Glimpse Clouds: Human Activity Recognition from Unstructured Feature Points

Fabien Baradel[1] , Christian Wolf[1,2], Julien Mille[3], Graham W. Taylor[4,5]
{fabien.baradel,christian.wolf}@liris.cnrs.fr, julien.mille@insa-cvl.fr, gwtaylor@uoguelph.ca

Abstract

We propose a method for human activity recognition from RGB data that does not rely on any pose information during test time, and does not explicitly calculate pose information internally. Instead, a visual attention module learns to predict glimpse sequences in each frame. These glimpses correspond to interest points in the scene that are relevant to the classified activities. No spatial coherence is forced on the glimpse locations, which gives the attention module liberty to explore different points at each frame and better optimize the process of scrutinizing visual information.

Tracking and sequentially integrating this kind of unstructured data is a challenge, which we address by separating the set of glimpses from a set of recurrent tracking/recognition workers. These workers receive glimpses, jointly performing subsequent motion tracking and activity prediction. The glimpses are soft-assigned to the workers, optimizing coherence of the assignments in space, time and feature space using an external memory module. No hard decisions are taken, i.e. each glimpse point is assigned to all existing workers, albeit with different importance. Our methods outperform the state-of-the-art on the largest human activity recognition dataset available to-date, NTU RGB+D, and on the Northwestern-UCLA Multiview Action 3D Dataset.

1. Introduction

We address human activity recognition in settings where activities are complex and diverse, either as performed by an individual, or involving multiple participants. These activities may even include people interacting with objects or the environment. The usage of RGB-D cameras is very popular for this case [44, 34, 56], as it allows for the use of articulated pose (skeletons) to be delivered in real time and relatively cheaply by some middleware. The exclusive usage of pose makes it possible to work on gesture and activity recognition without being a specialist in vision [61, 52], and with significantly reduced dimensionality of the input data. The combined usage of pose and raw depth and/or RGB images can often boost performance over a solution that uses a single modality [41].

We propose a method that only uses raw RGB images at test time. We avoid the use of articulated pose for two reasons: (i) depth data is not always available; for example, in applications involving smaller or otherwise resource-constrained robots; and (ii) the question of whether articulated pose is the optimal intermediate representation for activity recognition is unclear. We explore an alternative strategy, which consists of learning a local representation of video through a visual attention process.

We conjecture that the replacement of the articulated pose modality should keep one important property, which is its collection of local entities, which can be tracked over time and whose motion is relevant to the activity at hand. Instead of fixing the semantic meaning of these entities to the definition of a subset of joints in the human body, we learn it discriminatively. In our strategy, the attention process is completely free to attend to arbitrary locations at each time instant. In particular, we do not impose any constraints on spatio-temporal coherence of glimpse locations, which allows the model to vary its focus within and across frames. Certain similarities can be made to human gaze patterns which saccade to different points in a scene.

Activities are highly correlated with motion [47], and therefore tracking the motion of specific points of visual interest is essential, yielding a distributed representation of the collection of glimpses. Appearance and motion features need to be collected over time from local points and integrated into a sequential decision model. However, tracking a set of glimpse points, whose location is not spatio-temporally smooth and whose semantic meaning can change from frame to frame, is a challenge. Our objective is to match new glimpses with past ones of the same (or a nearby) location in the scene. Due to the unconstrained nature of the attention mechanism, it is not aware of when a point in the scene has been last scrutinized, or if it has been attended to in the past.

[1] Univ. Lyon, INSA-Lyon, CNRS, LIRIS, F-69621, Villeurbanne, France.

[2] INRIA, CITI Laboratory, Villeurbanne, France.

[3] Laboratoire d'Informatique de l'Univ. de Tours, INSA Centre Val de Loire, 41034 Blois, France.

[4] School of Engineering, Univ. of Guelph, Guelph, Ontario, Canada.

[5] Vector Institute, Toronto, Ontario, Canada.

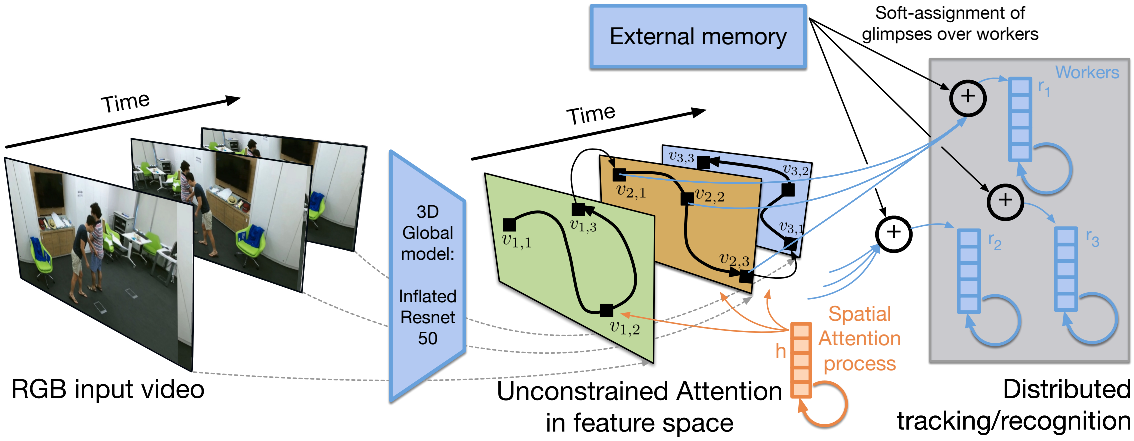
Time
Time
External memory
Soft-assignment of glimpses over workers
Workers
r_1
r_2
r_3
3D Global model: Inflated Resnet 50
$v_{1,1}$
$v_{1,2}$
$v_{1,3}$
$v_{2,1}$
$v_{2,2}$
$v_{2,3}$
$v_{3,1}$
$v_{3,2}$
$v_{3,3}$
h
Spatial Attention process
RGB input video
Unconstrained Attention in feature space
Distributed tracking/recognition

Figure 1. We recognize human activities from unstructured collections of spatio-temporal glimpses with distributed recurrent tracking/recognition and soft-assignment among glimpse points and trackers.

