Manchester Metropolitan University

https://e-space.mmu.ac.uk

2D Recurrent Neural Networks for Robust Visual Tracking of Non-Rigid Bodies

G.L. Masala, B. Golosio, M. Tistarelli, and E. Grosso.
Department of Political Science, Communication, Engineering and Information Technologies - Computer Vision Laboratory –University of Sassari (Italy),
{gilmasala,golosio,tista,grosso}@uniss.it

Abstract

The efficient tracking of articulated bodies over time is an essential element of pattern recognition and dynamic scenes analysis. This paper proposes a novel method for robust visual tracking, based on the combination of image-based prediction and weighted correlation. Starting from an initial guess, neural computation is applied to predict the position of the target in each video frame. Normalized cross-correlation is then applied to refine the predicted target position.

Image-based prediction relies on a novel architecture, derived from the Elman's Recurrent Neural Networks and adopting nearest neighborhood connections between the input and context layers in order to store the temporal information content of the video. The proposed architecture, named 2D Recurrent Neural Network, ensures both a limited complexity and a very fast learning stage. At the same time, it guarantees fast execution times and excellent accuracy for the considered tracking task. The effectiveness of the proposed approach is demonstrated on a very challenging set of dynamic image sequences, extracted from the final of triple jump at the London 2012 Summer Olympics. The system shows remarkable performance in all considered cases, characterized by changing background and a large variety of articulated motions.

Keywords: Recurrent Neural Network, Tracking, Video Analysis

1 Introduction

Motion has been one of the main cues studied in Computer Vision and Pattern Recognition. As stated by David Marr [1]: " Motion pervades the visual world, a circumstance that has not failed to influence substantially the process of evolution. The study of visual motion is the study of how information about only the organization of movement in an image can be used to make inferences about the structure and the movement of the outside world".

In many cases motion enables three-dimensional perception (consider for example the counter-rotating cylinders effect described by Ullman [2]) and, in its simplest form, explains the unrivaled ability of humans to perform scene segmentation and object tracking [3].

Object tracking can be defined as the estimation of the trajectory of an object in the image plane as it moves around a scene. Errors in tracking are often due to abrupt changes in object motion, changes in the objects appearance, non-rigidity of the object, occlusions and non-linear camera motion [4]. The robustness of the representation of target appearance, against these and other unpredictable events, is crucial to successfully track objects over time. Interestingly, assumptions are often made to constrain the tracking problem within the context of a particular application.

Recent tracking algorithms are classified into two major categories, based on the learning strategy adopted: generative and discriminative methods. Generative methods

describe the target appearance by a statistical model estimated from the previous frames. To maintain the integrity of the target appearance model, various approaches have been proposed, including sparse representation [5:9], on-line density estimation [10]. On the other hand, discriminative methods [11:13] directly implement classifiers to discriminate the target from the surrounding background. Several learning algorithms have been adopted, including on-line boosting [13], multiple instance learning [11], structured support vector machines [12] and random forests [14:15]. These approaches are often limited by the adoption of hand-crafted features for target representation, such as iconic templates, Haar-like features, histograms and others, which may not generalize well to handle the challenges arising in video sequences from everyday life scenes.

In this paper, an original method is proposed for robust visual tracking, based on a combination of image prediction and weighted correlation matching techniques. Image prediction is based on a novel recurrent neural network which can be easily generalized to track any visual pattern in dynamic scenes. The proposed approach is derived from both Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs).

A Recurrent Neural Network (RNN) is an artificial neural network with feedback connections between nodes, with the capability to model dynamic systems [16]. Elman's neural network, also known as Simple Recurrent Network (SRN), is a partially recurrent neural network first proposed by Elman[17]. Because of the context neurons and local recurrent connections between the context layer and the hidden layer, the Elman's neural network has several dynamic advantages over a static neural network. Training and convergence of SRNs usually take a long time, which makes them useless in time critical applications [10] and/or when dealing with high resolution images. Therefore, to efficiently process high resolution images, a compromise between the representation power and the dimension of the network must be sought.

A CNN is a feed-forward artificial neural network where individual neurons are arranged to respond to overlapping regions in the visual field. A CNN consists of multiple layers of small neuron collections taking as input small overlapping areas of the image. The outputs of each layer are tiled and overlapping to better represent the original image. This feature ensures a reasonable invariance to planar translation on the image plane [18]. Due to their representation power, Convolutional Neural Networks have recently attracted a considerable attention in the Computer Vision community [7], particularly for image- and video-based recognition. However, only few attempts can be found in the literature to employ CNNs for visual tracking. One reason is that off-line classifiers require a model of the objects class. On the other hand, performing on-line learning based on CNNs is not straightforward, due to the large network size and the lack of sufficiently large training sets. According to Hong at al [7], the extraction of features from the deep structure may not be appropriate for visual tracking because top layers encode semantic information and may provide a relatively poor localization accuracy.

