Space-Time Memory Networks for Multi-person Skeleton Body Part Detection

Rémi Dufour[1(✉)], Cyril Meurie[1,2(✉)], Olivier Lézoray[1,3(✉)], and Ankur Mahtani[1(✉)]

[1] FCS Railenium, 59300 Famars, France
{remi.dufour,ankur.mahtani}@railenium.eu
[2] Univ Gustave Eiffel, COSYS-LEOST, 59650 Villeneuve d'Ascq, France
cyril.meurie@univ-eiffel.f
[3] Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, Caen, France
olivier.lezoray@ensicaen.fr

Abstract. Deep CNNs have recently led to new standards in all fields of computer vision with specialized architectures for most challenges, including Video Object Segmentation and Pose Tracking. We extend Space-Time Memory Networks for the simultaneous detection of multiple object parts. This enables the detection of human body parts for multiple persons in videos. Results in terms of F1-score are satisfactory (a score of 47.6 with the best configuration evaluated on PoseTrack18 datatset) and encouraging for follow-up work.

Keywords: Space Time Memory Networks · Skeleton body part detection · Pose tracking

1 Introduction

Autonomous transportation systems, in particular autonomous cars and trains, have recently received much interest. To reach a high level grade of automation, many specific challenges need to be addressed. For autonomous trains, without any staff on board, both surveillance and security have to be performed automatically with cameras coupled with adequate computer vision algorithms. In this context, pose detection and tracking is a basic requirement of camera surveillance, that many other applications can use as input (action recognition, people counting, free seat detection, etc.). DeepPose [17] was the first Deep Neural Network (DNN) architecture for pose estimation, formulating it as a joint regression problem. Many works have extended it in several directions [2,6,18]. Rapidly, the challenging case of multi-person pose estimation has emerged where the number of persons to have their pose estimated is unknown. This is performed with either top-down or bottom-up approaches [8]. The former detects humans at a large scale and locates skeleton key-points at a smaller scale. The latter detects skeleton key-points first and skeletons parts are built from them. Bottom-up methods

M. El Yacoubi et al. (Eds.): ICPRAI 2022, LNCS 13364, pp. 78–90, 2022.
https://doi.org/10.1007/978-3-031-09282-4_7

have better scaling properties which makes them more suitable for surveillance tasks. Recently, Cao *et al.* have proposed [4] a bottom-up multi-stage refinement DNN trained with intermediate supervision. Their approach produces two outputs: body key-point parts' confidence maps and Part Affinity Fields (PAFs) that are vector fields indicating both the confidence and direction of a limb that links two body key-point parts. This approach has been extended in [10] with the use of Part Intensity Fields (PIFs). All these advances were made possible with the advent of new large scale datasets and benchmarks such as MC-COCO and PoseTrack [1,11]. Once the humans' pose skeletons have been extracted, they have to be tracked along video frames. As for pose estimation, they can be divided into top-down [7,13,19] and bottom-up [5,9,16] approaches. Top-down approaches use a person detector to obtain bounding boxes in which poses are estimated, and then track poses across time. Bottom-up approaches produce confidence maps to detect each body parts, and then group the key-points in frames (people skeletons) and across time. Raaj *et al.* have proposed an extension of PAFs with a temporal dimension, called Spatio-Temporal Affinity Fields (STAF) [16], by performing pose tracking and key-point matching across frames. Doering *et al.* [5] followed a similar direction. They built a siamese network encoding two consecutive frames to obtain belief maps, PAFs and Temporal Flow Fields (TFF) to track key-points among frames. Jin *et al.* [9] used a SpatialNet to produce key-point heat maps and key-points embeddings to group the proposals together into human pose proposals, and then used a TemporalNet to perform temporal grouping. These methods use frame-to-frame matching and do not maintain a long term memory of previous frames and estimated poses, even if this could be beneficial for performance during long surveillance tasks. At the same time, object tracking algorithms that incorporate a memory have recently been proposed within the domain of Video Object Segmentation (VOS) [15]. VOS takes a segmentation map of an object for the first frame of a video and aims at performing the segmentation for the other frames. The Space-Time Memory Networks (STM) [14] approach has recently made a breakthrough in VOS. Using the flexible memory networks system [12], it can make use of an arbitrary number of past images and predict the object segmentation in the current frame, only being limited by available memory. Pre-trained on synthetic sequences created from multiple image datasets, and then fine-tuned on video datasets such as Youtube-VOS [20] and DAVIS-2017 [15], it achieved state of the art performance when considering both quality and speed of tracking. However, maintaining a long term memory of skeleton parts along the frames of a video has still not been investigated even if this could be beneficial for performance during long surveillance tasks. In this paper, we investigate a new system for online multi-person skeleton body part detection in videos by adapting the STM architecture [14] for long-term tracking. In contrast to existing methods, our approach uses a memory of previous frames and estimated skeletons parts. This system could be integrated within most pose tracking systems by replacing its skeleton body part detection.

