Action Recognition with Fusion of Multiple Graph Convolutional Networks

Camille Maurice, Frédéric Lerasle

Action Recognition with Fusion of Multiple Graph Convolutional Networks

Camille Maurice
LAAS-CNRS
7 avenue du Colonel Roche, 31400, Toulouse, France
c.maurice@laas.fr

Frédéric Lerasle
University Paul Sabatier
Route de Narbonne, 31330, Toulouse, France
lerasle@laas.fr

Abstract

We propose two light-weight and specialized Spatio-Temporal Graph Convolutional Networks (ST-GCNs): one for actions characterized by the motion of the human body and a novel one we especially design to recognize particular objects configurations during human actions execution. We propose a late-fusion strategy of the predictions of both graphs networks to get the most out of the two and to clear out ambiguities in the action classification. This modular approach enables us to reduce memory cost and training times. Moreover we also propose the same late fusion mechanism to further improve the performance using a Bayesian approach.

We show results on 2 public datasets: CAD-120 and Watch-n-Patch. Our late-fusion mechanism yields performance gains in accuracy of respectively +21 percentage points (pp), +7 pp on Watch-n-Patch and CAD-120 compared to the individual graphs. Our approach outperforms most of the significant existing approaches.

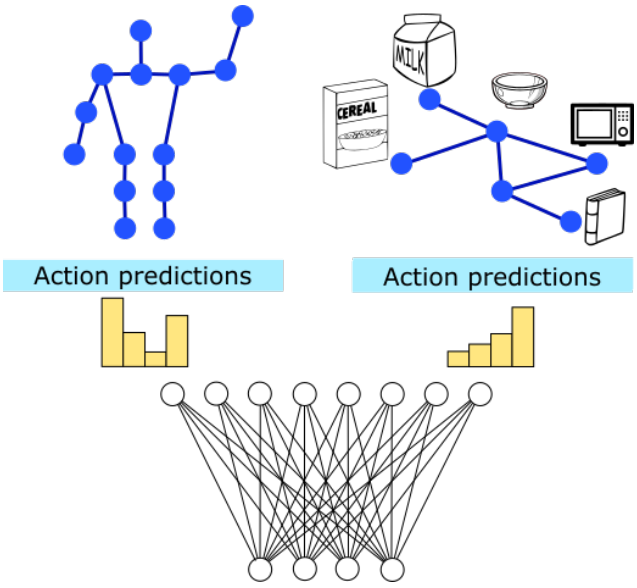

Figure 1: Skeleton-graph on the left and object-graph on the right. Their action predictions are concatenated and connected to a fully connected layer for late fusion decision.

1. Introduction

Human activity recognition is an important task in the development of many practical applications such as home health monitoring, human-robot interaction, among others. An activity can be seen as a sequence of actions occurring in the same environment *e.g.*: the activity *prepare a bowl of cereals* involves actions such as *add cereals* to the bowl, *pour* milk from the bottle to the bowl and maybe *place* a spoon in the bowl. Activities involved at home or in an industrial environment may differ but their underlying actions may be similar as they cover object displacement, object grasping, object interaction... Indeed, these atomic actions concern the movement of objects, their grasp, or the interactions they may have with their environment. Therefore, we are interested in the recognition of action based on the human pose involved as well as the objects in interaction. In this context the same object appear across multiple actions but its interactions with other objects or human limbs differ.

Data-driven approaches based on convolutional neural networks (CNN) adapted to the video domain with 3D convolutions allow the recognition of actions in video streams. 3D convolutional neural networks learn spatio-temporal features simultaneously. Approaches like C3D [27] ob-

tain an accuracy of 90.4% in the action recognition dataset UCF101 [26]. However, 3D convolutions increase the size of the network and thus the number of parameters to learn and they often rely on large datasets collected on YouTube. Often few to no objects are present nor in common across the videos or across different actions labels. Thus they often rely on large datasets for training such as Kinetics and UCF101 that are created from videos collected on YouTube. The different classes are performed in radically different environments with no to very few objects present in the scene. For example in *swimming* no objects are present and in *playing guitar* only the guitar is present. These are not suitable for daily living action recognition where the same objects in the same environment are used to perform different actions.

