. Weight regularization (L_1 or L_2) on the weights within LSTM nodes is a technique for reducing overfitting and improving model performance. To resolve this problem different solutions have been reported:

- Insert of a droup-out layer;
- Adding noise to input;
- Adding noise to weights;
- Early training stop.

After testing all the previous solutions, the most clever and efficient strategy has been that of adding Gaussian (with $\sigma = 0.2$) noise to LSTM weights.

3.3. Fully connected layer

To design a recursive network able to process graphs, a Convolutional network based on graph is adopted as Fully Connected layer. The same has been reported in [23], although we used a different design based on the works reported in [13,21]. Provided in input the g-th graph $G^g = \{J^g, A^g\}$, a fully connected layer can be seen as mapping function such that

$$FC_l(J^g, A^g) = ReLU(\hat{A}H_{l-1}W_l) \tag{13}$$

where:

- $\hat{A}$ is the re-normalized adjacency matrix, i.e. $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$
- $\tilde{A} = A^g + I_n$, with I_n is the identity matrix of size $n \times n$;
- $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ with $0 \le i \le n$ and $0 \le j \le n$;
- W_l is the weight matrix related to the l-th FC layer;
- H_l is computed as

$$H_l = \begin{cases} J^g & l = 0 \\ LSTM_{l-1}(G^g) & l \neq 0 \end{cases} \tag{14}$$

  where $LSTM_{l-1}(G^g)$ represents the output related to the g-th graph as produced by the LSTM at the previous layer of the actual FC layer;
- $ReLU(\cdot)$ is the activation function defined as

$$ReLU(x) = max(0, x). \tag{15}$$

3.4. TGLSTM structure

The TGLSTM is characterised by Fully Connected layer (FC_l) alternating with Long Short Term Memory ($LSTM_l$), with $l = 0, \ldots, L-1$. Specifically, we adopted $L = 4$ and layers FC_0, FC_1 e FC_2 are governed by Eq. (13) and are respectively characterised by 200, 220 and 200 neurons. Also, the recursive layers $LSTM_0$, $LSTM_1$ e $LSTM_2$ have respectively neurons in number equal to 200, 220 and 200. Layer FC_{out}, differently from previous layers, map the network output to classes, according to the following Equation

$$FC_{out} = (LSTM_2 \cdot W_{out}) \tag{16}$$

where $LSTM_2 \in \mathbb{R}^{200,200}$ is the output of the second LSTM layer, $W_{out} \in \mathbb{R}^{200,m}$ are the weights and m is the number of classes (i.e. actions in the action recognition or subjects to be identified in gait recognition).

4. Experimental results

The TGLSTM has been coded in Python with the auxiliary of the following open source libraries:

- TensorFlow, that allows to develop neural networks with GPU parallelism;
- Scikit-learn, that includes a great variety of machine learning functions;
- Numpy, that is a Python scientific library;
- OpenCV, that includes structures and functions for image processing.

Two different sets of experiments have been conducted. Firstly, we demonstrate the robustness of TGLSTM for action recognition problem. Lastly, the performance of TGLSTM are shown for gait recognition. In both cases, popular datasets have been adopted and comparisons with state-of-the-art are reported and commented.

4.1. Evaluation of robustness for action recognition

We evaluated the TGLSTM on the two most cited RGB-D datasets:

Please cite this article as: F. Battistone, A. Petrosino, TGLSTM: A time based graph deep learning approach to gait recognition, Pattern Recognition Letters (2018), https://doi.org/10.1016/j.patrec.2018.05.004

Table 1
MSR action 3D dataset results.