The paper is organized as follows. Section 2 presents our adapted STM architecture and dedicated training strategy. Section 3 presents experiments that

establish the ability of STM to detect skeleton body parts and provides results with our proposed network architecture. Last section concludes.

2 STM Multi-person Skeleton Body Part Detection

2.1 STM Architecture

The original STM architecture is one of the fastest running algorithms when it comes to single object VOS. It is also suited for the VOS task on the DAVIS-2017 dataset, in which the videos contain at most a few objects to track. It is illustrated in Fig. 1 and described in detail in [14]. Nevertheless, the STM architecture is not built to scale up easily to any number of objects, as each object has to be processed independently with a dedicated STM. This causes slower processing times and larger memory requirements when the number of objects to track increases. In this paper, we propose an adaptation of STM for detecting skeleton key-points and edges.

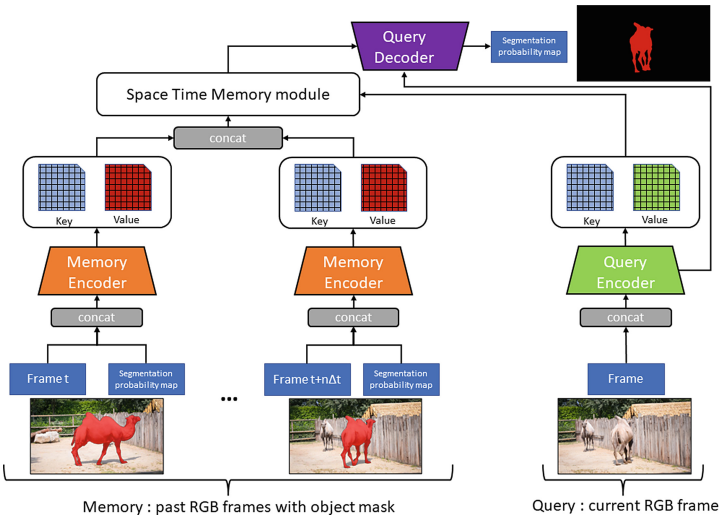

Fig. 1. Original STM architecture.

2.2 STM-skeletons Architecture

In order to adapt the STM architecture for multi-person skeleton body part detection, we have made several modifications to the original STM architecture so that several skeletons can be processed within a single inference. First, to be efficient, the proposed architecture must be able to produce several outputs in contrast to STM that produces only one segmentation map. Second, as we want to detect and track skeletons parts, we have to represent them by specific channels. This new

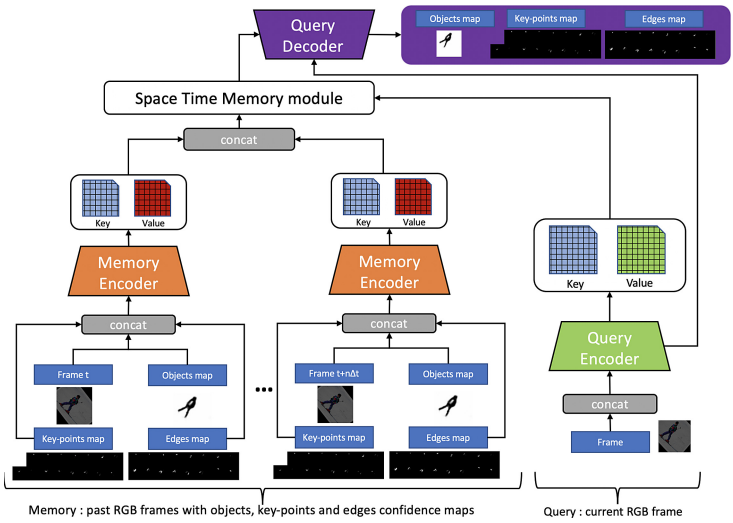

Fig. 2. Our proposed STM-skeletons architecture.