We solve this issue by separating the problem into two distinct parts: (i) selecting a distributed and local representation of G glimpse points through a sequential recurrent attention model; and (ii) tracking the set of glimpses by a set of C recurrent workers, which sequentially integrate features and participate in the final recognition of the activity (Figure 1). In general, G can be different from C, and the assignment between glimpses and workers is *soft*. Each worker is potentially assigned to all glimpses, albeit to a varying degree.

We summarize our main contributions as follows:

- We present a method for human activity recognition that does not require articulated pose during testing, and models activities using two attentional processes; one extracting a set of glimpses per frame and one reasoning about entities over time. This model has a number of interesting and general properties:
  - This unstructured "cloud" of glimpses produced by the attention process are tracked over time using a set of trackers/recognizers, which are soft-assigned using external memory. Each tracker can potentially track multiple glimpses.
  - Articulated pose is used during *training* time as an additional target, encouraging the attention process to focus on human structures.
  - All attentional mechanisms are executed in feature space, which is calculated jointly with a global model processing the full input image.
- We evaluate our method on two datasets, NTU RGB-D and N-UCLA Multiview Action 3D, outperforming the state-of-the-art by a large margin.

2. Related Work

Activities, gestures and multimodal data — Recent gesture and human activity recognition methods dealing with several modalities typically process 2D+T RGB and/or depth data as 3D. Sequences of frames are stacked into volumes and fed into convolutional layers at the first stages [3, 22, 40, 41, 57]. When additional pose data is available [37], the 3D joint positions are typically fed into a separate network. Preprocessing pose is reported to improve performance in some situations, e.g. augmenting coordinates with velocities and acceleration [62]. Fusing modalities is traditionally done as late [40], or early fusion [57]. In contrast, our method does not require pose during testing and only leverages it during training for regularization.

Recurrent architectures for action recognition — Recurrent neural networks (and their variants) are employed in much contemporary work on activity recognition, and a recent trend is to make recurrent models local. Part-aware LSTMs [44] separate the memory cell of an LSTM network [18] into part-based sub-cells and let the network learn long-term representations individually for each part, fusing the parts for output. Similarly, Du *et al.* [11] use bidirectional LSTM layers that fit an anatomical hierarchy. Skeletons are split into anatomically-relevant parts (legs, arms, torso, etc.) and let subnetworks specialize on them. Lattice LSTMs partition the latent space over a grid that is aligned with the spatial input space [50]. On the other hand, we soft-assign parts over multiple recurrent workers, each worker potentially integrating all points of the scene.

Tracking and distributed recognition — Structural RNNs [21] bear a certain resemblance to our work. They handle the temporal evolution of tracked objects in videos with a set of RNNs, each of which correspond to cliques in a graph that models the spatio-temporal relationships between these objects. However, this graph is hand-crafted for each application, and object tracking is performed using external trackers, which are not integrated into the neural model. Our model does not rely on external trackers and does not require the manual creation of a graph, as the assignments between objects (glimpses) and trackers are learned automatically.

Attention mechanisms and external memory — Attention mechanisms focus selectively on parts of the scene that are the most relevant to the task. Two types of attention have emerged in recent years. *Soft attention* weights each part of the observation dynamically [4, 25]. The objective function is usually differentiable, allowing gradient-based optimization. Soft attention was proposed for image [9, 58] and video understanding [46, 48, 59] with spatial, temporal, and spatio-temporal variants.

Towards action recognition, Sharma *et al.* [46] proposed a recurrent mechanism from RGB data, which integrates convolutional features from different parts of a space-time volume. Song *et al.* [48] proposed separate spatial and temporal attention networks for action recognition from pose. At each frame, the spatial attention model gives more importance to the joints most relevant to the current action, whereas the temporal model selects frames.

Hard attention takes explicit decisions when choosing parts of the input data. In a seminal paper, Mnih *et al.* [39] proposed visual hard attention for image classification built around an RNN, selecting the next location on based on past information. Similar hard attention was used in multiple object recognition [2], object localization [7, 38, 23], saliency map generation [27], and action detection [60]. While the early hard attention models were not differentiable, implying reinforcement learning, the DRAW algorithm [15] and spatial transformer networks (STN) [20] provide attention crops which are fully differentiable and can thus be learned using gradient-based optimization.

The addition of external memory proved to increase the capacity of neural networks by storing long-term information from past observations; this was mainly popularized by Neural Turing Machines and [14] and Memory Networks [49, 28]. In [1], a Fully Convolutional Network is coupled with an attention-based memory module to perform context selection and refinement for semantic segmentation. In [53], visual memory is used to learn a spatio-temporal representation of moving objects in a scene. Memory is implemented as a convolutional GRU with a 2D spatial hidden state. In [35], the ST-LSTM method of [34] is extended with a global context memory for skeleton-based action recognition. Multiple attention iterations are performed to optimize the global context memory, which is used for the final classification. In [51], an LSTM-based memory network is used for RGB and optical flow-based action recognition.

Our attention process is different from previously published work in that it produces an unstructured Glimpse Cloud in a spatio-temporal cube. The attention process is unconstrained, which we show to be an important design choice. In our work, the external memory module provides a way to remember past soft-assignments of glimpses in the recurrent workers. Furthermore, accessing the external memory is fully-differentiable, which allows for supervised end-to-end training.

3. Glimpse Clouds

We first introduce the following notation: We map an input video sequence $\boldsymbol{X} \in \mathbb{R}^{T \times H \times W \times 3}$ to a corresponding activity label y where H, W, T denote, respectively, the height, the width and the number of time steps. The sequence $\boldsymbol{X}$ is a set of RGB input images $\boldsymbol{X}_t \in \mathbb{R}^{H \times W \times 3}$ with $t{=}1...T$. We do not use any external information during testing, such as pose data, depth, or motion. However, if pose data is available during *training time*, our method is capable of integrating it through additional predictions and supervision, which we show increases the performance of the system (Section 4).

Many RGB-only state-of-the-art methods, which do not use pose data, extract features at a frame level by feeding the entire video frame to a pre-trained deep network. This yields global features, which do not capture local information well. Reasoning at a local level has, until now, been achieved using pose features, or attention processes that were limited to attention maps (e.g. [46, 33]). Here, we propose an alternative approach, where an attention process runs over each time instant ***and*** over time, creating sequences of sets of glimpse points, from which features are extracted.