Average on SubSets	TGLSTM	Li et al.	HOJ3D	GRUNTS	Vemulapalli	HBRNN
33% of Training set	93.7%	91.6%	**96.2%**	–	–	–
50% of Training set	**95.8%**	–	–	–	–	–
66% of Training set	96.6%	94.2%	**97.0%**	–	–	–
Cross Subject	**95.2%**	72.3%	79.0	85.3%	92.5%	94.5%

- MSR-Action3D dataset[1] is an action dataset of depth sequences captured by a depth camera and contains twenty actions. Each MSR-Action3D skeleton includes 20 junction nodes: each node is represented by a 4-tuple (u, v, d, c); (u, v) are the junction node coordinates, d is the depth and c is the belief value (anyway not considered in our work). Our performance have been compared to other recent techniques for action classification adopting depth information, like [26], HOJ3D [43,48], GRUNTS [2], HBRNN [15,44];
- CAD-60 dataset[2] contains the RGB frames, depth sequences and the tracked skeleton joint positions captured with Kinect cameras. The actions in this dataset are categorized into 5 different environments: office, kitchen, bedroom, bathroom, and living room. Three or four common activities were identified for each environment, giving a total of twelve unique actions. On this dataset TGLSTM is compared with [10,16,40,50].

Because there is no clear definition of robustness in action and in gait recognition, we consider a method as robust when it is not influenced by some conditions of environment or subject (e.g. colour of dresses).

For this reason we evaluate the TGLSTM on three different settings:

1. *Influence of Training set size* (on MSR Action 3D)
   - 33% of Training set and 66% of Testing set;
   - 50% of Training set and 50% of Testing set;
   - 66% of Training set and 33% of Testing set.

   The achieved results are shown in the first three rows of Tab.1; it figures out how the methods suffer from the training set size. The achieved performance of TGLSTM are comparable to those achieved by HOJ3D;
2. *Influence of subject* (on MSR Action 3D): the training set is composed by sequences of five actors, while the remaining actors are adopted for testing. It is a challenging way of doing training and it is described with more details in [27]. From the obtained results (last row of Table 1) it is clear the superiority of TGLSTM. The HOJ3D method, that in previous experiments got the best performance, loses in our experiments the 20% of accuracy. In general, it is useful to underline the importance of structured information, like skeletons: the methods based on skeleton like HBRNN do not degrade their performance;
3. *Influence of external environment* (on CAD-60): the dataset is usually evaluated by splitting the activities according to the environment and the global performance of the algorithm is given by the average precision and recall among all the environments. We employed leave-one-out cross-validation to test each person data, i.e. the model was trained on three out of four people from whom data were collected, and tested on the fourth one (for more details [42]). The results in Table 2 show that TGLSTM ranks at first place in terms of precision and at second place for the recall.

The achieved results show that TGLSTM results are always comparable, and in some cases the best ones, to other state-of-the-art methods even using the same model on both datasets (we change only the number of the neurons in the last fully connected layer).

Table 2
CAD-60 dataset results.

Method	Precision	Recall
[50]	93.2%	84.6%
[16]	91.1%	91.9%
[40]	93.8%	94.5%
[10]	93.9%	93.5%
TGLSTM	94.4%	93.7%

Table 3
CASIA dataset results.

Method	Exp1	Exp2	Exp3	Average Accuracy
[31]	70.2%	74.2%	58.6	67.7%
[41]	–	–	–	74.9%
Method1 in [12]	83.1%	85.5%	80.1%	82.9%
Method2 in [12]	83.9%	85.5%	81.7%	83.7%
TGLSTM	**86.1%**	**87.8%**	**85.2%**	**86.4%**

4.2. Experiments on gait recognition

We evaluated the described method on two most popular and challenging gait recognition datasets: CASIA Gait Dataset B and "TUM Gait from Audio, Image and Depth" (TUM-GAID) dataset. To obtain the graph for each frame, on these datasets, we apply the steps described in Section 2.

CASIA Dataset B[3] is a large database consisting of videos for 124 subjects captured from 11 different viewing angles (from 0^o to 180^o with the step 18^o). All the trajectories are straight and recorded indoor in the same room for all people. Besides the variety of the viewpoints, different clothing and carrying conditions are presented: wearing a coat and carrying a bag. We have 10 video sequences for each person captured from each view: 6 normal walks without extra conditions, two walks in an outdoor clothing and two walks with a bag. 90^o angle recordings have been adopted and three experiments were conducted as follows:

- **Exp1** focuses only on carrying conditions. Two normal sequences out of six are randomly selected along with the two carrying sequences. There are a total of 496 (124*x*4) sequences;
- **Exp2** pays attention to clothing changes. Two out of six normal sequences and the two coating sequences are chosen. There also exists a total of 496 (124*x*4) sequences;
- **Exp3** explores both above mentioned conditions together. Two out of six normal sequences, the two carrying and two clothing sequences form the dataset, in which 744 (124x6) gait samples are included.