Recent advances in pose estimation and in object detection allow us to extract out of video streams time series of skeleton joints and objects nature and positions. We believe these are the main cues that enable to detect actions. Thus, we aim to perform action recognition based on higher-level information from pre-processed videos with such detectors. Since the introduction of Graph Convolutional Networks (GCN) by Kipf and Welling [10] in 2017, GCNs gained in popularity among the Computer Vision community. Especially in the action recognition topic where the human skeleton joints and limbs naturally define a graph. We propose a modular framework for action recognition based on GCNs. In addition to a GCN based on a temporal skeleton graph similar to [29], we consider objects in the scene as well as another graph, where a connection exists if they can interact. We are able to model those time sequences of 2D positions in the image as spatio-temporal graphs and achieve recognition via GCN. As a result we obtain a much lighter network yet efficient compared either to graph neural networks or classical 3D CNN.

Our previous work [19] spotlights the late fusion of heterogeneous classifiers: parametric (knowledge-driven) and CNN (data-driven) ones. Here we study the late fusion of a Bayesian approach to our new graph convolution networks and demonstrate that such graph networks outperform clearly 3D CNN ones.

This modular framework shines a light on the importance of object interactions to clear-up ambiguities in the skeleton-based GCN human action prediction. Thus our two major contributions are: (1) The design of a object graph to model inter-objects interactions occurring during the execution of an action. (2) A modular approach with a late fusion mechanism to get the prediction out of both the skeleton-based and objects-based graphs. As an extension we propose: (3) the same late fusion mechanism, *i.e*. with a bayesian approach to further improve performance using a knowledge-driven (parametric model) based on a Bayesian framework. We observe that when two models complement each other, the late fusion mechanism leads to a substantial performance gain. Overall, with performance gain we also show better inference speed and fewer parameters to learn.

2. Related Work

Early approaches and datasets focus solely on human perception and its pose trajectories in video sequences for action classification. Often, each video in those datasets contains a single human action without any context: no backgrounds nor surrounding objects. Still today, most of the largest datasets focus on skeleton-based action detection with few to no objects to interact with, *e.g*.: MSRAction [14], BerkeleyMHAD [21], Human3.6M [7], NTU RGB+D [25]. Some datasets feature objects but they are involved in only one action. Thus the action can be recognized solely from the object presence *e.g*. in UCF101 [26] a guitar appears only in the sequence *to play guitar*. We also observe this in the action recognition dataset HMDB-51 [12], where actions does not occur in sequences. For example a brush appears only in the videos belonging to the action *to brush hair* and a club only appears in the videos corresponding to *to play golf*. For these reasons those dataset are not suitable to evaluate our model.

Action recognition is widely addressed by recurrent neural networks (RNN) and 3D convolutional neural networks [27, 2] to learn features and capture long term dependencies directly from video streams. They simultaneously extract temporal and spatial features at the 3D convolution layers stage. Thus they do not explicitly reason on skeleton and objects involved. For this reason, they require the learning of a huge number of parameters *e.g*. 13M for C3D and are poorly suited to small datasets.

Jain *et al*. [8] propose a Structural Recurrent Neural Network (S-RNN) with pre-defined spatio-temporal graph structures. They propose a mapping from a spatio-temporal graph to a mixture of recurrent neural network architecture where nodes and edges are represented as Long Short-Term Memory units (LSTM). They essentially rely on two independent RNNs: one for interactions evolutions and one for spatial inference. They reason on 3D skeletons and their interaction with objects. The pre-defined spatio-temporal graph changes with respect to the task. For example *drinking* only involve the human node and the cup node, whereas *making cereals* involves the human node and bowl, milk, cereals nodes. However they connect all the objects together in their spatio-temporal graph in order to handle the dynamic number of objects within the different sequences. They handle the variety of spatio-temporal graphs in a single structural RNN graph.

In an attempt to bring more generic properties, Qi *et al*. [23] propose to learn the graph structure instead of using pre-defined structures. The network learns to infer the adjacency matrix in an end-to-end manner. They rely on graph neural network (GNN). Through multiple iterations of mes-

sage passing and states updating of nodes, each node captures the semantic relation and structural information within its neighboring nodes.