Our model processes videos using several key components, as illustrated in Figure 1: i) a ***recurrent spatial attention model*** that extracts features from different local glimpses $v_{t,g}$ following an attention path in each video over frames t and multiple glimpses g in each frame; and ii) ***distributed soft-tracking workers***, which process these spatial features sequentially. As the input data is unstructured, the spatial glimpses are soft-assigned to the workers, such that no hard decisions are made at any point. To this end, iii) an ***external memory module*** keeps track of the glimpses seen in the past, their features, and past soft-assignments, and produces new soft-assignments optimizing spatio-temporal consistency. Our approach is fully-differentiable, allowing end-to-end training of the full model.

3.1. A joint global/local feature space

We recognize activities jointly based on global and local features. In order to speed up calculations and to avoid extracting redundant calculations, we use a single feature space computed by a global model. In particular, we map an input sequence $\boldsymbol{X}$ to a spatio-temporal feature map $\boldsymbol{Z} \in$

$\mathbb{R}^{T\times H'\times W'\times C'}$ using a deep neural network $f(\cdot)$ with 3D convolutions. Pooling is performed on the spatial dimensions, but not on time. This allows for the retention of the original temporal scale of the video, and therefore access to features in each frame. It should be noted, however, that due to the 3D convolutions, the temporal receptive field of a single "temporal" slice of the feature map is greater than a single frame. This is intended, as it allows the attention process to use motion. In an abuse of terminology, we will still use the term *frame* to specify the slice $\boldsymbol{Z}_t$ of a feature map with a temporal length of 1. More information on the architecture of $f(\cdot)$ is given in Section 5.

3.2. The attention process

Inspired by human behavior when scrutinizing a scene, we extract a fixed number of features from a series of G glimpses within each frame. The process of moving from one glimpse to another is achieved with a recurrent model. Glimpses are indexed by index $g{=}1\ldots\ G$, and each glimpse $\boldsymbol{Z}_{t,g}$ corresponds to a sub-region of $\boldsymbol{Z}_t$ using coordinates and scale $\boldsymbol{l}_{t,g}=\left[x_g, y_g, s^x_g, s^y_g\right]^\top_t$ output by a differentiable glimpse function, a Spatial Transformer Network (STN) [20]. STN allows the attention process to perform a differentiable crop operation on each feature map. Features are extracted using a *transformed ROI average pooling* at location $\boldsymbol{l}_{t,g}$, resulting in a 1D feature vector $\boldsymbol{z}_{t,g}$:

$$\boldsymbol{Z}_{t,g} = \text{STN}(\boldsymbol{Z}_t, \boldsymbol{l}_{t,g}) \tag{1}$$

$$\boldsymbol{z}_{t,g} = \Gamma(\boldsymbol{Z}_{t,g}) = \frac{1}{H'W'}\sum_m\sum_n \boldsymbol{Z}_{t,g}(m,n) \tag{2}$$

where $W'{\times}H'$ is the size of the glimpse region. The glimpse locations and scales $\boldsymbol{l}_{t,g}$ for $g{=}1\ldots\ G$ are predicted by a recurrent network, which runs over the glimpses. As illustrated in Figure 1, the model predicts a fixed-length sequence of glimpse points for each frame. It runs over the entire video at once, i.e. it is not restarted/reinitialized after each frame. The hidden state therefore carries information across frames and creates a globally coherent scrutinization process for the video. The recurrent model is given as follows (we use GRUs [10] for simplicity's sake, and we omit gates and biases in the rest of the equations to reduce notational complexity):

$$\boldsymbol{h}_g = \Omega(\boldsymbol{h}_{g-1}, [\boldsymbol{z}_{g-1}, \boldsymbol{r}]\,|\theta) \tag{3}$$

$$\boldsymbol{l}_g = W_l^\top\,[\boldsymbol{h}_g, \boldsymbol{c}] \tag{4}$$

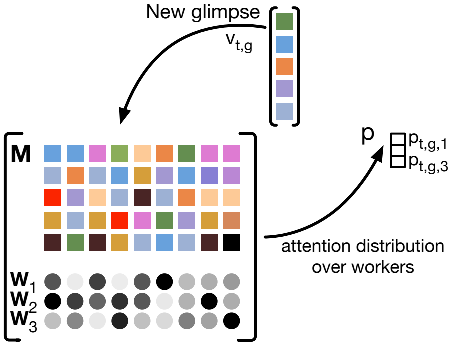

Figure 2. An external memory module determines an attention distribution over workers (a soft assignment) for each new glimpse $\boldsymbol{v}_{t,g}$ based on similarities with past glimpses M and their past attention probabilities w. Shown for a single glimpse and 3 workers.

where $\boldsymbol{h}$ denotes the hidden state of the RNN running over glimpses g, $\boldsymbol{c}$ is a frame context vector for making the process aware of frame transitions (described in Section 3.6), and $\boldsymbol{r}$ carries information about the high-level classification task. In essence, $\boldsymbol{r}$ corresponds to the global hidden state of the recurrent workers performing the actual recognition, as described in Section 3.3 and Equation (7). Note that $\boldsymbol{h}$, $\boldsymbol{z}$, $\boldsymbol{l}$, $\boldsymbol{r}$, and $\boldsymbol{c}$ depend on both the time and glimpse indices. Observing that the recurrence runs over glimpses g, the time index t is dropped from Eq. 3–4 for notational simplicity.

3.3. Distributed soft-tracking workers

Extracting motion cues from semantic points in a scene requires associating glimpse points from different frames over time. Due to the freedom of the attention process and fixed number of glimpses, subsequent glimpses of the same point in the scene are generally not in subsequent frames, which excludes the use of conventional tracking mechanisms. Instead, we avoid hard tracking and hard assignments between glimpse points in a temporal manner. We propose a soft associative model for automatically grouping similar spatial features over time.

As given in Equation (2), we denote $\boldsymbol{z}_{t,g}$ as the features extracted from the g^{th} glimpse in feature map $\boldsymbol{Z}_t$ for $g=1...G$ and $t=1...T$. We are interested in a joint encoding of spatial and feature dimensions and employ "*what*" and "*where*" features $\boldsymbol{v}_{t,g}$, as introduced in [29], defined by:

$$\boldsymbol{v}_{t,g} = \boldsymbol{z}_{t,g} \otimes \Lambda(\boldsymbol{l}_{t,g}|\theta_\Lambda) \tag{5}$$

where $\otimes$ is the Hadamard product and $\Lambda(\boldsymbol{l}_{t,g}|\theta_\Lambda)$ is a network providing an embedding of the spatial patch coordinates into a space of the same dimensionality as the features $\boldsymbol{z}_{t,g}$. The vector $\boldsymbol{v}_{t,g}$ contains joint cues about motion, appearance, and spatial localization.