Table 3 shows TGLSTM results compared with state-of-the-art methods [12,31,41]: TGLSTM gets always best performance. This holds also in the case TGLSTM is compared with similar two LSTM based methods [12], giving evidence of the great improvement in-

[1] http://research.microsoft.com.
[2] http://pr.cs.cornell.edu/humanactivities/data.php.
[3] http://www.cbsr.ia.ac.cn/english/Gait%20Databases.asp .

Please cite this article as: F. Battistone, A. Petrosino, TGLSTM: A time based graph deep learning approach to gait recognition, Pattern Recognition Letters (2018), https://doi.org/10.1016/j.patrec.2018.05.004

Table 4
TUMGAID dataset results.

Method	Accuracy
2D-CNN [9]	97.7%
3D-CNN [9]	96.7%
RSM [20]	92.0%
SVIM [46]	84.7%
GVI [46]	80.4%
[41]	97, 2%
TGLSTM	98.4%

ducted by the adoption in our method of structured data and their dynamic learning.

The TUMGAID Dataset[4] contains videos for 305 subjects going from left to right and from right to left captured from the side view. The videos show different walking conditions: 6 normal walks, 2 walks carrying a backpack, and 2 walks wearing coating shoes for each person. Although there are recordings only from one viewpoint, the number of subjects allowed to take around half of the people as training set and another half for evaluation, so that the net was trained using the data for 150 subjects and tested with the other 155 ones. The results reported in Table 4 show that the TGLSTM performance well compare with state-of-the-art methods, like [9], where instead of using video frames at 640×480, a resolution of 80×60 is used and optical flow (anyway dense and so time consuming) subsequences are passed through a CNN (non temporal as 2D-CNN or temporal 3D-CNN) to obtain gait signatures and a classification stage realized through an ensemble of one-vs-all linear SVMs.

5. Conclusions

We have introduced for the first time a Time based Graph neural network by using LSTM layers. The TGLSTM is composed by alternating recursive layers Long Short Time Memory (LSTM) and Fully Connected layers that catch information from both the graph nodes and its links.

We have assessed its performance on human activity recognition and gait analysis and specifically on four datasets MSR Action 3D, CAD-60, CASIA Gait B, "TUM Gait from Audio, Image and Depth" (TUM-GAID) datasets against some baselines, showing its superiority for supervised classification of sequences of graphs. Concerning the problem at hand, our approach deals the activity recognition frame-by-frame, but, in our opinion, interesting extension may consists in considering 3D volume of frames together with sequences of graphs.