The underlying model of the two previous works is a GNN. Recent studies proposed a variant to GNN: graph convolutional network (GCN). Convolutions are adapted from a 2D or 3D grid to a graph structure. This idea was initially applied to action recognition with the ST-GCN [29]. Nodes of the ST-GCN corresponds to human body joints, that are connected spatially and temporally. Spatial edges connect joints following the human skeleton structure, and temporal edges connect the same type of node across time. They solely use the skeleton information to infer actions. ST-GCN is computationally efficient compared to CNNs, especially regarding the memory usage, thank to the introduction of graph convolutions. However in this approach object and environment are not involved. Other recent approaches proposed the use of GCN over deep features [13, 3] to better capture temporal dependencies.

In an attempt to reduce 3D convolutional models parameters while maintaining performance, more recently volterra networks are proposed as in [24]. They introduced volterra filters based on the volterra series formulation [28]. They propose cascaded volterra filters to a significantly reduce model parameters and also for data fusion. They propose to a non linear fusion of the spatial and temporal streams.

Fusion strategies are often proposed to tackle numerous challenges. Fusion at the early stage may consists in the addition of the optical flow that describe the general motion to a sequence of images as in [5]. The fusion of different sources of information such as audio and video is also proposed as in [4]. These different approaches show the advantages of using a fusion mechanism to increase the overall performance. However, this gain is achieved at the expense of the amount of data required for training. The addition of more modalities increases the number of parameters in the convolutional network. This has two effects: first it requires the existence of a such dataset and second it increases the training time.

Here we design two independent and complementary GCNs: one based on spatio-temporal skeleton modeling and another one based on spatio-temporal objects modeling. We propose to merge both models at their prediction levels towards the same layer: a fully-connected layer. We study the late fusion of both models that *a priori* complement each other and the gains that this can yield.

3. Proposed Approach

This section briefly recalls the original spatio-temporal graph convolution network applied on skeleton sequence as in [29]. Then we describe our extension to objects interactions sequences. We also propose a late fusion strategy of the predictions from both graphs models. At last a Bayesian approach BM is also presented, we believe it may also show complementary to the graphs.

Graph convolution networks introduced by Kipf and Welling [10] apply convolutions to graphs instead of an image. The image is composed of pixels arranged in a grid, and the convolution operates on the ordered neighboring pixels. Unlike convolutions on images, for a graph node, the number of neighbors varies and are un-ordered. Following this adaptation of CNN to GCN a spatio-temporal graph convolution network for action recognition was introduced by Yan *et al*. [29]. They solely model the human skeleton as a spatio-temporal graph to perform action recognition motivated by the fact that some body parts are more relevant than others. The skeleton graph is defined by a set of nodes and edges $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$. $\mathcal{V}_s$ includes all the joints in a skeleton sequence. Each node is associated to a feature vector that contains the 2D positions of the joint and a confidence score. Two nodes are connected following natural connections of the human skeleton

We are interested in action recognition when a person is interacting with objects in its vicinity. Our work is based on Yan *et al*. [29] publication with their original spatio-temporal skeleton graph. It is able to model actions where the human pose is discriminant. However some actions require more contextual data. Thus we propose to build a second original graph convolutional network that aims to recognize actions from the object set behavior in the sequence. In our objects-graph $\mathcal{G}_o = (\mathcal{V}, \mathcal{E})$ the node set $\mathcal{V}$ contains all the object nodes in the sequence. Each node is associated to a feature vector. Similarly to the skeleton sequence the object sequence is represented by the sequence of 2D coordinates associated to the bounding box during the action. Thus, in the feature vector we add the coordinates of the center of the bounding box, a classification confidence score and a object label representing the object class. There are two types of edges $\{\mathcal{E}_S, \mathcal{E}_T\} \in \mathcal{E}$ and two nodes are connected by an edge $e \in \mathcal{E}_S$ if they can interact. Same objects are connected by a edge $e_t \in \mathcal{E}_T$ between two frames. It enables to add structural knowledge though the $e \in \mathcal{E}_S$ edges from our experience when designing this graph. For instance, in most cases, a knife does not interact with a bottle, whereas a glass may interact with a bottle when pouring water. Even though various objects appear in a given dataset, we define a single object-graph containing all the objects and their possible interactions.