Evolution of this information over time is modeled with a number C of so-called *soft-tracking workers* Ψ_c for $c{=}1...C$, each of which corresponds to a recurrent model capable of tracking entities over time. We *never* hard assign glimpses to workers. Inputs to each individual worker correspond to weighted contributions from all of the G glimpses. In general, the number of glimpse points G can be different from the number of workers C. At each instant, glimpses are thus soft-assigned to the workers on the fly by changing the weights of the contributions, as described further below.

Workers Ψ_c are GRUs following the usual update equations based on the past state $\boldsymbol{r}_{t-1,c}$ and input $\tilde{\boldsymbol{v}}_{t,c}$:

$$\boldsymbol{r}_{t,c} = \Psi_c(\boldsymbol{r}_{t-1,c}, \tilde{\boldsymbol{v}}_{t,c} | \theta_{\Psi_c}) \tag{6}$$

$$\boldsymbol{r}_t = \sum_c \boldsymbol{r}_{t,c} \tag{7}$$

where Ψ_c is a GRU and $\boldsymbol{r}_t$ carries global information about the current state (needed as input to the recurrent model of spatial attention). The input $\tilde{\boldsymbol{v}}_{t,c}$ to each worker Ψ_c is a linear combination of the different glimpses $\{\boldsymbol{v}_{t,g}\}, g = 1 \dots G$ weighted by a soft attention distribution $\boldsymbol{p}_{t,c} = \{p_{t,g,c}\},\ g = 1 \dots G$:

$$\tilde{\boldsymbol{v}}_{t,c} = \boldsymbol{V}_t \boldsymbol{p}_{t,c} \tag{8}$$

where $\boldsymbol{V}_t$ is a matrix whose rows are the different glimpse features $\boldsymbol{v}_{t,g}$. Workers are independent from each other in the sense that they do not share parameters θ_{Ψ_c}. This can potentially lead to specialization of the workers on types of tracked and integrated scene entities.

3.4. Soft-assignment using External Memory

The role of the attention distribution $\boldsymbol{p}_{t,c}$ is to give higher weights to glimpses that have been soft-assigned to worker c in the past; therefore, workers extract different kinds of features from each other. To accomplish this, we employ an external memory bank denoted $\boldsymbol{M} = \{\boldsymbol{m}_k\}$, which is common to all workers (see Figure 2). In particular, $\boldsymbol{M}$ is a fixed-length array of K entries $\boldsymbol{m}_k$, each capable of storing a feature vector $\boldsymbol{v}_{t,g}$. Even if the external memory is common to each worker, they have their own ability to extract information from it. Each worker Ψ_c has its own weight bank denoted $\boldsymbol{W}_c = \{w_{c,k}\}$. The scalar $w_{c,k}$ holds the importance of the entry $\boldsymbol{m}_{c,k}$ for worker Ψ_c . Hence, the overall external memory is defined by the set $\{\boldsymbol{M}, \boldsymbol{W}_1, \dots \boldsymbol{W}_C\}$.

Two operations can be performed on the external memory: *reading* and *writing*. *Memory reading* consists of extracting knowledge that is already stored in memory banks. Meanwhile, *memory writing* consists of adding a new memory entry to the memory bank. We describe these two fully-differentiable operations below.

Attention from memory reads — The attention distribution $\boldsymbol{p}_{t,c}$ is a distribution over glimpses g, i.e. $\boldsymbol{p}_{t,c} = \{p_{t,c,g}\}$, $0 \leq p_{t,c,g} \leq 1$ and $\sum_g p_{t,c,g}$=1. We want the glimpses to be distributed appropriately across the workers, and encourage worker specialization. In particular, at each timestep, we want to assign a glimpse of high importance to a worker if this worker has been soft-assigned similar glimpses in the past (also with high importance). To this end, we define a fully trainable distance function $\phi(.,.)$, which is implemented in a quadratic form:

$$\phi(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x}-\boldsymbol{y})^\top \boldsymbol{D} (\boldsymbol{x}-\boldsymbol{y})} \tag{9}$$

where $\boldsymbol{D}$ is a learned weight matrix. Within each batch, we normalize $\phi(\cdot,\cdot)$ with min-max normalization to scale it between 0 and 1.

A glimpse g is soft-assigned to a given worker c with a higher weight $p_{t,c,g}$ if $\boldsymbol{v}_{t,g}$ is similar to vectors $\boldsymbol{m}_k$ from the memory bank $\boldsymbol{M}$, which had a high importance for the worker in the past Ψ_c :

$$p_{t,c,g} = \sigma_\alpha \left(\sum_k e^{-t^{\boldsymbol{m}_k}} \times w_{c,k} \left[1 - \phi(\boldsymbol{v}_{t,g}, \boldsymbol{m}_k) \right] \right) \tag{10}$$

where σ is the softmax function over the G glimpses and $e^{-t^{\boldsymbol{m}_k}}$ is an exponential rate over time to give higher importance to recent feature vectors compared to those in the distant past. $t^{\boldsymbol{m}_k}$ is the corresponding timestep of the memory entry $\boldsymbol{m}_k$. In practice, we add a temperature term α to the softmax function σ. When $\alpha \to 0$, the output vector is sparser. The negative factor multiplied with ϕ is justified by the fact that ϕ is initially pre-trained as a Mahalanobis distance by setting $\boldsymbol{D}$ to the inverse covariance matrix of the glimpse data. The factor therefore transforms the distance into a similarity. After pre-training, $\boldsymbol{D}$ is trained end-to-end.

The attention distribution $\boldsymbol{p}_{t,c}$ is computed for each worker Ψ_c. Thus, each glimpse g potentially contributes to each worker Ψ_c through its input vector $\tilde{\boldsymbol{v}}_{t,c}$ (c.f. Equation (8)), albeit with different weights.