In this paper a variation of Elman's architecture, the Two-dimensional Recurrent Neural Network (2D-RNN) is proposed. This neural architecture is derived from a CNN where the input layer captures small areas of the input image. This mapping of the image pixels allows to reduce both the training time and the network dimension, yet keeping the temporal information embedded in the video and the image details unaltered.

The paper is organized as follows: in section 2, the tracking problem is analytically stated, the solution based on the novel 2D-RNN architecture is described and compared with the Elman's SRN. A case study for video tracking (triple-jumping runner and

related dataset) is first introduced in section 3; then experimental steps and experimental protocols are defined. Section 4 is devoted to the comparison and discussion of the experimental results. Conclusions and future developments are finally discussed in section 5.

2 Object tracking in real-time video

In this paragraph we first define the tracking problem for scenes including non-rigid and articulated bodies; thence the two types of neural networks used in the experimental section , the original Elman RNN and the proposed 2D-RNN are detailed.

2.1 Tracking

In a tracking scenario, an object can be defined as "anything that is of interest for further analysis"[6]. Objects can be represented by their varying shapes and appearances; the position of a single object can be traced through a single point as the centroid or by a set of points related to a small region in the image; for example primitive geometric shapes (suitable for rigid object but also used for tracking of non rigid objects), object silhouette and contour, articulated shape models or skeletal models. In the proposed approach a primitive rectangular shape (bounding box or BB) is used. The BB has a fixed dimension for all frames of the database. Note that for the purposes of this paper, the initialization of the tracking process, for example by moving objects detection or direct object recognition, it is not explicitly considered; as a consequence the object of interest must be defined at the time step 0 by manually placing a starting BB in the first frame.

Afterward, the tracking algorithm iteratively determines the object position. At each time step t, it can be assumed that the object position has been detected in the previous $t-i$ time steps, through the centroid of the bounding box. The past i images inside the BB are fed as input to a RNN, which produces as output the prediction of the image in the bounding box for the current time step. An important outcome of such a prediction is that the expected position of the bounding box for the current time step can be evaluated and refined through the correlation between the predicted sub-image (RNN next frame prediction) and the current image. At the same time also the predicted content of the BB can be evaluated (both the dynamic background and the object of interest) by considering the residual error corresponding to the maximum of the correlation.
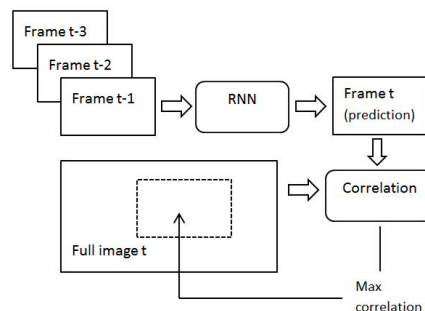

Fig. 1. The Recurrent Neural Network predicts the bounding box of the object at the next frame starting from the bounding box at the previous frame. The location refinement is performed by the correlation between the predicted bounding box and the entire image

Figure 1 depicts in detail the tracking scheme based on the RNN next frame prediction. The correlation matrix is computed by convolution in the Fourier domain; the position of the maximum of the correlation matrix corresponds to the best prediction of the BB position for the current time step. Note that, in general, the correlation matrix can have more than one local maximum, and it can happen that the target BB position is close to a local maximum that is not the absolute maximum.

This problem is particularly important when the moving object is subject to abrupt deformations, partial occlusions, etc. In order to deal with this problem, in our approach the correlation matrix is weighted with a Gaussian function centered on the extrapolated position of the moving object, based on the two most recent observations, i.e. the positions at time t−1 and t−2.

More precisely, the coordinates $X_c^{t,extr}$ and $Y_c^{t,extr}$ of the extrapolated center position are defined through :

$$X_c^{t,extr} = X_c^{t-1} + \Delta X_c^{t-1} \tag{1}$$

$$Y_c^{t,extr} = Y_c^{t-1} + \Delta Y_c^{t-1} \tag{2}$$

Where

$$\Delta X_c^{t-1} = X_c^{t-1} - X_c^{t-2} \tag{3}$$

$$\Delta Y_c^{t-1} = Y_c^{t-1} - Y_c^{t-2} \tag{4}$$

2.2 The Elman neural network

The Elman's Simple Recurrent Network (SRN) consists of an input layer, a hidden layer, a context layer, and an output layer. The outputs of the context neurons and the external input neurons are fed to the hidden neurons. Context neurons are known as memory units as they store the previous output of hidden neurons. At the time step t, the context layer nodes carry the output of hidden layer nodes of the time step t−1 iteration and supply that as input during processing of the time step t data. The SRN architecture is presented in figure Fig. 2.
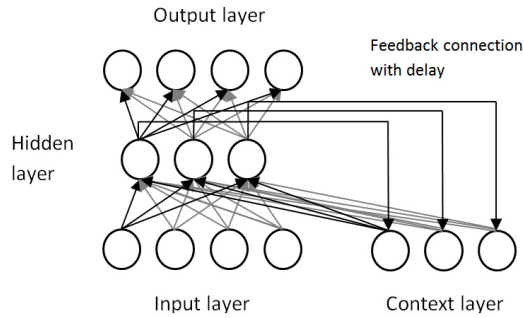

Fig. 2. Architecture of the SRN. The layers are fully connected with a feedback connection between the hidden and the context layers. The context layer provides both actual and delayed inputs to the hidden layer.

