Multi-task Deep Learning for Real-Time 3D Human Pose Estimation and Action Recognition

Diogo C Luvizon, David Picard, Hedi Tabia

Multi-task Deep Learning for Real-Time 3D Human Pose Estimation and Action Recognition

Diogo C. Luvizon, David Picard, and Hedi Tabia

Abstract—Human pose estimation and action recognition are related tasks since both problems are strongly dependent on the human body representation and analysis. Nonetheless, most recent methods in the literature handle the two problems separately. In this work, we propose a multi-task framework for jointly estimating 2D or 3D human poses from monocular color images and classifying human actions from video sequences. We show that a single architecture can be used to solve both problems in an efficient way and still achieves state-of-the-art or comparable results at each task while running with a throughput of more than 100 frames per second. The proposed method benefits from high parameters sharing between the two tasks by unifying still images and video clips processing in a single pipeline, allowing the model to be trained with data from different categories simultaneously and in a seamlessly way. Additionally, we provide important insights for end-to-end training the proposed multi-task model by decoupling key prediction parts, which consistently leads to better accuracy on both tasks. The reported results on four datasets (MPII, Human3.6M, Penn Action and NTU RGB+D) demonstrate the effectiveness of our method on the targeted tasks. Our source code and trained weights are publicly available at https://github.com/dluvizon/deephar.

Index Terms—Human action recognition, Human pose estimation, Multitask deep learning, Neural networks.

1 Introduction

Human action recognition has been intensively studied in the last years, specially because it is a very challenging problem, but also due to the several applications that can benefit from it. Similarly, human pose estimation has also rapidly progressed with the advent of powerful methods based on convolutional neural networks (CNN) and deep learning. Despite the fact that action recognition benefits from precise body poses, the two problems are usually handled as distinct tasks in the literature [1], or action recognition is used as a prior for pose estimation [2], [3]. To the best of our knowledge, there is no recent method in the literature that tackles both problems in a joint way to the benefit of action recognition. In this paper, we propose a unique end-to-end trainable multi-task framework to handle human pose estimation and action recognition jointly, as illustrated in Fig. 1.

One of the major advantages of deep learning methods is their capability to perform end-to-end optimization. This is all the more true for multi-task problems, where related tasks can benefit from one another, as suggested by Kokkinos [4]. Action recognition and pose estimation are usually hard to be stitched together to perform a beneficial joint optimization, usually requiring 3D convolutions [5] or heatmaps transformations [6]. Detection based approaches require the non-differentiable argmax function to recover the joint coordinates as a post processing stage, which breaks the backpropagation chain needed for end-to-end learning. We propose to solve this problem by extending the differentiable soft-argmax [7], [8] for joint 2D and 3D pose estimation. This allows us to stack action recognition on top of pose estimation, resulting in a multi-task framework trainable from end-to-end.

In comparison with our previous work [9], we propose a new network architecture carefully designed for pose and action prediction simultaneously at different feature map resolutions. Each prediction is supervised and re-injected into the network for further refinement. Differently from [9], where we first predict poses then actions, here poses and actions are predicted in parallel and successively refined, strengthening the multi-task aspect of our method. Another improvement is the proposed depth estimation approach for 3D poses, which allows us to depart from learning the costly volumetric heat maps while improving the overall accuracy of the method.

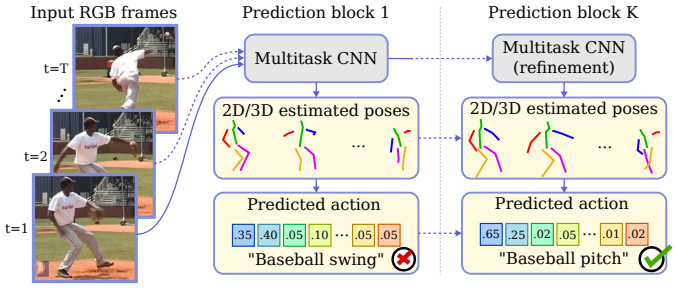

Fig. 1: The proposed multi-task approach for human pose estimation and action recognition. Our method provides 2D/3D pose estimation from single images or frame sequences. Pose and visual information are used to predict actions in a unified framework and both predictions are refined by K prediction blocks.

The main contributions of our work are presented as follows: *First*, we propose a new multi-task method for jointly estimating 2D/3D human poses and recognizing associated actions. Our method is simultaneously trained from end-to-end for both tasks with multimodal data, including still images and video clips.