Memory writes — For each frame, the feature representations $\boldsymbol{v}_{t,g}$ are stored in the memory bank $\boldsymbol{M}$. However, the attention distribution $\boldsymbol{p}_{t,c} = \{p_{t,c,g}\}$ is used to weight these entries for each worker Ψ_c. If a glimpse feature $\boldsymbol{v}_{t,g}$ is stored in a slot $\boldsymbol{m}_k$, then its importance weight $w_{c,k}$ for worker Ψ_c is set to $p_{t,c,g}$. The only limitation is the size K of the memory bank. When the memory is full, we delete the oldest memory entry. More flexible storing processes, for example, trained mappings, are left for future work.

3.5. Recognition

Since workers proceed in a independent manner through time, we need an aggregation strategy to perform classification. Each worker Ψ_c has its own hidden state $\{\boldsymbol{r}_{t,c}\}_{t=1\dots T}$ and is responsible for its own classification through a fully-connected layer. The final classification is done by averaging logits of the workers:

$$\boldsymbol{q}_c = W_c \cdot \boldsymbol{r}_c \tag{11}$$

$$\hat{\boldsymbol{y}} = \text{softmax} \left(\sum_c^C \boldsymbol{q}_c \right) \tag{12}$$

where $\hat{\boldsymbol{y}}$ is the probability vector of assigning the input video $\boldsymbol{X}$ to each class.

3.6. Context vector

In order to make the spatial attention process (Section 3.2) aware of frame transitions, we introduce a context vector $\boldsymbol{c}_t$ which contains high level information about humans present in the current frame t. $\boldsymbol{c}_t$ is obtained by global average pooling over the spatial domain of the penultimate feature maps of a given timestep. If pose is available at training time, we regress the 2D pose coordinates of humans from the context vector $\boldsymbol{c}_t$ using the following mapping:

$$\boldsymbol{y}_t^p = W_p^\top \boldsymbol{c}_t \tag{13}$$

Pose $\boldsymbol{y}_t^p$ is linked to ground truth pose (only during *training*) using a supervised term described in Section 4. This leads to hierarchical feature learning in that the penultimate feature maps have to detect human joints present in each frame.

4. Training

We train the model end-to-end with the sum of a collection of loss terms, which are explained below:

$$\mathcal{L} = \mathcal{L}_D(\hat{\boldsymbol{y}}, \boldsymbol{y}) + \mathcal{L}_P(\hat{\boldsymbol{y}}^p, \boldsymbol{y}^p) + \mathcal{L}_G(\boldsymbol{l}, \boldsymbol{y}^p) \tag{14}$$

Supervision — $\mathcal{L}_D(\hat{\boldsymbol{y}}, \boldsymbol{y})$ is a supervised loss term (cross-entropy loss on activity labels $\boldsymbol{y}$).

Pose prediction — Articulated pose $\boldsymbol{y}^p$ is available for many datasets. Our goal is to *not* depend on pose during testing; however, its usage during training can provide additional information to the learning process and reduce the tendency of activity recognition methods to memorize individual elements in the data for recognition. We therefore add an additional term $\mathcal{L}_P(\hat{\boldsymbol{y}}^p, \boldsymbol{y}^p)$, which encourages the model to perform pose regression during training only from intermediate feature maps (described in Section 3.6). Pose regression over time leads to a faster convergence of the overall model.

Making glimpses similar to humans — $\mathcal{L}_G(\boldsymbol{l}, \boldsymbol{y}^p)$ is a loss encouraging the glimpse points to be as sparse as possible within a frame, but at the same time, close to humans in the scene. Recall that $\boldsymbol{l}_{t,g} = \left[x_{t,g}, y_{t,g}, s^x_{t,g}, s^y_{t,g}\right]^T$, so $\mathcal{L}_G$ is defined by:

$$\mathcal{L}_{G_1}(\boldsymbol{l}_t) = \frac{1}{1 + \sum_{g_1}^{G}\sum_{g_2}^{G} ||\boldsymbol{l}_{t,g_1}, \boldsymbol{l}_{t,g_2}||} \tag{15}$$

$$\mathcal{L}_{G_2}(\boldsymbol{l}_t, \boldsymbol{y}_t^p) = \sum_{g}^{G} \min_j ||\boldsymbol{l}_{t,g}, \boldsymbol{y}_{t,j}^p|| \tag{16}$$

$$\mathcal{L}_G(\boldsymbol{l}, \boldsymbol{y}^p) = \sum_{t}^{T} \left(\mathcal{L}_{G_1}(\boldsymbol{l}_t) + \mathcal{L}_{G_2}(\boldsymbol{l}_t, \boldsymbol{y}_t^p)\right) \tag{17}$$

where $\boldsymbol{y}_{t,j}^p$ denotes the 2D coordinates of joints j at time t, and Euclidean distance on $\boldsymbol{l}_{t,g}$ is computed using the central focus point $(x_{t,g}, y_{t,g})$. $\mathcal{L}_{G_1}$ encourages diversity between glimpses within a frame. $\mathcal{L}_{G_2}$ ensures that all the glimpses are not taken too far away from the subjects.

5. Pre-trained architecture

We designed the 3D convolutional network $f(\cdot)$ by computing the global feature maps in Section 3.1, such that the temporal dimension is maintained (i.e. without any temporal subsampling). Using a pre-trained Resnet-50 network [17] as a starting point, we inflate the 2D spatial convolutional kernels into 3D kernels, artificially creating new a temporal dimension, as described by Carreira *et al.* [8]. This allows us to take advantage of the 2D kernels learned by pre-training on image classification on the Imagenet dataset. The inflated ResNet $f(\cdot)$ is then trained as a first step by minimizing the loss $\mathcal{L}_D + \mathcal{L}_P$. The supervised loss $\mathcal{L}_D$ on the global model is applied on a path attached to global average pooling on the last feature maps, followed by a fully-connected layer that is subsequently removed.

The recurrent spatial attention module $\Omega(\cdot)$ is a GRU with a hidden state of size 1024; $\Lambda(\cdot)$ is an MLP with a single hidden layer of size 256 and a ReLU activation; the soft-trackers Ψ_c are GRUs with a hidden state of size 512. There is no parameter sharing among them.

6. Experimental Results

We evaluated the proposed method on two human activity recognition datasets: NTU RDB+D Dataset [44] and Northwestern-UCLA Multiview Action 3D Dataset [55].

NTU RDB+D Dataset (NTU) — NTU was acquired with a Kinect v2 sensor and contains more than 56K videos and 4 million frames with 60 different activities including individual activities, interactions between multiple people, and health-related events. The activities were performed by 40 subjects and recorded from 80 viewpoints. We follow the cross-subject and cross-view split protocol from [44]. Due to the large number of videos, this dataset is highly suitable for deep learning modeling.