Considering I, S, C and O as input, hidden, context and output layer vectors, respectively, the vector components at the tth iteration can be written as [19]:

$$i_p^t \in I, p = 1,2,..,n \tag{5}$$
$$s_q^t \in S, q = 1,2,..,m \tag{6}$$
$$o_r^t \in O, r = 1,2,..,l \tag{7}$$
$$c_q^t = C \tag{8}$$
$$s_q^t = f(b_q^t) \tag{9}$$
$$c_q^t = s_q^{t-1} \tag{10}$$

In the above equations, n, m and l represent the numbers of nodes of input, hidden, and output layer, respectively, $f(\cdot)$ indicates the activation function of the qth hidden node at the tth iteration, while c_q^t denotes the input of the qth context layer node at the tth iteration and b_q^t is the linear output of the hidden node q at t^{th} iteration.

Let $W1$, $W2$ and $W3$ be the weight matrices between input and hidden layer, hidden and context layer and hidden and output layer , respectively. The output of hidden layer and output layer nodes at the tth iteration with these weight matrices can be represented by the following equations :

$$s_q^t = f(\sum_{p=1}^{n} w_{1qp} i_p^t + \sum_{j=1}^{m} w_{2qj} s_j^{t-1}) \tag{11}$$

$$o_r^t = f(\sum_{q=1}^{m} w_{3rq} s_q^t) \tag{12}$$

where the w_{1qp} , w_{2qj} , w_{3rq} are the elements of the weight matrices $W1$, $W2$, and $W3$, respectively.

The training of the network can be accomplished by exploiting the error back propagation algorithm [20]. In this algorithm, the error is minimized to converge to the target value by updating the link weights at each iteration using equation (13).

$$W^{new} = W^{old} + \alpha \Delta W \tag{13}$$

where α is the learning rate.

The error E expresses the difference between the set target at the output nodes and the actual output obtained as expressed in equation (14):

$$E(W) = \frac{1}{2} \sum_{k=1}^{e} \sum_{r=1}^{l} (O_r^t - o_r^t)^2 \tag{14}$$

where $O^{t,r}$ and $o^{t,r}$ represents the set target and the actual output from the network at the tth iteration, respectively, and e is the number of epochs.

2.3 Two-dimensional Recursive Neural Networks

In the proposed 2D-RNN, hidden, context and output layers are organized in two-dimensional arrays all having the same dimensions as the input image. Unlike the Elman's network, the layers of the proposed network are not fully connected to each other. In particular , denoting by (x,y) the index of row and column of the matrix of the hidden layer, respectively, 2D-RNN uses for each element (x',y') of the input matrix

also its nearest elements in the connection with the correspondent element of the hidden layer (x,y). Such type of association is replicated in the connection of the context layer with the hidden layer and in the connection between the hidden layer and the output layer, as shown in figure 3.
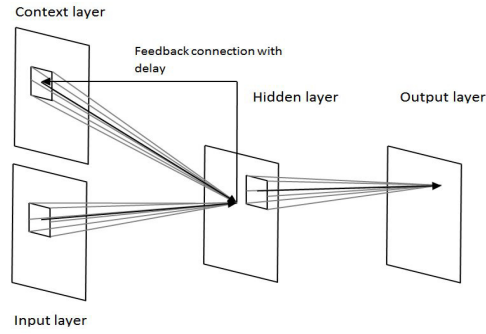

Fig. 3. Architecture of the 2D-RNN. Mapping of the image pixels from the input and context layers. Each node in the hidden layer receives input from both the actual and delayed image. Spatial information is preserved through the layers

Note that neuron (x,y) of the hidden layer is connected to all neurons (x',y') of the input layer and to all neurons (x'',y'') of the context layer with:

$$x - k \leq x' \leq x + k , y - k \leq y' \leq y + k \tag{15}$$

$$x - k \leq x'' \leq x + k , y - k \leq y'' \leq y + k \tag{16}$$

In other words, the neuron at position (x,y) of the hidden layer is connected to the corresponding neuron of the input layer and to its nearest neighbors, and to the corresponding neuron of the context layer and to its nearest neighbors. Analogously, each neuron of the output layer is connected to the corresponding neuron of the hidden layer and to its nearest neighbors.