- *D. C. Luvizon is with SAMSUNG Research Institute, Brazil. E-mail: diogo.luvizon@ensea.fr*
- *D. Picard is with LIGM, IMAGINE, Ecole des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-Vallée, France. E-mail: david.picard@enpc.fr*
- *H. Tabia is with IBISC, Univ Evry, Université Paris-Saclay, 91025, Evry, France. E-mail:hedi.tabia@univ-evry.fr*

Manuscript received January XX, 2019; revised August XX, XXXX.

Second, we propose a new regression approach for 3D pose estimation from single frames, benefiting at the same time from images "in-the-wild" with 2D annotated poses and 3D data. This has been proven a very efficient way to learn good visual features, which is also very important for action recognition. *Third*, our action recognition approach is based only on RGB images, from which we extract 3D poses and visual information. Despite that, our multi-task method achieves state-of-the-art on both 2D and 3D scenarios, even when compared with methods using ground-truth poses. *Fourth*, the proposed network architecture is scalable without any additional training procedure, which allows us to choose the right trade-off between speed and accuracy *a posteriori*. Finally, we show that the hard problem of multi-tasking pose estimation and action recognition can be tackled efficiently by a single and carefully designed architecture, handling both problems together and in a better way than separately. As a result, our method provides acceptable pose and action predictions at more than 180 frames per second (FPS), while achieving its best scores at 90 FPS on a customer GPU.

The remaining of this paper is organized as follows. In Section 2 we present a review of the most relevant works related to our method. The proposed multi-task framework is presented in Section 3. Extensive experiments on both pose estimation and action recognition are presented in Section 4, followed by our conclusions in Section 5.

2 Related Work

In this section, we present some of the most relevant methods related to our work, which are divided into *human pose estimation* and *action recognition*. Since an extensive literature review is out of the scope of the paper, we encourage the readers to refer to the surveys in [10], [11] for respectively pose estimation and action recognition.

2.1 Human Pose Estimation

2.1.1 2D Pose Estimation

The problem of human pose estimation has been intensively studied in the last years, from Pictorial Structures [12], [13], [14] to more recent CNN based approaches [15], [16], [17], [18], [19], [20], [21], [22], [23], [24]. We can identify from the literature two distinct families of methods for pose estimation: detection and regression based methods. Recent detection methods handle pose estimation as a heat map prediction problem, where each pixel in a heat map represents the detection score of a given body joint being localized at this pixel [25], [26]. Exploring the concepts of stacked architectures, residual connections, and multiscale processing, Newell *et al*. [27] proposed the Stacked Hourglass networks (SHG), which improved scores on 2D pose estimation challenges significantly. Since then, methods in the state of the art are frequently proposing complex variations of the SHG architecture. For example, Chu *et al*. [28] proposed an attention model based on conditional random field (CRF) and Yang *et al*. [29] replaced the residual unit from SHG by the Pyramid Residual Module (PRM). Very recently, [30] proposed a high-resolution network that keeps a high-resolution flow, resulting in more precise predictions. With the emergence of Generative Adversarial Networks (GANs) [31], Chou *et al*. [32] proposed to use a discriminative network to distinguish between estimated and target heat maps. This process could increase the quality of predictions, since the generator is stimulated to produce more plausible predictions. Another application of GANs in that sense is to enforce the structural representation of the human body [33].

However, all the previous mentioned detection based approaches do not provide body joint coordinates directly. To recover the body joints in (x, y) coordinates, predicted heat maps have to be converted to joint positions, generally using the argument of the maximum a posteriori probability (MAP), called $argmax$. On the other hand, regression based approaches use a nonlinear function to project the input image directly to the desired output, which can be the joint coordinates. Following this paradigm, Toshev and Szegedy [23] proposed a holistic solution based on cascade regression for body part regression and Carreira *et al*. [34] proposed the Iterative Error Feedback. The limitation of current regression methods is that the regression function is frequently sub-optimal. In order to tackle this weakness, the soft-argmax function [7] has been proposed to compute body joint coordinates from heat maps in a differentiable way.