Both spatio-temporal graphs are able to model temporal coherence within the action execution but it lacks from temporal coherence between two successive actions. For this reason we propose to model the correlations between two successive action states by adding a recurrent layer: a Gated Recurrent Unit (GRU).

Designed this way, both graph convolution models are lightweight in terms of number of nodes as well as in the

size of their associated features. Thus training time are consequently reduced.

It comes appropriate to use hyper-parameter optimization tools to tune the training parameters. Given a metric to optimize and an algorithm to run, such tools intend to find optimal parameters. We propose to use SMAC [6] to optimize the following free parameters: learning rate initialization, dropout and batch-size as shown in Tab 1. We choose to optimize with respect to the F1-score to overcome issues related to datasets eventually imbalanced. SMAC aims to build a model of the objective function in order to test at each iteration a promising set of parameters. SMAC is designed, through this model building to optimize costly evaluation function.

Out of the two graph models, we have one model specialized to predict actions where the body motion is highly characteristic and another model to better leverage ambiguities using the objects motion in the scene. We propose a fusion of their respective predictions. Both approaches estimate probability distributions for each class. We have two vectors corresponding to the soft-max layers: one for skeleton-graph model and one for the objects-graph model. We propose a strategy that takes as input video clips that are first pre-processed to extract skeleton and objects detections when not available in the ground truth.

We thus obtain two prediction vectors for each of the models that are later concatenated. This concatenation is connected to a dense layer of the same size as the number of classes, as shown with only $N = 4$ classes as example in Fig. 1. So there are only $N^2 + N$ parameters to learn (N^2 weights related to the dense layer and N bias related to activation). This interconnection aims to take the advantage of both models in the final prediction. Hereafter, this approach is named SKLO.

3.1. Bayesian approach with human-object observations

In this work we propose also propose to combine the graph models to a bayesian model. The bayesian model BM proposed by Maurice *et al*. [20] is based on the same observations: human pose, human-object and object-to-object interactions. Moreover transitions between actions, performed during the execution of an activity, provide spatio-temporal information that allows the recognition of the ongoing action. This approach relies on observations in order to estimate, at each time of the video, the probabilities of each considered actions.

All the elements of the scene are first localized in the image plane by 2D state-of-the-art detectors one for human pose estimation an another for the objects. Then they are modeled in 3D space using RGB-D sensor (*e.g*. Kinect) calibration data. The detection of the human pose in the image is based on OpenPose [1], which is trained on MSCOCO Keypoints Challenge [15]. Single Shot Multi-Box Detector (SSD) [17] is used to recognize objects, which is trained with the MSCOCO dataset [15].

They associated a model to each action a. Let $\mathrm{A} = \{a^1, ..., a^N\}$ be the set of N actions. The joint observation of the human pose s_t and the set of objects Ω_t is described at time t by $O_t = \{s_t, \Omega_t\}$ where $\Omega_t = \{\omega^1, ..., \omega^{Card(\Omega)}\}$ with $Card(\Omega)$ being the number of objects in the scene. The inference is performed on a sliding window of T frames, so that this approach does not require video clips segmentation beforehand, and ensure temporal consistency of the observations. They model the *a posteriori* probability of the actions given the observations as follows:

$$p(a_{0:T}|O_{0:T}) \propto \prod_{t=0}^{T} p(O_t|a_t) \prod_{t=1}^{T} p(a_t|a_{t-1}). \qquad (1)$$

Where $p(O_t|a_t)$ is the likelihood of the observation given the action a_t. $p(a_t|a_{t-1})$ characterizes the probabilities of transitions between two successive actions. All the observations of the scene in this approach are modeled in 3D thanks to the sensor calibration data.

In the end with BM, we infer probabilities for each action that we can merge to the predictions of SKLO. Similarly we propose to merge them through a fully-connected layer. Weights are learned to favor one model over the others depending on the predictions for each of the actions.

4. Implementation and datasets

This section briefly details the implementation and datasets selected for further evaluation.

4.1. Implementation details

In order to build our graphs we need the nature and the 2D positions of the human joints as well as objects present in the videos. Human skeleton positions are inferred using OpenPose [1]. Objects detections are either provided as ground truth by datasets as it is the case for CAD-120 either detected by SSD [18] pre-trained on MSCOCO [16].