The training of the network can be accomplished again by the standard error back propagation algorithm. Note that the k parameter (dimension of the neighborhood) is also optimized and the equations (11) and (12) are modified as follows:

$$s_{xy}^t = f(\sum_{u=x-k}^{x+k} \sum_{w=y-k}^{y+k} w_{1xyuw} \, i_{uw}^t + \sum_{u=x-k}^{x+k} \sum_{w=y-k}^{y+k} w_{2xyuw} \, S_{uw}^{t-1}) \tag{17}$$

$$o_{xy}^t = f(\sum_{u=x-k}^{x+k} \sum_{w=y-k}^{y+k} w_{3xyuw} \, S_{uw}^t) \tag{18}$$

3 Experimental results

3.1 Basic assumptions

The proposed tracking algorithm has been validated on a limited but challenging set of sequences, extracted from the final of triple jump at the London 2012 Summer Olympics. In this case study, the computation of the runner's trajectory is subject to several critical issues such as moving background, noise, articulated motion, scene illumination changes and dynamic background, as illustrated in figure 4.

Fig. 4. Frames extracted from the triple jump sequence. Several visual artifacts can be noticed, such as moving background, change in the object (the runner) shape, changes in lighting and occlusions.

Several object-tracking methods impose constraints on the motion and/or the object's appearance of objects. Most tracking algorithms assume that the object motion is smooth and without abrupt changes. Some approaches constrain the object motion to be of constant velocity or constant acceleration based on a priori information. As stated in the previous section, prior knowledge about the number and the size of objects, or the object appearance and shape, have also been used to simplify the problem. The proposed method does not make assumptions. Furthermore, it does not use any pre-processing of the image to remove external objects (i.e. TV-written), it does not apply any pre-processing such as band-pass filtering or segmentation. The developed object tracker shows a bounding box that contains the athlete in all different frames of a video, as in figure 5. The gold standard for each frame is provided through manual labeling of the region of interest and more specifically by defining the position of the pelvic bones of the athlete.

The main processing steps for the experimental phase are the following: extraction of the single JPG frames for each sequence of the MP4 video; resizing of all frames from 1280×720 pixels to 128×72 pixels; conversion of the frames to gray levels; RNNs training and testing by applying a BB of 50×50 pixels to the resized images.

3.2 Dataset

The experimental dataset is composed of 10 sequences, downloaded from the youTube platform. Each sequence relates to a different athlete in the final of the triple jump at the Olympics London 2012.

The sequences are characterized by a frame rate of 29 images/s; the dimension of each original frame is 1280×720 pixels. Each sequence has a duration of about 45s; only one frame every ten is considered for further processing, therefore for each sequence the number of frames processed varies between 97 and 127.

3.3 Configuration

A comparison between the original SRN with respect to the novel 2D-RNN is performed. Input data are the same for both networks.

The SRN can identify the single-order dynamic system using fixed coefficients in the context neurons, using weight = 1 in the delay connections with the context layer; SRN best architecture needs 2500 input, 250 hidden, 250 context and 2500 output neurons. Note that the input and output layers are related to the frame input matrix (the 50x50 pixels bounding box) while the number of neurons of the context and hidden layers have been optimized trying several configurations.

2D-RNN is not fully connected as the SRN; it requires 2500 input, hidden, context and output neurons (the numbers of neurons for all layers is fixed with respect to the frame input matrix). Best results are obtained for a number of nearest neighbors k=3, using weight=1 in the delay connections with the context layer. For both RNNs and for all neurons a logistic standard transfer functions has been adopted.

4 Results and Discussion

4.1 Performances of RNNs

In order to check the independence from the sampling of the dataset, a k-folder cross validation (5x2) has been used in the experiments. One round of cross-validation involves partitioning a sample of data in two complementary subsets, performing the analysis on one subset (train set of 5 videos), and validating the analysis on the other subset (test set of 5 videos); after that the simulation is repeated exchanging train and test sets. To reduce variability, 5 rounds of cross-validation are performed using random different partitions, and the validation results are averaged over the 10 (5X2) rounds.

In table 1 the comparison of the best configuration for both RNNs on the same random train test and blind test set is presented; learning times refer to a simple desktop architecture based on a Intel CoreTM 2 DUO CPU E 8400 @3.00 GHz and 4 GB RAM.