2.1.2 3D Pose Estimation

Recently, deep architectures have been used to learn 3D representations from RGB images [35], [36], [37], [38], [39], [40] thanks to the availability of high precise 3D data [41], and are now able to surpass depth-sensors [42]. Chen and Ramanan [43] divided the problem of 3D pose estimation into two parts. First, they target 2D pose estimation considering the camera coordinates and second, the 2D estimated poses are matched to 3D representations by means of a nonparametric shape model. However, this is an ill-defined problem, since two different 3D poses could have the same 2D projection. Other methods propose to regress the 3D relative position of joints, which usually presents a lower variance than the absolute position. For example, Sun *et al*. [44] proposed a bone representation of the human body. However, since the errors are accumulative, such a structural transformation might effect tasks that depend on the extremities of the human body, like action recognition.

Pavlakos *et al*. [45] proposed the volumetric stacked hourglass architecture, but the method suffers from significant increase in the number of parameters and from the required memory to store all the gradients. A similar technique is used in [46], but instead of using argmax for coordinate estimation, the authors use a numerical integral regression, which is similar to the soft-argmax operation [9]. More recently, Yang *et al*. [47] proposed to use adversarial networks to distinguish between generated and ground truth poses, improving predictions on uncontrolled environments. Differently form our previous work in [9], we show that a volumetric representation is not required for 3D prediction. Similarly to methods on hand pose estimation [48] and on 3D human pose estimation [42], we predict 2D depth maps which encode the relative depth of each body joint.

2.2 Action Recognition

2.2.1 2D Action Recognition

In this section we revisited some methods that exploit pose information for action recognition. For example, classical methods for feature extraction have been used in [49], [50], where the key idea is to use body joint locations to select visual features in space and time. 3D convolutions have been stated as the best option to handle the temporal dimension of images sequences [51], [52], [53], but they involve a high number of parameters and cannot

efficiently benefit from the abundant still images during training. Another option to integrate the temporal aspect is by analysing motion from image sequences [1], [54], but these methods require the difficult estimation of optical flow. Unconstrained temporal and spatial analysis are also promising approaches to tackle action recognition, since it is very likely that, in a sequence of frames, some very specific regions in a few frames are more relevant than the remaining parts. Inspired on this observation, Baradel *et al*. [55] proposed an attention model called Glimpse Clouds, which learns to focus on specific image patches in space and time, aggregating the patterns and soft-assigning each feature to workers that contribute to the final action decision. The influence of occlusions could be alleviated by multi-view videos [56] and inaccurate pose sequences could be replaced by heat maps for better accuracy [57]. However, this improvement is not observed when pose predictions are sufficiently precise.

2D action recognition methods usually use the body joint information only to extract localized visual features [1], [49], as an attention mechanism. Methods that directly explore the body joints usually do not generate it [50] or present lower precision with estimated poses [51]. Our approach removes these limitations by performing pose estimation together with action recognition. As such, our model only needs the input RGB frames while still performing discriminative visual recognition guided by the estimated body joints.

2.2.2 3D Action Recognition

Differently from video based action recognition, 3D action recognition is mostly based on skeleton data as the primary information [58], [59]. With depth sensors such as the Microsoft Kinect, it is possible to capture 3D skeletal data without a complex installation procedure frequently required for motion capture systems (MoCap). However, due to the required infrared projector, depth sensors are limited to indoor environments, have a low range of operation, and are not robust to occlusions, frequently resulting in noisy skeletons. To cope with the noisy skeletons, Spatio-Temporal LSTM networks [60] have been widely used to learn the reliability of skeleton sequences or as an attention mechanism [61], [62]. In addition to the skeleton data, multimodal approaches can also benefit from visual cues [63]. In that direction, pose-conditioned attention mechanisms have been proposed [64] to focus on image patches centered around the hands.

Since our architecture predicts precise 3D poses from RGB frames, we do not have to cope with the noisy skeletons from Kinect. Moreover, we show in the experiments that, despite being based on temporal convolution instead of the more common LSTM, our system is able to reach state of the art performance on 3D action recognition, indicating that action recognition does not necessarily require long term memory.

3 Proposed Multi-task Approach

The goal of the proposed method is to jointly handle human pose estimation and action recognition, prioritizing the use of predicted poses on action recognition and benefiting from shared computations between the two tasks. For convenience, we define the input of our method as either a still RGB image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ or a video clip (sequence of images) $\mathbf{V} \in \mathbb{R}^{T \times H \times W \times 3}$, where T is the number of frames in a video clip and $H \times W$ is the frame size. This distinction is important because we handle pose estimation as a single frame problem. The outputs of our method for each frame are: predicted human pose $\hat{\mathbf{p}} \in \mathbb{R}^{N_j \times 3}$ and per body joint confidence score $\hat{\mathbf{c}} \in \mathbb{R}^{N_j \times 1}$, where N_j is the number of body joints. When taking a video clip as input, the method also outputs a vector of action probabilities $\hat{\mathbf{a}} \in \mathbb{R}^{N_a \times 1}$, where N_a is the number of action classes. To simplify notation, in this section we omit batch normalization layers and ReLU activations, which are used in between convolutional layers as a common practice in deep neural networks.