architecture is illustrated in Fig. 2. The first modifications do concern the encoder inputs. The classical STM takes an RGB frame and its segmentation probability map as concatenated inputs. The proposed architecture, called STM-skeletons, represents skeletons by both their key-points and the edges joining two key-points. The skeleton key-points and edges are each represented in dedicated confidence maps: one per skeleton key-point for all persons and one per skeleton edge for all persons. The value provides the belief that a skeleton key-point or edge of one person (among all those that appear) is present at this pixel. Therefore, we consider four different kinds of inputs that are concatenated:

- The input RGB frame (as in STM),
- A segmentation probability map for all the persons that appear in the frame (it was for only one person in the original STM),
- N_K confidence maps that represent the probability of occurence of a given skeleton key-point for all the persons,
- N_E confidence maps that represent the probability of occurence of a given skeleton edge for all the persons.

The confidence maps for the edges do not contain any orientation information as in PAFs [4] and are more likely to be called as Part Affinity Maps (PAMs). With such a modification, the encoder can deal with the simultaneous encoding of the skeleton key-points and edges of several persons. This makes the STM-skeletons model more suitable for tracking multi-person key-points and edges. As for the original STM, for the memory encoder, the input channels can either be given from a ground-truth or estimated from previous predictions. The memory encoder has its first layer of the backbone ResNet modified to be adapted to the new dimensionality of the input. For the decoder, a similar configuration to STM is kept, except

for the last layer that now produces several prediction maps instead of a single one (that was a segmentation map). Therefore, the last layer produces:s

- A segmentation probability map for all the persons that appear in the frame,
- N_K skeleton key-point confidence maps,
- N_E skeleton edge confidence maps.

This new architecture slightly sacrifices the generality of the trained model. Indeed, each output channel is specialized for one particular skeleton key-point or edge, and cannot be reused to track another person' key-points. However, this change drastically reduces the memory usage and the computation time when detecting and tracking many persons' key-points, which makes it necessary for real-world and real-time usage.

2.3 Training

Confidence Maps for Key-Points and Edges. To evaluate the performance during the training, the loss function needs a comparison with a ground-truth. We construct it in a similar manner to [4]. The ground truth confidence maps are constructed from ground truth key-points and edges. For a body part key-point j in a given skeleton k at location $x_{j,k} \in \mathbb{R}^2$, the value of the key-point confidence map $S_{j,k}$ at pixel location p is $S_{j,k}(p) = exp\left(\frac{\|p-x_{j,k}\|}{\sigma}\right)$ where σ controls the spread of the peak around the key-point. We proceed similarly for edges $E_{i,j,k}$ joining two key-points i and j in the skeleton k and generate a spread along the edge line and its extremities. The predicted confidence maps of key-points or edges are aggregated with a max operator. Ground-truth are obtained from two well know datasets: MS-COCO [11] and PoseTrack18 [1].

Hyper-parameters and Tuning. Training is done using one Nvidia Tesla V100 GPU with a batch size of 1. The used optimizer is Adam, with a learning rate of 10^{-6}. The considered losses are the MSE, the Pearson Correlation Coefficient and the Focal losses [3]. The official weights (obtained from a pre-training on several image datasets and fine-tuned on DAVIS 2017 as in [14]) are used for initialization of both STM and STM-skeletons. For the later only the layers in common with the original STM architecture are initialized with the official weights. Moreover, for all experiments the batch normalization layers are disabled as in [14].

Pre-training on Synthetic Data. The PoseTrack18 dataset features a large amount of video sequences, but the diversity of persons and contexts are inferior to large-scale image datasets such as MS-COCO. In order to leverage the large quantity of images in MS-COCO key-points for pose tracking, we created synthetic video sequences from singular images. A given sample image is translated, rotated, scaled, sheared at random N times in a cumulative way for the creation of a sequence of N frames (see Fig. 3). In the rest of the paper, pre-trainings have a duration of 5 epochs on the MS-COCO_Train set.

Fig. 3. Synthetic sequence created from a single image from MS-COCO key-points.