Parameters	SRN	2D-RNN
Input	2500	2500
Output	2500	2500
Hidden	250	2500
Context	250	2500
Learning rate	0.005	0.05
Epocs	280	130
Connections	1312500	367500
Learning time (s)	9230	1092
Best rmse	0.114	0.104

Table 1: Performance comparison of the of SRN and the 2D-RNN for the same blind test set

Table 1 clearly shows that the learning phase of 2D-RNN is faster than SRN, and 2D-RNN produces the best results. Obviously the best learning rate for both RNNs are reported,in particular, the root-mean-square deviation (rmse) is repeatedly computed on the test set after a random selection of the training set followed by the learning phase. The results for the 2D-RNN, in 5x2 cross validation,   is a mean   rmse = 0.105 ± 0.003 . In summary, Table 1   shows that   2D-RNN, compared to SRN on the same dataset, provides a better rmse; the results are stable for the 5x2 cross-validation and 2D-RNN is faster than SRN in terms of learning time and epochs. The complexity of the 2D-RNN is minor than SRN in terms of connections.

4.2   Results for tracking

Visual tracking results can be described through the distances between the center of manual annotation (the pelvic bones) of the athlete and the center of bounding box in the 2D-RNN next frame prediction, illustrated in previous figure 5. Using only one frame every ten and starting from the original frame rate information, the RNN previsions correspond to one image every 0.344 s . In figure 5 corresponding samples for the SRN (left) with actual frames and next frame prediction are shown, together with the   correlation diagram.   The same results are shown for the 2D-RNN in figure 5 (right). In the surface plot, the peak of the cross-correlation matrix occurs where the sub images are best correlated.
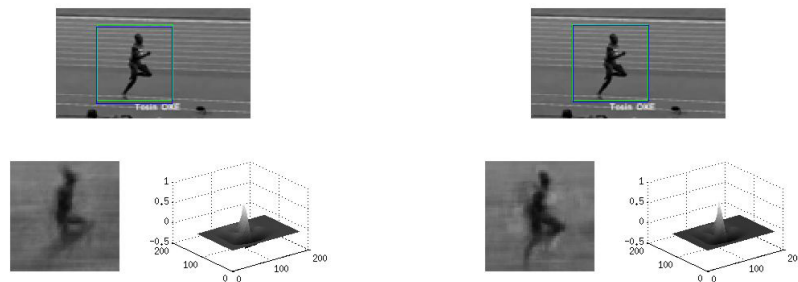

Fig. 5. Actual frame, predicted next frame and the correlation diagram computed by the SRN on the left and by the 2D-RNN on the right. The blue bounding box represents the computed gold standard while the green box represents the position computed by the RNNs.

It should be noted that all the next frames prediction in all figures are blurred because the RNN produces a distribution of positions related to the probability density. This distribution reflects the variability of the   images used to train the RNN. Naturally, the athletes move their limbs during the run in different ways. The fact that the body image is blurred is the consequence of the inability to produce an accurate prediction. On the other hand, images with clear prediction of the part of the body with respect to blurred images, would give rise to lower correlations on the average, and consequently more average errors due to the variability and not exact predictability of the next image. This is a compromise tolerable because in the tracking problem it is only necessary to have an accurate prediction of the center of the BB to follow the object of interest.

Furthermore figure 6 show a qualitative comparison between the real next frame and calculated next frame prevision of the SRN and 2D-RNN.
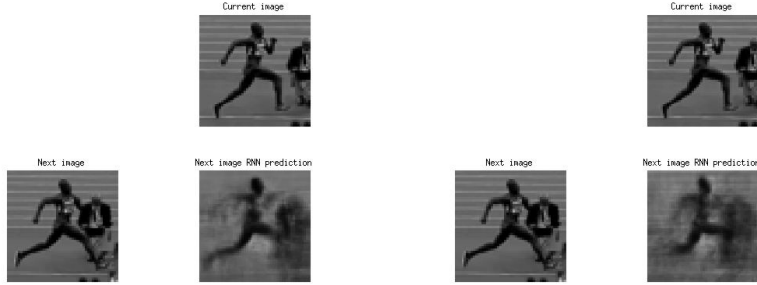
Fig. 6. Actual frame (top), next frame and predicted next frame (bottom) from the SRN on the left and 2D-RNN on the right.

In fig. 7 the diagrams of the Euclidean distances between the gold standard and the center of the RNNs prevision are shown, normalized with respect to the dimension of the BB.
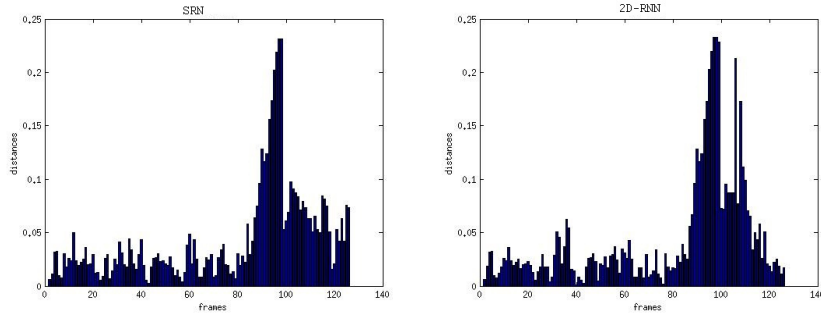

Fig. 7. Euclidean distances between the center of the computed bounding box and the gold standard for each frame, normalized with respect to the side of bounding box. The large deviation shown at about 85 frames is due to the runner landing on the sand.