Northwestern-UCLA Multiview Action 3D Dataset (N-UCLA) — This dataset [55] contains 1494 sequences, covering ten action categories, such as *drop trash* or *sit down*. Each sequence is captured simultaneously by 3 Kinect v1 cameras. RGB, depth and human pose are available for each video, and each action is performed one to six times by ten different subjects. Most actions involve human-object interaction, making this dataset challenging. We followed the cross-view protocol defined by [55], and we trained our method on samples from two camera views, and tested it on samples from the remaining view. This produced three possible cross-view combina-

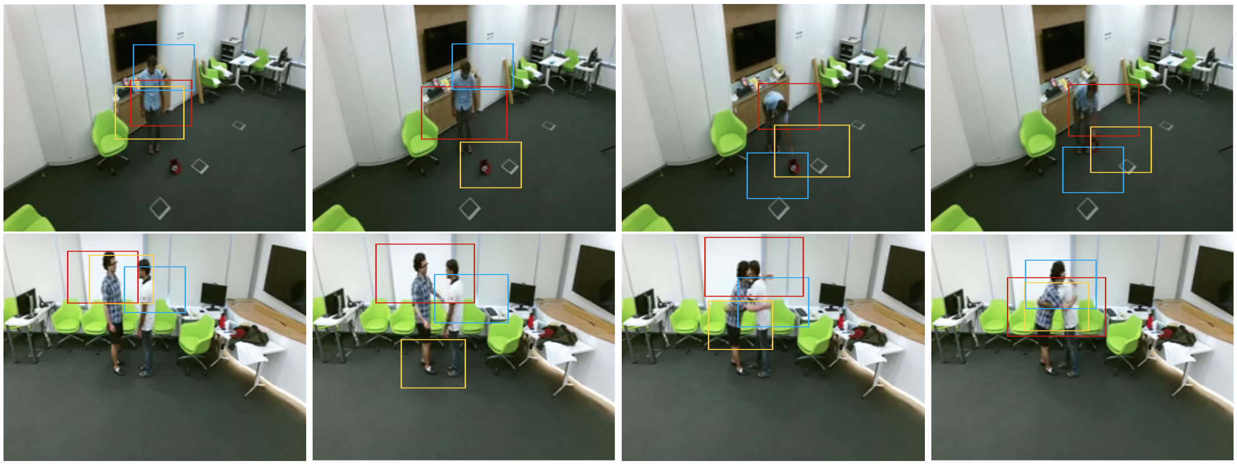

Figure 3. An illustration of the glimpse distribution for several sequences of the NTU dataset. Here we set 3 glimpses per frame (G=3, Red: first, Blue: second, Yellow: third).

Table 1. Results on the Northwestern-UCLA Multiview Action 3D dataset with Cross-View Setting (accuracy as a percent). V, D, and P mean Visual (RGB), Depth, and Pose, respectively.

Methods	Data	$V_{1,2}^3$	$V_{1,3}^2$	$V_{2,3}^1$	Avg
DVV [32]	D	58.5	55.2	39.3	51.0
CVP [64]	D	60.6	55.8	39.5	52.0
AOG [55]	D	45.2	-	-	-
HPM+TM [43]	D	**91.9**	75.2	71.9	79.7
Lie group [54]	P	74.2	-	-	-
HBRNN-L [12]	P	78.5	-	-	-
Enhanced viz. [36]	P	86.1	-	-	-
Ensemble TS-LSTM [30]	P	89.2	-	-	-
Hankelets [31]	V	45.2	-	-	-
nCTE [16]	V	68.6	68.3	52.1	63.0
NKTM [42]	V	75.8	73.3	59.1	69.4
Global model	V	85.6	84.7	79.2	83.2
Glimpse Clouds	V	90.1	**89.5**	**83.4**	**87.6**

tions: $V_{1,2}^3, V_{1,3}^2, V_{2,3}^1$. The combination $V_{1,2}^3$ means that samples from view 1 and 2 are used for training, and samples from view 3 are used for testing.

6.1. Implementation details

Similar to [44], we cut videos into sub-sequences of 8 frames and sample sub-sequences. During training, a single sub-sequence is sampled. During testing, 5 sub-sequences and logits are averaged. RGB videos are rescaled to 256 × 256 and random cropping of size 224 × 224 is done during training and testing.

Training is performed using the Adam Optimizer [26] with an initial learning rate of 0.0001. We use minibatches of size 40 on 4 GPUs. Following [44], we sample 5% of the initial training set as a validation set, which is used for hyper-parameter optimization and for early stopping. All hyperparameters have been optimized on the validation sets of the respective datasets. We used the model trained on NTU as a pre-trained model and fine-tuned it on N-UCLA.

Table 2. Results on the NTU RGB+D dataset with Cross-Subject and Cross-View settings (accuracies in %); († indicates method has been re-implemented).

Methods	Pose	RGB	CS	CV	Avg
Lie Group [54]	✓	-	50.1	52.8	51.5
Skeleton Quads [13]	✓	-	38.6	41.4	40.0
Dynamic Skeletons [19]	✓	-	60.2	65.2	62.7
HBRNN [11]	✓	-	59.1	64.0	61.6
Deep LSTM [44]	✓	-	60.7	67.3	64.0
Part-aware LSTM [44]	✓	-	62.9	70.3	66.6
ST-LSTM + TrustG. [34]	✓	-	69.2	77.7	73.5
STA-LSTM [48]	✓	-	73.2	81.2	77.2
Ensemble TS-LSTM [30]	✓	-	74.6	81.3	78.0
GCA-LSTM [35]	✓	-	74.4	82.8	78.6
JTM [56]	✓	-	76.3	81.1	78.7
MTLN [24]	✓	-	79.6	84.8	82.2
VA-LSTM [63]	✓	-	79.4	87.6	83.5
View-invariant [36]	✓	-	80.0	87.2	83.6
DSSCA - SSLM [45]	✓	✓	74.9	-	-
STA-Hands [5]	X	X	82.5	88.6	85.6
Hands Attention [6]	✓	✓	84.8	90.6	87.7
C3D†	-	✓	63.5	70.3	66.9
Resnet50+LSTM†	-	✓	71.3	80.2	75.8
Glimpse Clouds	-	✓	**86.6**	**93.2**	**89.9**

6.2. Results

Comparison with the state of the art — Our method outperforms state-of-the-art methods on NTU and N-UCLA by a large margin, and this also includes several methods which use multiple modalities, in addition to RGB, depth and pose. Tables 1 and 2 provide detailed results comparing to the state-of-the-art on the NTU dataset. Sample visual results can be seen in Figure 3.