Fine-Tuning on Real Video Sequences. We fine-tune our models on the PoseTrack18_Train set that contains real video sequences. Training samples are created by choosing a video at random, and then taking a subset of N frames in the video sequence, keeping their ordering. We consider 30 epochs for the fine-tuning training schedule.

Data Augmentation for Refinement. For the long-term tracking, the algorithm needs to be able to correct the mistakes it can have made in the previous memorized frames in order to be able to "refine" them. Ideally, its prediction at frame T should be better than its prediction at frame $T-1$. However, the samples shown to the model during training contain ground-truth annotations, which do not feature many mistakes. In order to prepare the model to deal with these mistakes, we conceived a method similar to [6]. It consists in implementing a data augmentation scheme for refinement, where during the construction of the key-point and edge confidence maps for the memory frames, random transformations are applied. These transformations are: i) small random displacement of the ground truth key-points positions (called jitter), ii) randomization of the size, shape and orientation of the key-points' or edges' Gaussian peaks (called rand), iii) Key-points or edges false positives added to the confidence maps (called baits), iv) Randomization of the intensity of the Gaussian peaks by multiplication with a random factor in the interval $[0, 1]$ (called dull_clouds). This data augmentation scheme is illustrated in Fig. 4. In Sect. 3 we will examine the impact of these options on tracking performance.

Cyclic Training. When being used for long-term tracking, the memorization mode of our model will take as input the prediction it has made for the previous frames. However, in the normal training procedure, the confidence maps that are shown to the model are created from the ground-truth annotations. This difference between the way it is trained and the way it is meant to be used for long-term tracking might lead to worse performance. We therefore created a specific training procedure that we call "cyclic". In the latter, for a sample sequence, the model is initialized by memorizing the ground truth annotation for the first few frames, and then, for the last few frames, detects the skeletons' parts independently, based on the memory of its own predictions. In this procedure,

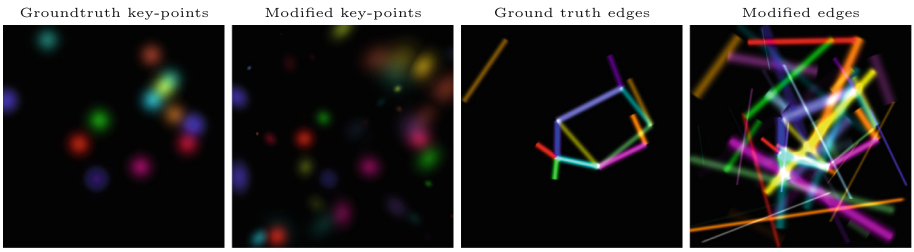

Fig. 4. Confidence maps for skeleton key-points and edges before and after data augmentation. Each key-point or edge type is shown in a particular color.

the model can produce predictions for multiple frames, therefore the training loss is the sum of the loss of each prediction.

3 Results

In this section, we experimentally show that STM architecture, without modification, can be used for the task of detecting and tracking skeleton body parts. These experiments also enable to compare different training parameters and procedures and to define the best ones. The metrics used for these experiments are precision, recall and F1-score. We can not use, at this stage, MOTA and mAP metrics generally used for a complete tracking system because we propose in this paper a component of such a system. Then, we provide results with our proposed architecture better adapted for the multi-person case. When we refer to validation datasets, these have not been used during training.

3.1 Video Skeleton Segmentation

First, we aim at showing the ability of STM to detect and track several people skeletons simultaneously. The object probability map that STM takes as input for memory frames (and produces as output for query frames) is considered to be a map that provides the belief for a pixel to belong to the skeleton of one person with possibly several persons' skeletons in the image. The STM was not intended for that as it was designed for segmenting a *single* object. As the output we want to predict is not binary (in contrast to the classical STM that outputs a binary segmentation map), we examine the impact of different loss functions on the performance: the Pearson Correlation Coefficient (CC) and focal losses (see [3] for an overview of these losses). The training procedure is the following. From the official STM weights, we do a pre-training on the synthetic MS-COCO videos (detailed in the previous section). We evaluate the trained model on the validation subset of MS-COCO key-points. We initialize the tracking with a ground truth for the first frame, and use STM to obtain the skeletons confidence map for the rest of the video sequences. We binarize the predictions with a threshold of 0.5 and compute classical performance metrics: precision, recall,

and F1-score. Table 1 presents the quantitative results. We can notice that the different losses functions led to similar results with a small advantage to the focal loss in terms of F1-score. We therefore keep the focal loss for our next experiments. On the synthetic MS-COCO_Validation the skeletons are mostly well detected. This shows that the STM architecture is able to process confidence maps instead of segmentation masks. The pre-trained model has then been used as is on the PoseTrack18_Validation. As expected the results are worse, as it was not fine-tuned on PoseTrack18_Train, but as shown in Fig. 5 the predictions results are good nevertheless, even if not very precisely located. This validates the interest of STM for detection tasks instead of segmentation.