Videos from CAD-120 and Watch-n-Patch are trimmed into video clips. Each video clip represents one action. During training and testing, we set our maximum temporal dimension to be 16. When the length of a video clip is less than 16, we zero-pad the missing positions. If the length of a video clip is greater than 16, we sub-sample the video clip.

Networks are trained using the standard loss for multi-class classification task: the categorical cross entropy loss. Training and evaluations are carried out on a NVIDIA 1080Ti graphic card.

Table 1: Hyper-parameters intervals

Hyper-parameter	Interval
Learning rate initialization	[0.01, 0.1]
Dropout	[0.5, 1]
Batch-size	[1, 160]

Figure 2: Example images from Watch-n-Patch, office environment on the left, kitchen on the right. In blue: upper-body joints detected by OpenPose. In yellow: objects detected by SSD.

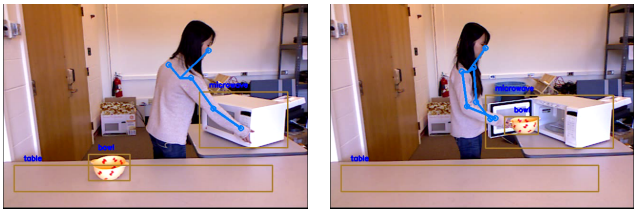

Figure 3: Example images from CAD-120 [11] dataset: actor 1, video #2305260828, activity *microwaving-food*. Action label on the left: reach, on the right: move.

4.2. Datasets

Our proposed approach combines skeleton and objects cues in two light spatio-temporal convolutional graph networks. To evaluate this proposition we choose datasets with numerous human objects interactions as well as some interactions between objects. The advantage of action sequences is that they allow to observe the positions of the same objects present in the scene during the execution of different actions.

CAD-120 - The CAD-120 [11] dataset consists of 120 videos with RGB-D channels, played by 4 actors. It contains 10 daily life activities (preparing a bowl of cereal, taking medication...). These activities involve 10 actions: reaching, moving, pouring, eating, drinking, placing, cleaning, opening, closing, null. Since its release it has been cited over 590 times and more than 100 times within the past year, approaches [22, 23] also evaluate themselves on this dataset. It features rich objects interactions.

Table 2: Detail of class distribution within datasets and the number of clips (Nc).

Dataset	Nc	Distribution (% per class)
CAD-120	1149	[23,30,3,3,3,15,4,3,1,14]
Watch-n-Patch ofc.	1148	[12,16,21,6,4,14,9,9,5,3]
Watch-n-Patch ktc.	1207	[15,11,8,8,6,10,6,17,6,6,6]

Watch-n-Patch - This dataset offers two environments with different actions and their associated training and testing sets. The office environment consists of 196 videos recorded in 8 different offices. There are 10 annotated actions: read, walk, leave-office, fetch-book, put-back-book, put-down-item, pick-up-item, play-computer, turn-on computer, turn-off computer. The kitchen environment consists of 117 videos recorded in 5 different kitchens. There are 11 annotated actions: fetch-from-fridge, put-back-to-fridge, prepare-food, microwaving, fetch-from-oven, pouring, drinking, leave-kitchen, fill-kettle, plug-in-kettle, move-kettle.

Both datasets are illustrated in Figs. 2 and 3.

4.3. Evaluation Metrics

Various evaluation metrics exist for the multi-class classification task. We evaluate our graph models and their fusion proposed in section 3 with two metrics. The first one is the usual accuracy, which is defined as follows:

$$\text{accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}. \quad (2)$$

This measures the ratio of correctly recognized actions to the total number of actions to recognize, independently of the number of samples in each class. We also compare our model to the literature according to the F1-score averaged over all classes. F1-score is a metric that takes into account the precision and recall of the different action classes. Those two metrics are interesting to study especially in presence of imbalanced datasets.

5. Evaluations

We first present the results obtained from each individual graph and benefits to be derived from their late fusion. In Tab. 3, we notice that the skeleton-graph model (SKL) outperforms the object-graph (O) model when comparing the raw accuracy on all datasets. Indeed, an advantage of the object-graph model is its ability to discriminate particular object configurations with respect to the actions to be detected. However if particular actions and associated object configurations offer a great specificity they are also less frequent in the dataset, therefore this performance counts less

Table 3: Accuracy of graphs models (SKL, O) and a bayesian model BM [20] and of their fusion. Gain in percentage points (pp).