References

[1] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, A. Baskurt, Sequential Deep Learning for Human Action Recognition, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 29–39.
[2] F. Battistone, A. Petrosino, G. Sanniti di Baja, GRUNTS: Graph Representation for UNsupervised Temporal Segmentation, Springer International Publishing, Cham, pp. 225–235.
[3] G. Borgefors, Distance transformations in digital images, Comput. Vis. Graph. Image Process. 34 (3) (1986) 344–371.
[4] N.V. Boulgouris, D. Hatzinakos, K.N. Plataniotis, Gait recognition: a challenging signal processing technology for biometric identification, IEEE Signal Process. Mag. 22 (6) (2005) 78–90.
[5] T. Bouwmans, Traditional and recent approaches in background modelling for foreground detection: an overview, Comput. Sci. Rev. 11 (2014) 31–66.
[6] T. Bouwmans, L. Maddalena, A. Petrosino, Scene background initialization: a taxonomy, Pattern Recognit. Lett. 96 (2017) 3–11, doi:10.1016/j.patrec.2016.12.024.
[7] T. Bouwmans, A. Sobral, S. Javed, S.K. Jung, E.-H. Zahzah, Decomposition into low-rank plus additive matrices for background/foreground separation: a review for a comparative evaluation with a large-scale dataset, Comput. Sci. Rev. 23 (2017) 1–71.
[8] E.R. Caianiello, A. Petrosino, Neural Networks, Fuzziness and Image Processing, Springer US, Boston, MA, pp. 355–370.
[9] F.M. Castro, M.J. Marín-Jiménez, N. Guil, S.L. Tapia, N.P. de la Blanca, Evaluation of cnn architectures for gait recognition based on optical flow maps, in: International Conference of the Biometrics Special Interest Group, BIOSIG 2017, Darmstadt, Germany, September 20–22, 2017, 2017, pp. 251–258.
[10] E. Cippitelli, S. Gasparrini, E. Gambi, S. Spinsante, A human activity recognition system using skeleton data from rgbd sensors, Comput. Intell. Neurosci. Volume 2016 (2016).
[11] C. Cuevas, R. Martínez, N. García, Detection of stationary foreground objects: a survey, Comp. Vis. Image Understanding (2016).
[12] X.L.F.Z. Dan Liu Mao Ye, L. Lin, Memory-based gait recognition, in: E.R.H. Richard C. Wilson, W.A.P. Smith (Eds.), Proceedings of the British Machine Vision Conference (BMVC), BMVA Press, 2016, pp. 82.1–82.12.
[13] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, CoRR abs/1606.09375 (2016).
[14] J. Ding, R. Ma, S. Chen, A scale-based connected coherence tree algorithm for image segmentation, IEEE Trans. Image Process. 17 (2) (2008) 204–216.
[15] Y. Du, W. Wang, L. Wang, Hierarchical recurrent neural network for skeleton based action recognition, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1110–1118.
[16] D.R. Faria, C. Premebida, U. Nunes, A probabilistic approach for human everyday activities recognition using body motion from rgb-d images, in: The 23rd IEEE International Symposium on Robot and Human Interactive Communication, 2014, pp. 732–737.
[17] P. Frasconi, M. Gori, A. Sperduti, A general framework for adaptive processing of data structures, IEEE Trans. Neural Networks 9 (5) (1998) 768–786.
[18] M. Gori, A. Petrosino, Encoding nondeterministic fuzzy tree automata into recursive neural networks, Trans. Neur. Netw. 15 (6) (2004) 1435–1449.
[19] A. Grushin, D.D. Monner, J.A. Reggia, A. Mishra, Robust human action recognition via long short-term memory, in: The 2013 International Joint Conference on Neural Networks (IJCNN), 2013, pp. 1–8.
[20] Y. Guan, C.T. Li, A robust speed-invariant gait recognition system for walker and runner identification, in: 2013 International Conference on Biometrics (ICB), 2013, pp. 1–8.
[21] D.K. Hammond, P. Vandergheynst, R. Gribonval, Wavelets on graphs via spectral graph theory, Appl. Comput. Harmon. Anal. 30 (2) (2011) 129–150.
[22] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[23] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint:arXiv 1609.02907(2016).
[24] S.H.D.H.T.D. L. Sloman, M. Berridge, Gait patterns of depressed patients and normal subjects, Am. J. Psychiatry 139 (1) (1982) 94–97.
[25] G. Lefebvre, S. Berlemont, F. Mamalet, C. Garcia, BLSTM-RNN Based 3D Gesture Classification, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 381–388.
[26] W. Li, Z. Zhang, Z. Liu, Action recognition based on a bag of 3d points, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, 2010, pp. 9–14.
[27] W. Li, Z. Zhang, Z. Liu, Action recognition based on a bag of 3d points, 2010.
[28] L. Maddalena, A. Petrosino, A self-organizing approach to background subtraction for visual surveillance applications, IEEE Trans. Image Process. 17 (7) (2008) 1168–1177.
[29] L. Maddalena, A. Petrosino, A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection, Neural Comput. Appl. 19 (2010) 179–186.
[30] L. Maddalena, A. Petrosino, The SOBS algorithm: What are the limits? in: IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), 2012, pp. 21–26.
[31] R. Martín-Félez, T. Xiang, Gait Recognition by Ranking, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 328–341.
[32] T. Minematsu, A. Shimada, R. i. Taniguchi, Analytics of deep neural network in change detection, in: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2017, pp. 1–6.
[33] C.W. Omlin, C.L. Giles, Stable encoding of large finite-state automata in recurrent neural networks with sigmoid discriminants, Neural Comput. 8 (4) (1996) 675–696.
[34] A. Petrosino, G. Salvi, A two-subcycle thinning algorithm and its parallel implementation on simd machines, IEEE Trans. Image Process. 9 (2) (2000) 277–283.
[35] U. Ramer, An iterative procedure for the polygonal approximation of plane curves, Comput. Graph. Image Process 1 (3) (1972) 244–256.
[36] A. Rosenfeld, Digital straight line segments, IEEE Trans. Comput. 23 (12) (1974) 1264–1269.
[37] G. Sanniti di Baja, Well-shaped, stable, and reversible skeletons from the (3,4)-distance transform, J. Vis. Commun. Image Represent. 5 (1) (1994) 107–115.
[38] F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Trans. Neural Netw. 20 (1) (2009) 61–80.
[39] M. Shah, J.D. Deng, B.J. Woodford, Video background modeling: recent approaches, issues and our proposed techniques, Mach. Vis. Appl. 25 (5) (2014) 1105–1119.
[40] J. Shan, S. Akella, 3d human action segmentation and recognition using pose kinetic energy, in: 2014 IEEE International Workshop on Advanced Robotics and its Social Impacts, 2014, pp. 69–75.