Table 1. STM performances for producing a multi-person skeleton confidence map.

	Precision	Recall	F1-score
	MS-COCO_Validation		
CC-loss	**80.7**	64.6	71.8
MSE loss	79.1	67.4	72.8
Focal loss	77.8	**70.8**	**74.1**
	PoseTrack18_Validation		
CC-loss	**48.**0	26.7	34.3
MSE loss	46.8	27.5	34.6
Focal loss	44.3	**32.1**	**37.2**

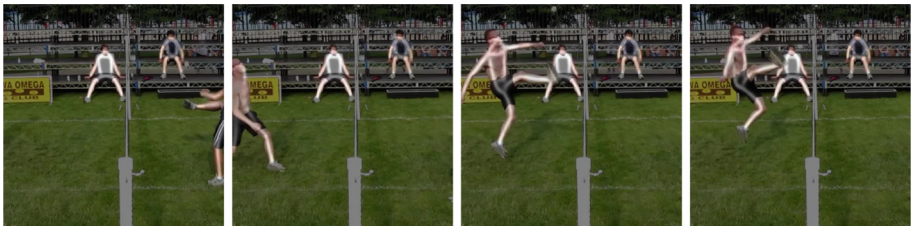

Fig. 5. STM detection results for producing a multi-person skeleton confidence map.

3.2 Video Skeleton Edge Prediction

Second, we now consider our proposed STM-skeletons architecture but to predict only edge skeleton confidence maps (i.e., the N_K confidence maps are discarded). The aim of this experiment is to show that the modification that we propose enables to detect different skeleton parts instead of a single skeleton confidence map. In addition, this will enable us to perform a fine-tuning of the model on PoseTrack18_Train. We ran multiple pre-training and fine-tuning and evaluated the resulting models on MS-COCO_Validation, and PoseTrack18_Validation.

Table 2 presents the quantitative results. Figure 6 shows some detection results, where every skeleton edge is assigned a particular color. On MS-COCO the results are better than with the original STM (in the first experiment) and shows the benefit of our approach of multi-person multi-part detection. On PoseTrack18, fine-tuning the model provides some improvement but the results are still low. This shows that the training of our model, especially for long and difficult sequences, such as these of PoseTrack18 needs a more carefully designed training.

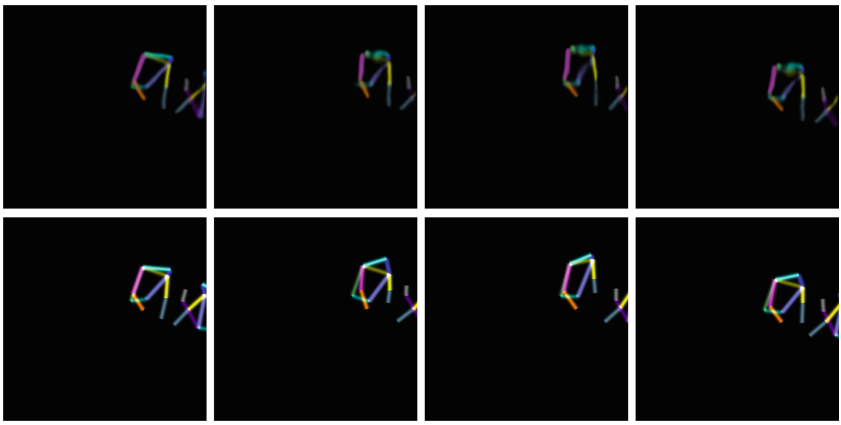

Fig. 6. Edges skeleton detection results by STM-skeletons. Top row: prediction, Bottom row: ground truth.

3.3 Video Pose Estimation

For this third experiment, we investigate more deeply different training procedures and data augmentation options for STM-skeletons to enhance the results on PoseTrack18. This time we consider the full STM-skeletons architecture that

Table 2. Proposed STM-skeletons performances for detecting multi-person edges' skeleton confidence maps.