The literature is divided in two types of measures for precision and recall: one is based on the localization of objects as a whole such as the F-score or other index [36] and one based on the position at pixel level. In the proposed approach object tracking is performed at pixel level. There are no lost frames in the proposed approach, therefore evaluation metrics based on accuracy is not used; however, with the aim to show the performances on the correct location of the BB, a position based measure (PBM) can be used. PBM is defined as [21]:

$$BPM = \frac{1}{N_f} \sum_i [1 - \frac{D(i)}{T_h}] \tag{19}$$

Where

$$T_h = \frac{1}{2}[(BB_w) + (BB_h)] \tag{20}$$

depends on the dimensions (width and height) of the bounding box.

In equation 19, N_f is the total number of frames considered whilst $D(i)$ is the L1-norm distance between the gold standard and the BB predicted by RNN. Using such index in our dataset the resulting mean of BMP (proposed system, first 85 frames) is expressed in the table 2. In particular it is possible to note a better performance of the 2D-RNN with respect to the SNR.

RNN	BMP (first 85 frames)
SNR	0.95±0.040
2D-RNN	0.97±0.002

Table 2: Comparison of the two RNNs tracking system in terms of BPM

Considering the entire test set, the distribution of the BB position errors between the predicted coordinates and the gold standard, in pixels, with respect to the original image, is shown in figure 8.
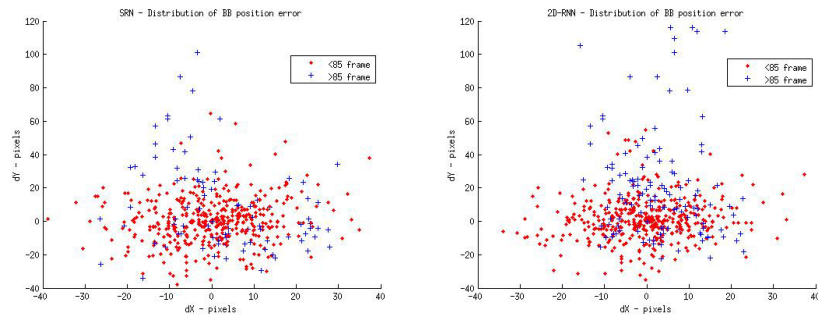

Fig.8. Scatter plots representing the distribution of the bounding box errors, between the computed box coordinates and the gold standard for each sequential frame. The red dots represent the target positions computed before the runner landing on the sand, while the blue dots represent the target positions computed after the runner landing on the sand.
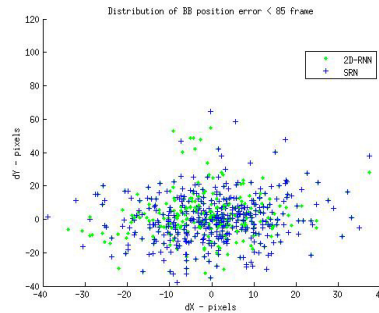

Figure 9: Scatter plot representing the distribution of the bounding box errors, between the coordinates of the computed box and the gold standard for the first 85 frames. The colors represent the method applied to compute the target position

Note that the interesting part of the sequences is composed by the first 85 frames; in fact, after frame 85 we typically register a degradation of the image due to the sand effect after landing. In figure 9 a comparative scatter plot represents the distribution of the BB position error on the original images, for both RNNs considering only the first 85 frames.

In particular from figure 9 it turns out that the most part of errors for both RNNs are within ±30 pixels with respect to the original image of 1280X720 pixels, where the dimension of the BB is 500x500 pixels; in table 3 are represented the position errors in pixels for both RNNs and the details of the coordinates of the position   mean square error (Position MSE).

RNN	Position error dX (pixels)	Position error dY (pixels)	Position MSE (pixels)
SNR	13	19	23
2D-RNN	11	14	19

Table 3: Comparison of the two RNNs position error in pixels for the first 85 frames.

An exhaustive comparison of the proposed approach with respect to other existing datasets obtained with very different aim and techniques is not simple. In particular we could not find any public database of sport scenes with measured gold standard coordinates. However, the results of table 2 can be directly compared to the BMP results reported in the paper [11] where the MILTrack algorithm, that uses a novel Online Multiple Instance Learning algorithm, is presented. In their work the authors provide a diagram with several algorithm tested on eight database for images 320x240 pixels. Normalizing the results to the scale of the adopted BB, it is possible to conclude that our algorithm, without lost frames, obtains similar performances of the best proposed MILTrack algorithm.