Dataset	**CAD-120**					**WnP**
Fold	S1	S2	S3	S4	Mean	F0
SKL	65	74	63	61	66	83
O	43	32	44	34	38	27
SKLO	**83**	**91**	**86**	**78**	**85**	**85**
Gain vs. SKL	+18	+17	+23	+17	+19	+2
BM [20]	84	78	82	82	82	78
BM + C3D [19]	86	89	84	83	86	93
BM + SKLO	**89**	**91**	**87**	**85**	**88**	**90**
Gain vs. C3D	+6	0	+1	+7	+4	+5
Gain vs. SKLO	+6	0	+1	+7	+4	+5

Table 4: Performance comparison to the literature based on the F1-score using the CAD-120 dataset, in the two evaluation protocols.

Dataset	**Protocol**	**Approach**	**F1-score**
CAD-120	cross-subject	C3D	63.2
		LHAOD [11]	80.4
		S-RNN [8]	83.2
		BM [20] + SKLO	**87.5**
	cross-random	GPNN [23]	88.9
		BM [20] + SKLO	**89.4**
WnP	authors'split	KMHIS [9]	59.8
		C3D	71.1
		BM [20]	76.1
		GEPHAPP [22]	81.5
		BM + SKLO	**90.0**

Table 5: Memory and speed comparison.

Approach	**Number of parameters**	**Speed**
C3D [27]	15,929,225	17
Volterra networks [24]	4,600,000	-
GPNN [23]	24,356,171	11
SKLO	**1,581,669**	**80**

in the global accuracy computation. For example, the action *pouring* in the CAD-120 dataset implies the milk to be nearby the bowl and this action represents only 3 % of the actions.

We now study the gain derived from the fusion of (S) and (O) called SKLO, if the two models have a high specificity towards different action classes we observe a better overall accuracy. From their fusion, we obtain gains of +2 pp on Watch-n-Patch office and +19 pp on CAD-120. In Tab. 3, we also show that the fusion of SKLO and BM can further improve the performance in terms of accuracy with +4pp on CAD-120 and +5pp on Watch-n-Patch.

We present two evaluation protocols for CAD-120, the first (S1,...,S4) follows the instruction of the authors where the 4-fold cross-validation is build in order to train over 3 actors and test over the remaining 1 actor. In order to compare ourselves with a recent approach GPNN [23], we also follow their experimentation protocol, which differs from the original release of the dataset. In this experiment folds are built randomly with a 80 % in the train set and 20 % in the test set. They show a F1-score of 88.9 for action recognition, whereas we obtain in average 88.5 as shown in Tab. 4.

As we can see in Tab. 5, our approach requires 15 times less model parameters than GPNN and 10 times less than C3D. Even compared to recent networks aiming to reduce the number of parameters as in [24] our model uses 3 times less parameters. Our number of parameters of our model depends on the number of graph nodes, the number of frame chosen to sub-sample the video, the neighborhood extent during the convolution operation. Defining the prediction speed as the number of video clips processed per second, we are able to process 7 times more video clips per second compared to GPNN and almost 5 times compared to C3D.

On Watch-n-Patch kitchen, we observe that the skeleton-graph model (S) detects with an accuracy of 0.98 and 0.87 the actions *leave-kitchen* and *fetch-from-oven* respectively. Whereas it struggles regarding the actions *pouring* and *move kettle*. The later is often mistaken for *leave-kitchen* or *fetch-from-oven*. When we use the object graph, it is easier to clear-out the ambiguity as in *leave-kitchen* all the objects are static. The performance and sources of confusion are different and then the fusion (S+O) is able to benefit of both.

For a in-depth analysis, we provide in Figs. 4, 5 and 6 the confusion matrices obtained from the different graph models and from their fusion on CAD-120. On the diagonals we can clearly see that the performances of the models complement each other with respect to classes. Also, we can note that the source of confusion differs.