	Precision	Recall	F1-score
	MS-COCO_Validation		
Focal loss pre-trained	88.1	**74.3**	**80.7**
Focal loss fine-tuned 30 epochs	**88.5**	65.1	75.0
	PoseTrack18_Validation		
Focal loss pre-trained	**35.2**	16.1	22.1
Focal loss fine-tuned 30 epochs	29.0	**31.2**	**30.1**

predicts both key-points and edges. We analyze the impact of different choices of training configurations, and different options for data augmentation as presented in Sect. 2.3. The models are pre-trained during 5 epochs on MS-COCO key-points, and fine-tuned during 50 epochs on PoseTrack18_Train. The evaluation is done on PoseTrack18_Validation and we use for this a dedicated metric more suited to evaluate the accuracy of the key-points prediction. We match every key-point with its closest corresponding prediction, and consider it as a True Positive if the prediction is within a radius of 10 pixels from the ground truth. Edges are not considered in the evaluation results. Results are shown in Table 3. Several data augmentation configurations are considered and each checkmark tells which one is considered. We also look at the influence of fine-tuning. When comparing the same configurations, before and after fine-tuning, we notice a systematic improvement, this shows the importance of fine-tuning on the real video sequences from PoseTrack18_Train. If we compare Configurations 1 and 2 with the others, we can see that cyclic training provides a significant improvement, and on its own is almost enough to replace data augmentation. This shows that it is important to perform memorization not only with the ground truth but also with predictions. The augmentations rand, baits, and jitter can be considered as useful options that show a consistent improvement, in particular when cyclic training is disabled. These options are required to obtain the best performance measured in terms of F1-score, obtained after fine-tuning with Configuration 5.

Table 3. Performances of STM-skeletons with different pre-training procedures, and different data augmentation options.

	Cyclic	rand	baits	jitter	dull_clouds	Precision	Recall	F1-score
Without fine-tuning						PoseTrack18_Validation		
Configuration 1						**82,6**	6.1	11.4
Configuration 2		✓	✓	✓		57.4	25.9	35.7
Configuration 3	✓					67.3	25.7	37.2
Configuration 4	✓	✓				46.9	28.2	35.2
Configuration 5	✓	✓	✓	✓		52.1	**31.0**	**38.9**
Configuration 6	✓	✓	✓	✓	✓	48.7	27.3	35.0
With fine-tuning						PoseTrack18_Validation		
Configuration 1						**81.3**	17.9	29.4
Configuration 2		✓	✓	✓		73.1	31.8	44.4
Configuration 3	✓					70.9	35.0	46.8
Configuration 4	✓	✓				62.3	**36.4**	46.0
Configuration 5	✓	✓	✓	✓		69.8	36.2	**47.6**
Configuration 6	✓	✓	✓	✓	✓	69.8	32.2	44.0

The precision, recall and F1-score are good aggregate metrics to compare the performance of different training procedures, however, they do not inform us

on whether the differences are in long-term or short-term tracking. In order to make sure that our training procedure is advantageous for long term tracking, we compare in Fig. 7 the evolution of recall scores over time, on sequences from PoseTrack18_Validation. We can see that the differences in the first few frames are negligible, and that they increase over time. This shows that fine-tuning on PoseTrack18_Train and cyclic training improve long-term rather than short-term tracking performance.

Finally, to obtain better detection results, the proposed multi-person skeleton body part detection has to be included within a complete tracking system as in [4] where the detection is processed with non-maximum suppression, body part detection association and skeleton matching across frames. This will obviously further enhance the predictions our architecture gives.

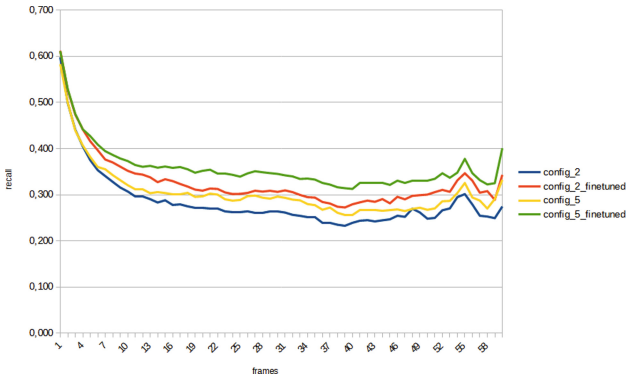

Fig. 7. Evolution of recall over time depending on model training configuration.