Ablation study — Table 3 shows several experiments to study the effect of our design choices. Classification from the Global Model alone (Inflated-Resnet-50) is clearly inferior to the distributed recognition strategy using the set of workers (+1.9 points on NTU and +4.4 points on N-

Methods	Spatial Attention	Soft Workers	L_D	L_P	L_G	CS	CV	Avg
GM	-	-	✓	-	-	84.5	91.5	88.0
GM	-	-	✓	✓	-	85.5	92.1	88.8
GM+$\sum$ Glimpses + GRU	-	-	✓	✓	-	85.8	92.4	89.1
GC	✓	✓	✓	-	-	85.7	92.5	89.1
GC	✓	✓	✓	✓	-	86.4	93.0	89.7
GC	✓	✓	✓	-	✓	86.1	92.9	89.5
GC	✓	✓	✓	✓	✓	**86.6**	**93.2**	**89.9**
GC + GM	✓	✓	✓	✓	✓	86.6	93.2	89.9

Table 3. Results on NTU: ablation study. GM stands for Global Model and GC stands for Glimpse Clouds.

Glimpse	Type of attention	CS	CV	Avg
3D tubes	Attention	85.8	92.7	89.2
Seq. 2D	Random sampling	80.3	87.8	84.0
Seq. 2D	Saliency	86.2	92.9	89.5
Seq. 2D	**Attention**	**86.6**	**93.2**	**89.9**

Table 4. Results on the NTU: different attention and alternative strategies.

UCLA). The bigger gap obtained on N-UCLA can be explained by the larger portion of the frame occupied by people and therefore higher efficiency of a local representation. The additional loss predicting pose during training helps, even though pose is not used during testing. An important question is whether the Glimpse Cloud could be integrated with an easier mechanism than a soft-assignment. We tested a baseline which sums glimpse features for each time step and which integrates them temporally (row #3). This gave only a very small improvement over the global model. Distributed recognition from Glimpse Clouds with soft-assignment clearly outperforms the simpler baselines. Adding the global model does not gain any improvement.

Importance of losses — Table 3 also shows the relative importance of our three loss functions. Cross-entropy only L_D gives 89.1%. Adding pose prediction L_P we gain 0.6 points and adding pose attraction L_G we gain 0.4 points, which are complementary.

Unstructured vs. coherent attention — We also evaluated the choice of unstructured attention, i.e. the decision to give the attention process complete freedom to attend to a new (and possibly unrelated) set of scene points in each frame. We compared this with an alternative choice, where glimpses are axis-aligned space-time tubes over the whole temporal length of the video. In this baseline, the attention process is not aligned with time. At each iteration, a new tube is attended in the full space-time volume, and no tracking or soft-assignment to worker modules is necessary. As indicated in Table 4, this choice is sub-optimal. We conjecture that tubes cannot cope with moving objects and object parts in the video.

Attention vs. saliency vs. random — We evaluated whether a sequential attention process contributes to performance, or whether the gain is solely explained from the sampling of local features in the space-time volume. We compared our choice with two simple baselines: (i) complete random sampling of local features, which leads to a drop of more than 6 points, indicating that the location of the glimpses is clearly important; and (ii) with a saliency model, which predicts glimpse locations in parallel through different outputs of the location network. This is not a full attention process in that a glimpse prediction does not depend on what the model has seen in the past. This choice is also sub-optimal.

Learned weight matrix — Random initialization and fine-tuning of D matrix in Equation 9 loses 0.4 points and leads to slower convergence by a factor of 1.5. Fixing D (to inverse covariance) w/o any training loses 0.8 points.

The Joint encoding — "What and where" features are important for correctly weighting their respective contribution. Plainly adding concatenating coordinates and features loses 1.1 points.

Hyper-parameters C, G, T — Number of glimpses and workers: C and G were selected by cross-validation on the validation set by varying them from 1 to 4, giving an optimum of G=C=3 over all 16 combinations. More leads the model to overfit. The size of the memory bank K is set to T where T=8 is the length of the sequence.

Runtime — The model has been trained using data-parallelism over 4 Titan Xp GPUs. Pre-training the global model on the NTU dataset takes 16h. Training the Glimpse Cloud model end-to-end takes a further 12h. A single forward pass over the full model takes 97ms on 1 GPU. The method has been implemented in PyTorch.

7. Conclusion

We proposed a method for human activity recognition that does not rely on depth images or articulated pose, though it is able to leverage pose information available during training. The method achieves state-of-the-art performance on the NTU and N-UCLA datasets even when compared to methods that use pose, depth, or both at test time. An attention process over space and time produces an unstructured *Glimpse Cloud*, which is soft-assigned to a set of tracking/recognition workers. In our experiments, we showed that this distributed recognition outperforms a global convolutional model, as well as local models with simple baselines for the localization of glimpses.

Acknowledgements — This work was funded by grant Deepvision (ANR-15-CE23-0029, STPGP-479356-15), a joint French/Canadian call by ANR & NSERC.

References

[1] A. Abdulnabi, B. Shuai, S. Winkler, and G. Wang. Episodic camn: Contextual attention-based memory networks with iterative feedback for scene labeling. In *CVPR*, 2017. 3

[2] J. Ba, V. Mnih, and K. Kavukcuoglu. Multiple object recognition with visual attention. In *ICLR*, 2015. 3

[3] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In *HBU*, 2011. 2

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015. 3

[5] F. Baradel, C. Wolf, and J. Mille. Human action recognition: Pose-based attention draws focus to hands. In *ICCV Workshop*, 2017. 7

[6] F. Baradel, C. Wolf, and J. Mille. Pose-conditioned spatio-temporal attention for human action recognition. *Pre-print: arxiv:1703.10106*, 2017. 7

[7] M. Bellver, X. Giro-i Nieto, F. Marques, and J. Torres. Hierarchical object detection with deep reinforcement learning. In *Deep Reinforcement Learning Workshop, NIPS*, December 2016. 3