3.1 Network Architecture

Differently from our previous work [9] where poses and actions are predicted sequentially, here we want to strengthen the multi-task aspect of our method by predicting and refining poses and actions in parallel. This is implemented by the proposed architecture, illustrated in Fig. 2. Input images are fed through the entry-flow, which extracts low level visual features. The extracted features are then processed by a sequence of downscaling and upscaling pyramids indexed by $p \in \{1, 2, \ldots, P\}$, which are respectively composed of downscaling and upscaling units (DU and UU), and prediction blocks (PB), indexed by $l \in \{1, 2, \ldots, L\}$. Each PB is supervised on pose and action predictions, which are then re-injected into the network, producing a new feature map that is refined by further downscaling and upscaling pyramids. Downscaling or upscaling units are respectively composed by maxpooling or upsampling layers followed by a residual unit that is a standard or a depthwise separable convolution [65] with skip connection. These units are detailed in Fig. 3.

In order to be able to handle human poses and actions in a unified framework, the network can operate into two distinct modes: (i) *single frame* processing or (ii) *video clip* processing. In the first operational mode (single frame), only layers related to pose estimation are active, from which connections correspond to the blue arrows in Fig. 2. In the second operational mode (video clip), both pose estimation and action recognition layers are active. In this case, layers in the single frame processing part handle each video frame as a single sample in the batch. Independently on the operational mode, pose estimation is always performed from single frames, which prevents the method from depending on the temporal information for this task. For video clip processing, the information flow from single frame processing (pose estimation) and from video clip processing (action recognition) are independently propagated from one prediction block to another, as demonstrated in Fig. 2 respectively by blue and red arrows.

3.1.1 Multi-task Prediction Block

The main challenges related to the design of the network architecture is how to handle multimodal data (single frames and video clips) in a unified way and how to allow predictions refinement for both poses and actions. To this end, we propose a multi-task prediction block (PB), detailed in Fig. 4. In the PB, pose and action are simultaneously predicted and re-injected into the network for further refinement. In the global architecture, each PB is indexed by pyramid p and level l, and produces the following three feature maps:

$$\mathcal{X}_t^{p,l} \in \mathbb{R}^{H_f \times W_f \times N_f} \quad (1)$$

$$\mathcal{Z}_t^{p,l} \in \mathbb{R}^{H_f \times W_f \times N_f} \quad (2)$$

$$\mathcal{Y}^{p,l} \in \mathbb{R}^{T \times N_j \times N_v}. \quad (3)$$

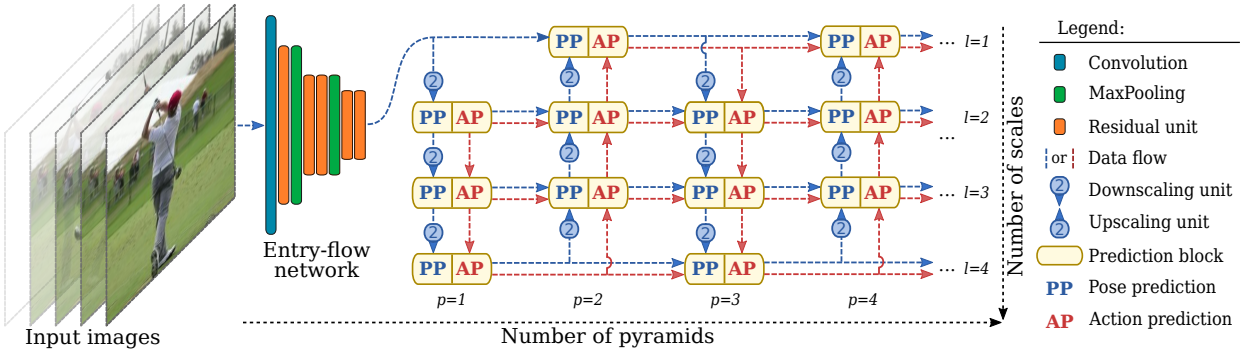

Fig. 2: Overview of the proposed multi-task network architecture. The entry-flow extracts feature maps from the input images, which are fed through a sequence of CNNs composed of prediction blocks (PB), downscaling and upscaling units (DU and UU), and simple (skip) connections. Each PB outputs supervised pose and action predictions that are refined by further blocks and units. The information flow related to pose estimation and action recognition are independently propagated from one prediction block to another, respectively depicted by blue and red arrows. See Fig. 3 and Fig. 4 for details about DU, UU, and PB.