4 Conclusion

We have proposed a new algorithm for multi-person skeleton body part detection. Building up on the recent Video Object Segmentation architecture called Space-Time Memory Networks, we have modified it so that it is adapted to multi-person skeletons key-points and edge prediction. We have designed a two-stage pre-training/fine-tuning procedure for this architecture that aims at improving the capacities of the model. In addition we use a specific data augmentation and a cyclic training scheme. The impact of these different elements has been evaluated on the PoseTrack18 dataset. While at this stage the results cannot yet be compared to the state-of-the-art of skeleton pose estimation in videos (as several additional steps for filtering and matching have to be done), we have shown that our method can be interesting. In particular, in contrast to existing approaches, the proposed architecture can make use of a long-term memory.

Acknowledgments. This research work contributes to the french collaborative project TASV (autonomous passengers service train), with SNCF, Alstom Crespin, Thales, Bosch, and SpirOps. It was carried out in the framework of FCS Railenium, Famars and co-financed by the European Union with the European Regional Development Fund (Hauts-de-France region).

References

1. Andriluka, M., et al.: PoseTrack: a benchmark for human pose estimation and tracking. In: CVPR, pp. 5167–5176 (2018)
2. Belagiannis, V., Zisserman, A.: Recurrent human pose estimation. In: FG, pp. 468–475 (2017)
3. Bruckert, A., Tavakoli, H.R., Liu, Z., Christie, M., Meur, O.L.: Deep saliency models?: the quest for the loss function. Neurocomputing **453**, 693–704 (2021)
4. Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., Sheikh, Y.A.: OpenPose: realtime multi-person 2D pose estimation using part affinity fields. IEEE Trans. Pattern Anal. Mach. Intell. **43**(1), 172–186 (2019)
5. Doering, A., Iqbal, U., Gall, J.: JointFlow: temporal flow fields for multi person pose estimation. In: BMVC, pp. 261–272 (2018)
6. Fieraru, M., Khoreva, A., Pishchulin, L., Schiele, B.: Learning to refine human pose estimation. In: CVPR, pp. 318–327 (2018)
7. Girdhar, R., Gkioxari, G., Torresani, L., Paluri, M., Tran, D.: Detect-and-track: efficient pose estimation in videos. In: CVPR, pp. 350–359 (2018)
8. Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., Schiele, B.: DeeperCut: a deeper, stronger, and faster multi-person pose estimation model. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 34–50. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_3
9. Jin, S., Liu, W., Ouyang, W., Qian, C.: Multi-person articulated tracking with spatial and temporal embeddings. In: CVPR, pp. 5657–5666 (2019)
10. Kreiss, S., Bertoni, L., Alahi, A.: PifPaf: composite fields for human pose estimation. In: CVPR, pp. 11977–11986 (2019)
11. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
12. Miller, A., Fisch, A., Dodge, J., Karimi, A.H., Bordes, A., Weston, J.: Key-value memory networks for directly reading documents. In: EMNLP, pp. 1400–1409 (2016)
13. Ning, G., Huang, H.: LightTrack: a generic framework for online top-down human pose tracking. In: CVPR, pp. 4456–4465 (2020)
14. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Video object segmentation using spacetime memory networks. In: ICCV, pp. 9225–9234 (2019)
15. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 Davis challenge on video object segmentation. arXiv:1704.00675 (2017)
16. Raaj, Y., Idrees, H., Hidalgo, G., Sheikh, Y.: Efficient online multi-person 2D pose tracking with recurrent spatio-temporal affinity fields. In: CVPR, pp. 4620–4628 (2019)
17. Toshev, A., Szegedy, C.: DeepPose: human pose estimation via deep neural networks. In: CVPR, pp. 1653–1660 (2014)

18. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: CVPR, pp. 4724–4732 (2016)
19. Xiu, Y., Li, J., Wang, H., Fang, Y., Lu, C.: Pose flow: efficient online pose tracking. In: BMVC, pp. 53–64 (2018)
20. Xu, N., et al.: Youtube-VOS: A large-scale video object segmentation benchmark. arXiv:1809.03327 (2018)