[8] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 6

[9] K. Cho, A. Courville, and Y. Bengio. Describing multimedia content using attention-based encoder-decoder networks. *IEEE-T-Multimedia*, 17:1875 – 1886, 2015. 3

[10] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-Decoder approaches. *arXiv preprint arXiv:1507.05738*, 2014. 4

[11] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, June 2015. 2, 7

[12] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 7

[13] G. Evangelidis, G. Singh, and R. Horaud. Skeletal quads:human action recognition using joint quadruples. In *ICPR*, pages 4513–4518, 2014. 7

[14] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. Oct. 2014. 3

[15] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra. DRAW: A recurrent neural network for image generation. In *ICML*, 2015. 3

[16] A. Gupta, J. Martinez, L. J, and W. R. 3d pose from motion for cross-view action recognition via non- linear circulant temporal encoding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 7

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 6

[18] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 2

[19] J. Hu, W.-S. Zheng, J.-H. Lai, and J. Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. In *CVPR*, pages 5344–5352, 2015. 7

[20] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *NIPS*, pages 2017–2025, 2015. 3, 4

[21] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-RNN: Deep Learning on Spatio-Temporal Graphs. In *CVPR*, 2016. 2

[22] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE TPAMI*, 35(1):221–231, 2013. 2

[23] Z. Jie, X. Liang, J. Feng, X. Jin, W. Lu, and S. Yan. Tree-structured reinforcement learning for sequential object localization. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 127–135. Curran Associates, Inc., 2016. 3

[24] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid. A new representation of skeleton sequences for 3d action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 7

[25] Y. Kim, C. Denton, L. Hoang, and A. Rush. Structured attention networks. In *ICLR*, 2017 (to appear). 3

[26] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICML*, 2015. 7

[27] J. Kuen, Z. Wang, and G. Wang. Recurrent attentional networks for saliency detection. In *CVPR*, pages 3668–3677, 2015. 3

[28] A. Kumar, O. Irsoy, J. Su, R. Bradbury, R. English, and B. Pierce. Ask me anything: Dynamic memory networks for natural language processing. In *ICML*, 2016. 3

[29] H. Larochelle and G. Hinton. Learning to combine foveal glimpses with a third-order Boltzmann machine. In *NIPS*, pages 1243–1251, 2010. 4

[30] I. Lee, D. Kim, S. Kang, and S. Lee. Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In *ICCV*, Oct 2017. 7

[31] B. Li, O. Camps, and M. Sznaier. Cross-view activity recognition using hankelets. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 7

[32] R. Li and T. Zickler. Discriminative virtual views for cross-view action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 7

[33] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. Snoek. VideoLSTM convolves, attends and flows for action recognition. *CVIU*, 2017. 3

[34] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition. In *ECCV*, pages 816–833, 2016. 1, 3, 7

[35] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. Kot. Global context-aware attention LSTM networks for 3D action recognition. In *CVPR*, 2017. 3, 7

[36] M. Liu, H. Liu, and C. Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68(Supplement C):346 – 362, 2017. 7

[37] D. Luvizon, D. Picard, and H. Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[38] S. Mathe, A. Pirinen, and C. Sminchisescu. Reinforcement learning for visual object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 3

[39] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu. Recurrent models of visual attention. In *NIPS*, pages 2204–2212, 2014. 3

[40] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *CVPR*, pages 4207–4215, 2016. 2

[41] N. Neverova, C. Wolf, G. Taylor, and F. Nebout. Moddrop: adaptive multi-modal gesture recognition. *IEEE TPAMI*, 38(8):1692–1706, 2016. 1, 2

[42] H. Rahmani and A. Mian. Learning a non-linear knowledge transfer model for cross-view action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 7

[43] H. Rahmani and A. Mian. 3d action recognition from novel viewpoints. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 7

[44] 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. 1, 2, 6, 7

[45] A. Shahroudy, T.-T. Ng, Y. Gong, and G. Wang. Deep multimodal feature analysis for action recognition in rgb+d videos. In *arXiv*, 2016. 7

[46] S. Sharma, R. Kiros, and R. Salakhutdinov. Action recognition using visual attention. *ICLR Workshop*, 2016. 3

[47] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014. 1

[48] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu. An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data. In *AAAI Conf. on AI*, 2016. 3, 7

[49] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. End-to-end memory networks. In *NIPS*, pages 2440–2448, 2015. 3

[50] L. Sun, K. Jia, K. Chen, D. Yeung, B. Shi, and S. Savarese. Lattice long short-term memory for human action recognition. In *ICCV*, 2017. 2

[51] L. Sun, K. Jia, K. Chen, D. Yeung, B. E. Shi, and S. Savarese. Lattice long short-term memory for human action recognition. In *ICCV*, 2017. 3

[52] L. Tao and R. Vidal. Moving poselets: A discriminative and interpretable skeletal motion representation for action recognition. In *ICCV Workshops*, pages 303–311, 2015. 1

[53] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. arXiv:1704.05737, 2017. 3

[54] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *CVPR*, pages 588–595, 2014. 7

[55] J. Wang, N. Xiaohan, X. Yin, W. Ying, and Z. Song-Chun. Cross-view action modeling, learning, and recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 6, 7

[56] P. Wang, W. Li, C. Li, and Y. Hou. Action Recognition Based on Joint Trajectory Maps with Convolutional Neural Networks. In *ACM Conference on Multimedia*, 2016. 1, 7

[57] D. Wu, L. Pigou, P.-J. Kindermans, N. D.-H. Le, L. Shao, J. Dambre, and J. Odobez. Deep dynamic neural networks for multimodal gesture segmentation and recognition. *IEEE TPAMI*, 38(8):1583–1597, 2016. 2

[58] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057, 2015. 3

[59] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *arXiv preprint arXiv:1507.05738*, 2015. 3

[60] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end Learning of Action Detection from Frame Glimpses in Videos. In *CVPR*, 2016. 3

[61] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, and D. Samaras. Two-person interaction detection using body-pose features and multiple instance learning. In *CVPR Workshop*, pages 28–35, 2012. 1

[62] M. Zanfir, M. Leordeanu, and C. Sminchisescu. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *ICCV*, pages 2752–2759, 2013. 2

[63] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *ICCV*, 2017. 7

[64] Z. Zhang, C. Wang, B. Xiao, W. Zhou, S. Liu, and C. Shi. Cross-view action recognition via a continuous virtual path. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 7