Weights of the dense layer can favor one model over another but it can also encompass the inter-correlations between the actions predictions. Weights learn to capture the specificity of each of the model. Actions that can't be distinguished by the observation of the objects, weights put emphasis on the prediction from the skeleton model. Meanwhile for some actions the emphasis is put on the prediction from the object model. Similar behavior is observed in the fusion of BM with SKLO, each approach has their own specificity.

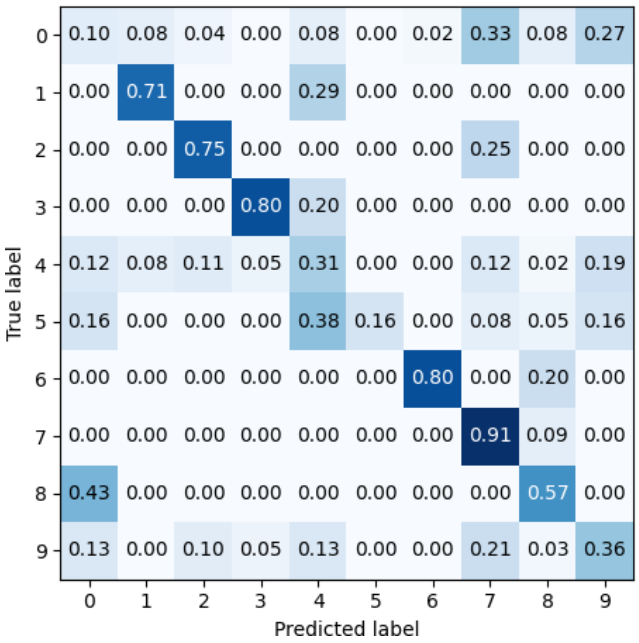

Figure 4: Confusion matrix of the Object Graph on CAD-120 dataset

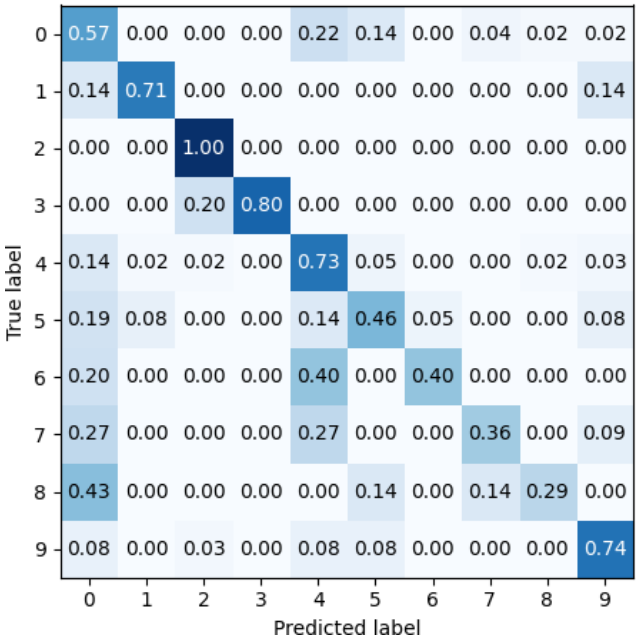

Figure 5: Confusion matrix of the Skeleton Graph on CAD-120 dataset

6. Conclusion and future work

We propose a modular and light-weight convolutional graph networks for action recognition. The prediction of the two networks are blended through a dense layer, this enables to learn weights that can encompass the interconnections that exists between the skeleton trajectory and the evolution of the objects configuration in the scene during any action.

We show that both networks achieve a great specificity towards some actions and they both complement each other. Thanks to this synergy the dense layer is able to clear out ambiguities. We show on CAD-120 and Watch-n-Patch that the addition of a object graph network and the fusion of the graph networks to BM approach yields performance gains in accuracy of respectively +21pp and +7pp over a baseline approach. We also exhibit a great reduction of the number of parameters, between 10 and 15 times less, compared to usual models of the literature.

Future work includes the use of other contextual information such as the room nature or the use of synthetically generated human pose and trajectories.