[4] https://www.mmk.ei.tum.de/verschiedenes/tum-gaid-database/ .

Please cite this article as: F. Battistone, A. Petrosino, TGLSTM: A time based graph deep learning approach to gait recognition, Pattern Recognition Letters (2018), https://doi.org/10.1016/j.patrec.2018.05.004

[41] A. Sokolova, A. Konushin, Gait recognition based on convolutional neural networks, 2nd International ISPRS Workshop on PSBB, 2017.
[42] J. Sung, C. Ponce, B. Selman, A. Saxena, Unstructured human activity detection from rgbd images, in: 2012 IEEE International Conference on Robotics and Automation, 2012, pp. 842–849.
[43] R. Vemulapalli, F. Arrate, R. Chellappa, Human action recognition by representing 3d skeletons as points in a lie group, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 588–595.
[44] J. Wang, Z. Liu, J. Chorowski, Z. Chen, Y. Wu, Robust 3D Action Recognition with Random Occupancy Patterns, in: A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (Eds.), Computer Vision ECCV 2012, Springer Berlin Heidelberg, 2012, pp. 872–885.
[45] J. Wang, Z. Liu, Y. Wu, J. Yuan, Mining actionlet ensemble for action recognition with depth cameras, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1290–1297.
[46] T. Whytock, A. Belyaev, N.M. Robertson, Dynamic distance-based shape features for gait recognition, J. Math. Imag. Vis. 50 (3) (2014) 314–326.
[47] D. Wu, L. Shao, Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 724–731.
[48] L. Xia, C.-C. Chen, J.K. Aggarwal, View invariant human action recognition using histograms of 3d joints., in: CVPR Workshops, IEEE Computer Society, 2012, pp. 20–27.
[49] Y. Xu, J. Dong, B. Zhang, D. Xu, Background modeling methods in video analysis: a review and comparative evaluation, CAAI Trans. Intell. Tech. (2016).
[50] Y. Zhu, W. Chen, G. Guo, Evaluating spatiotemporal interest point features for depth-based action recognition, Image Vis. Comput. 32 (8) (2014) 453–464.

Please cite this article as: F. Battistone, A. Petrosino, TGLSTM: A time based graph deep learning approach to gait recognition, Pattern Recognition Letters (2018), https://doi.org/10.1016/j.patrec.2018.05.004