An alternative measure quite convenient for comparison is deviation. Deviation represents the capability of a tracker to determine the correct position of the target and measures the accuracy of tracking [12]. In particular, by using Deviation as the error of the center location expressed in pixels as a tracking accuracy measure:

$$Deviation = 1 - \frac{\sum_{i \in Ms} d(T^i, GT^i)}{|Ms|} \tag{21}$$

where $d(T^i, GT^i)$ is the normalized distance between the centroids of bounding box (BB) and the gold standard and Ms denotes the set of frames in a video where the tracked BB matches with the gold standard BB.

In the proposed approach, again normalizing with respect to the side of the BB and using the first 85 frames, for 10 sequences, a Deviation equal to about 0.98 for both RNNs is obtained. Taking into account all frames of the 10 sequences in the dataset the Deviation value slightly decreases to about 0.96 for both RNNs.

This result can be compared with the values reported in [21] and related to the articles [11:12], [22]–[37], where the target is considered tracked correctly each time the overlap between the current forecast and the real position of the object area overlap for more than 50%. As shown in Table 4, the proposed approach achieves the same or even better accuracy than the algorithms at the state of the art.

RNN	Deviation
Elman's neural network (SRN)	0.96
2D Recurrent Neural Network (2D-RNN)	0.96
Normalized Cross-Correlation (NCC) [22]	0.95
Lucas-Kanade Tracker (KLT) [23]	0.95
Kalman Appearance Tracker (KAT) [24]	0.95
Fragments-based Robust Tracking (FRT) [25]	0.94
Mean Shift Tracking (MST) [26]	0.93
Locally Orderless Tracking (LOT) [27]	0.94
Incremental Visual Tracking (IVT) [28]	0.95
Tracking on the Affine Group (TAG) [29]	0.95
Tracking by Sampling Trackers (TST) [30]	0.94
Tracking by Monte Carlo sampling (TMC) [31]	0.96
Adaptive Coupled-layer Tracking (ACT) [32]	0.94
L1-minimization Tracker (L1T) [33]	0.95
L1 Tracker with Occlusion detection (L1O) [33]	0.95
Foreground-Background Tracker (FBT) [34]	0.95
Hough-Based Tracking (HBT) [35]	0.93
Super Pixel tracking (SPT) [36]	0.93
Multiple Instance learning Tracking (MIT) [11]	0.94
Tracking, Learning and Detection (TLD) [37]	0.93
STRuck: Structured output tracking with kernels (STR) [12]	0.94

Table 4. Comparison of different approaches for target tracking applied to the jumping sequence.

5   Conclusion

A novel tracking algorithm has been presented, where two complementary RNN topologies are used   without any pre-processing of the images. The temporal memory of the recursive neural networks is used to keep the correlation among processed pixels and to perform the next frame prediction at the temporal distances of ten frames, with respect to the frame of interest.

The novel RNN algorithm proposed performs well for generic, iconic based, image tracking. This is mainly due to the two dimensional approach where for each pixel of the input image   also the information of its k nearest pixels are considered. Such kind of connection of the layers (input-hidden and hidden-output) is preferred with respect to the full connection, with great advantages in terms of rmse, learning times and BMP of the tracking.

A qualitative comparison with different approaches on different datasets is also performed, obtaining good results on measures such as deviation, that reveals an excellent performance compared to the literature.

The extension of this approach will be applied in the future to large benchmark datasets with different types of object of interest, and replacing the manual selection of the BB in the first frame with an automatic procedure designed to recognize objects belonging to predefined classes.

The results are originally measured on a triple jump dataset and could be very helpful for analysis of athlete errors in the jump in computer aided coaching or for TV highlight. However the novel method doesn't require any information related to the object of interest in the scene and it is therefore suitable for a large set of applications from sport activities to video-surveillance.

References

[1] D. Marr, Vision: A computational approach, Freeman & Co., San Francisco, 1982.

[2] S. Ullman, The interpretation of structure from motion, Proceedings of the Royal Society of London B: Biological Sciences 203 (1153) (1979) 405–426.

[3] J. J. Gibson, The Ecological Approach to Visual Perception: Classic Edition, Psychology Press, 2014.

[4] H. Denman, N. Rea, A. Kokaram, Content-based analysis for video from snooker broadcasts, Computer Vision and Image Understanding 92 (2) (2003) 176–195.

[5] A. Kokaram, F. Pitie, R. Dahyot, N. Rea, S. Yeterian, Content controlled image representation for sports streaming, Proc. of Content-Based Multimedia Indexing (CBMI05).

[6] A. Yilmaz, O. Javed, M. Shah, Object tracking: A survey, Acm computing surveys (CSUR) 38 (4) (2006) 13.

[7] S. Hong, T. You, S. Kwak, B. Han, Online tracking by learning discriminative saliency map with convolutional neural network, arXiv preprint arXiv:1502.06796.