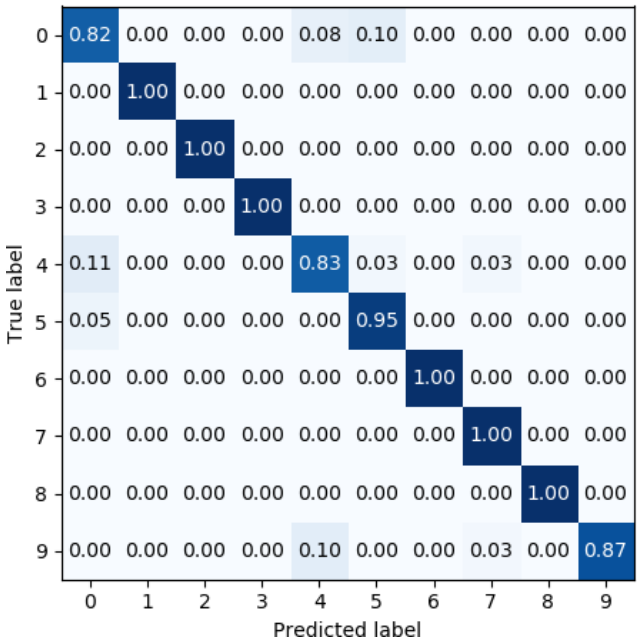

Figure 6: Confusion matrix of the fusion SKLO by a Fully Connected layer on CAD-120 dataset

7. Acknowledgement

This work has been partially supported by Bpifrance within the French Project LinTO and funded by the French government under the Investments for the Future Program (PIA3)

References

[1] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, pages 7291–7299, 2017. 4

[2] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 2

[3] Y. Chen, B. Guo, Y. Shen, W. Wang, W. Lu, and X. Suo. Boundary graph convolutional network for temporal action detection. *Image and Vision Computing*, 109:104144, 2021. 3

[4] Y. Fan, X. Lu, D. Li, and Y. Liu. Video-based emotion recognition using cnn-rnn and c3d hybrid networks. In *ACM International Conference on Multimodal Interaction*, pages 445–450, 2016. 3

[5] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, pages 1933–1941, 2016. 3

[6] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *ICLIO*, pages 507–523. Springer, 2011. 4

[7] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *PAMI*, 36(7):1325–1339, 2013. 2

[8] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *CVPR*, pages 5308–5317, 2016. 2, 6

[9] H. Kataoka, Y. Miyashita, M. Hayashi, K. Iwata, and Y. Satoh. Recognition of transitional action for short-term action prediction using discriminative temporal cnn feature. In *BMVC*, 2016. 6

[10] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *In ICLR*, 2017. 2, 3

[11] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013. 5, 6

[12] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pages 2556–2563. IEEE, 2011. 2

[13] J. Li, X. Liu, Z. Zong, W. Zhao, M. Zhang, and J. Song. Graph attention based proposal 3d convnets for action detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4626–4633, 2020. 3

[14] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pages 9–14. IEEE, 2010. 2

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 4

[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 4

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016. 4

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016. 4

[19] C. Maurice, F. Madrigal, and F. Lerasle. Late fusion of bayesian and convolutional models for action recognition. In *International Conference on Pattern Recognition (ICPR)*, pages 3296–3303. IEEE, 2021. 2, 6

[20] C. Maurice, F. Madrigal, A. Monin, and F. Lerasle. A new bayesian modeling for 3d human-object action recognition. In *AVSS*, pages 1–8. IEEE, 2019. 4, 6

[21] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Berkeley mhad: A comprehensive multimodal human action database. In *2013 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 53–60. IEEE, 2013. 2

[22] S. Qi, B. Jia, S. Huang, P. Wei, and S.-C. Zhu. A generalized earley parser for human activity parsing and prediction. *IEEE Transactions on PAMI*, 2020. 5, 6

[23] S. Qi, W. Wang, B. Jia, J. Shen, and S.-C. Zhu. Learning human-object interactions by graph parsing neural networks. In *ECCV*, pages 401–417, 2018. 2, 5, 6

[24] S. Roheda and H. Krim. Conquering the cnn over-parameterization dilemma: A volterra filtering approach for action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11948–11956, 2020. 3, 6

[25] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *CVPR*, pages 1010–1019, 2016. 2

[26] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2

[27] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015. 1, 2, 6

[28] V. Volterra. Theory of functionals and of integral and integro-differential equations. 1959. 3

[29] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018. 2, 3