[8] C. Bao, Y. Wu, H. Ling, H. Ji, Real time robust l1 tracker using accelerated proximal gradient approach, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 1830–1837.

[9] X. Jia, H. Lu, M.-H. Yang, Visual tracking via adaptive structural local sparse appearance model, in: Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 1822–1829.

[10] X. Mei, H. Ling, Robust visual tracking using L1 minimization, in: Computer Vision, 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 1436–1443.

[11] B. Babenko, M.-H. Yang, S. Belongie, Robust object tracking with online multiple instance learning, Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (8) (2011) 1619–1632.

[12] S. Hare, A. Saffari, P. H. Torr, Struck: Structured output tracking with kernels, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 263–270.

[13] H. Grabner, M. Grabner, H. Bischof, Real-time tracking via on-line boosting., in: BMVC, Vol. 1, 2006, p. 6.

[14] J. Gall, A. Yao, N. Razavi, L. Van Gool, V. Lempitsky, Hough forests for object detection, tracking, and action recognition, Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (11) (2011) 2188–2202.

[15] S. Schulter, C. Leistner, P. M. Roth, H. Bischof, L. J. Van Gool, On-line hough forests., in: BMVC, 2011, pp. 1–11.

[16] X. Wang, L. Ma, B. Wang, T. Wang, A hybrid optimization-based recurrent neural network for real-time data prediction, Neurocomputing 120 (2013) 547–559.

[17] J. L. Elman, Finding structure in time, Cognitive science 14 (2) (1990) 179–211.

[18] K. Korekado, T. Morie, O. Nomura, H. Ando, T. Nakano, M. Matsugu, A. Iwata, A convolutional neural network vlsi for image recognition using merged/mixed analog-digital architecture, in: Knowledge-Based Intelligent Information and Engineering Systems, Springer, 2003, pp. 169–176.

[19] S. Şeker, E. Ayaz, E. Türkcan, Elman's recurrent neural network applications to condition monitoring in nuclear power plant and rotating machinery, Engineering Applications of Artificial Intelligence 16 (7) (2003) 647–656.

[20] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd Edition, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.

[21] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, M. Shah, Visual tracking: an experimental survey, Pattern Analysis and Machine Intelligence, IEEE Transactions on 36 (7) (2014) 1442–1468.

[22] K. Briechle, U. D. Hanebeck, Template matching using fast normalized cross correlation, in: Aerospace/Defense Sensing, Simulation, and Controls, International Society for Optics and Photonics, 2001, pp. 95–102.

[23] S. Baker, I. Matthews, Lucas-kanade 20 years on: A unifying framework, International journal of computer vision 56 (3) (2004) 221–255.

[24] H. T. Nguyen, A. W. Smeulders, Fast occluded object tracking by a robust appearance filter, Pattern Analysis and Machine Intelligence, IEEE Transactions on 26 (8) (2004) 1099–1104.

[25] A. Adam, E. Rivlin, I. Shimshoni, Robust fragments-based tracking using the integral histogram, in: Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on, Vol. 1, IEEE, 2006, pp. 798–805.

[26] D. Comaniciu, V. Ramesh, P. Meer, Real-time tracking of non-rigid objects using mean shift, in: Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, Vol. 2, IEEE, 2000, pp. 142–149.

[27] S. Oron, A. Bar-Hillel, D. Levi, S. Avidan, Locally orderless tracking, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 1940–1947.

[28] D. A. Ross, J. Lim, R.-S. Lin, M.-H. Yang, Incremental learning for robust visual tracking, International Journal of Computer Vision 77 (1-3) (2008) 125–141.

[29] J. Kwon, K. M. Lee, F. C. Park, Visual tracking via geometric particle filtering on the affine group with optimal importance functions, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 991–998.

[30] J. Kwon, K. M. Lee, Tracking by sampling trackers, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 1195–1202.

[31] J. Kwon, K. M. Lee, Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 1208–1215.

[32] L. Čehovin, M. Kristan, A. Leonardis, An adaptive coupled-layer visual model for robust visual tracking, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 1363–1370.

[33] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded efficient l1 tracker with occlusion detection," in Proc. IEEE CVPR, Providence, RI, USA, 2011.

[34] H. T. Nguyen, A. W. Smeulders, Robust tracking using foreground-background texture discrimination, International Journal of Computer Vision 69 (3) (2006) 277–293.

[35] M. Godec, P. M. Roth, H. Bischof, Hough-based tracking of non-rigid objects, Computer Vision and Image Understanding 117 (10) (2013) 1245–1256.

[36] F. Yang, H. Lu, M.-H. Yang, Robust superpixel tracking, Image Processing, IEEE Transactions on 23 (4) (2014) 1639–1651.

[37] Z. Kalal, J. Matas, K. Mikolajczyk, Pn learning: Bootstrapping binary classifiers by structural constraints, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 49–56.