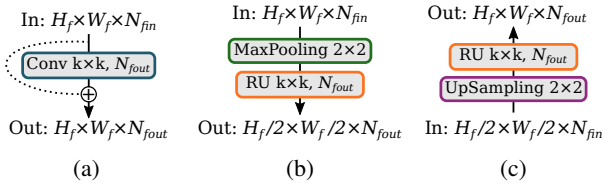

Fig. 3: Network elementary units: in (a) residual unit (RU), in (b) downscaling unit (DU), and in (c) upscaling unit (UU). N_{fin} and N_{fout} represent the input and output number of features, $H_f \times W_f$ is the feature map size, and k is the filter size.

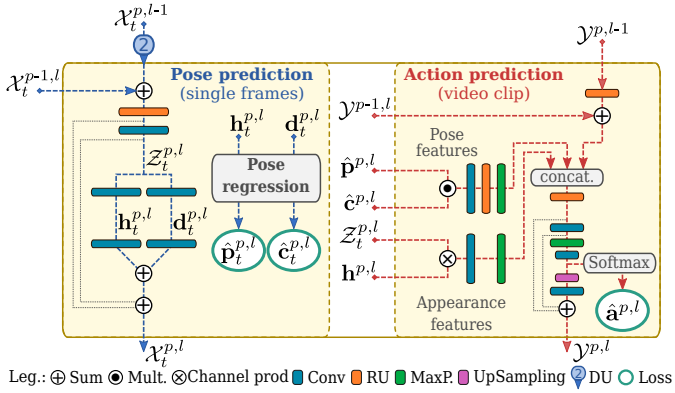

Fig. 4: Network architecture of prediction blocks (PB) for a downscaling pyramid. With the exception of the PB in the first pyramid, all PB get as input features from the previous pyramid in the same level ($\mathcal{X}_t^{p-1,l}$, $\mathcal{Y}^{p-1,l}$), and features from lower or higher levels ($\mathcal{X}_t^{p,l\mp1}$, $\mathcal{Y}^{p,l\mp1}$), depending if it composes a downscaling or an upscaling pyramid, respectively.

Namely, $\mathcal{X}_t^{p,l}$ is a tensor of single frame features, which is propagated from one PB to another, $\mathcal{Z}_t^{p,l}$ is a tensor of multi-task (single frame) features used for both pose and action, and $\mathcal{Y}^{p,l}$ is a tensor of video clip features, exclusively used for action predictions and also propagated from one PB to another. $t = \{1, \ldots, T\}$ is the index of single frames in a video clip, and N_f and N_v are respectively the size of single frame features and video clip features.

For pose estimation, prediction blocks take as input the single frame features $\mathcal{X}_t^{p-1,l}$ from the previous pyramid and the features $\mathcal{X}_t^{p,l\mp1}$ from lower or higher levels, respectively for downscaling and upscaling pyramids. A similar propagation of previous features $\mathcal{Y}^{p-1,l}$ and $\mathcal{Y}^{p,l\mp1}$ happens for action. Note that both $\mathcal{X}_t^{p,l}$ and $\mathcal{Y}^{p,l}$ feature maps are three-dimensional tensors (2D maps plus channels) that can be easily handled by 2D convolutions.

The tensor of multi-task features is defined by:

$$\mathcal{Z}_t^{'p,l} = RU(\mathcal{X}_t^{p-1,l} + DU(\mathcal{X}_t^{p,l-1})) \quad (4)$$

$$\mathcal{Z}_t^{p,l} = \mathbf{W}_z^{p,l} * \mathcal{Z}_t^{'p,l}, \quad (5)$$

where DU is the downscaling unit (replaced by UU for upscaling pyramids), RU is the residual unit, $*$ is a convolution, and $\mathbf{W}_z^{p,l}$ is a weight matrix. The choice of including a residual unit in Equation (4) was inspired from [27] and prevents $\mathcal{Z}_t^{'p,l}$ from becoming a direct summation of its previous terms. Then, $\mathcal{Z}_t^{p,l}$ is used to produce body joint probability maps:

$$\mathbf{h}_t^{p,l} = \Phi(\mathbf{W}_h^{p,l} * \mathcal{Z}_t^{p,l}), \quad (6)$$

and body joint depth maps:

$$\mathbf{d}_t^{p,l} = Sigmoid(\mathbf{W}_d^{p,l} * \mathcal{Z}_t^{p,l}), \quad (7)$$

where Φ is the spatial softmax [7], and $\mathbf{W}_h^{p,l}$ and $\mathbf{W}_d^{p,l}$ are weight matrices. Probability maps and body joint depth maps encode, respectively, the probability of a body joint being at a given location and the depth with respect to the root joint, normalized in the interval $[0, 1]$. Both $\mathbf{h}_t^{p,l}$ and $\mathbf{d}_t^{p,l}$ have shape $\mathbb{R}^{H_f \times W_f \times N_j}$.

3.2 Pose Regression

Once a set of body joint probability maps and depth maps are computed from multi-task features, we aim to estimate the corresponding 3D points by a differentiable and non-parametrized function. For that, we decouple the problem in *2D pose estimation* and *depth estimation*, and the final 3D pose is the concatenation of the intermediate parts.

3.2.1 The Soft-argmax Layer for 2D Estimation

Given a 2D input signal, the main idea is to consider that the argument of the maximum (*argmax*) can be approximated by the expectation of the input signal after being normalized to have the properties of a distribution. Indeed, for a sufficiently

pointy (Leptokurtic) distribution, the expectation should be close to the maximum a posteriori (MAP) estimation. For a 2D heat map as input, the normalized exponential function (softmax) can be used, since it alleviates the undesirable influences of values below the maximum and increases the "pointiness" of the resulting distribution, producing a probability map, as defined in Equation 6.

Let's define a single probability map for the jth joint as h^j, in such a way that $\mathbf{h} \equiv [h^1, \ldots, h^{N_j}]$. Then, the expected coordinates (x^j, y^j) are given by the function Ψ:

$$\Psi(h^j) = \left(\sum_{c=0}^{W_h} \sum_{r=0}^{H_h} \frac{c}{W_h} h_{r,c}, \sum_{c=0}^{W_h} \sum_{r=0}^{H_h} \frac{r}{H_h} h_{r,c} \right), \quad (8)$$

where $H_h \times W_h$ is the size of the input probability map, and l and c are line and column indexes of h. According to Equation 8, the coordinates (x^j, y^j) are constrained between the interval $[0, 1]$, which corresponds to the normalized limits of the input image.

3.2.2 Depth Estimation

Differently from our previous work [9], where volumetric heat maps were required to estimate the third dimension of body joints, here we use a similar apprach to [48], where specialized depth maps $\mathbf{d}$ are used to encode the depth information. Similarly to the probability maps decomposition from section 3.2.1, here we define d^j as a depth map for the jth body joint. Thus, the regressed depth coordinate z^j is defined by:

$$z^j = \sum_{c=0}^{W_h} \sum_{r=0}^{H_h} h^j_{r,c} d^j_{r,c}. \quad (9)$$

Since h^j is a normalized unitary and positive probability map, Equation 9 represents a spatially weighted pooling of depth map d^j based on the 2D body joint location.

3.2.3 Body Joint Confidence Scores

The probability of a certain body joint being present (even if occluded) in the image is computed by the maximum value in the corresponding probability map. Considering a pose layout with N_j body joints, the estimated joint confidence vector is represented by $\hat{\mathbf{c}} \in \mathbb{R}^{N_j \times 1}$. If the probability map is very pointy, this score is close to 1. On the other hand, if the probability map is uniform or has more than one region with high response, the confidence score drops.

3.2.4 Pose Re-injection

As systematically noted in recent works [25], [26], [27], [45], predictions re-injection is a very efficient way to improve precision on estimated poses. Differently from all previous methods based on direct heat map regression, our approach can benefit from prediction re-injection at different resolutions, since our pose regression method is invariant to the feature map resolution. Specifically, in each PB at different pyramid and different level, we compute a new set of features $\mathcal{X}^{p,l}_t$ based on features from previous blocks and on the current prediction, as follows:

$$\mathcal{X}^{p,l}_t = \mathbf{W}^{p,l}_r * \mathbf{h}^{p,l}_t + \mathbf{W}^{p,l}_s * \mathbf{d}^{p,l}_t + \mathcal{Z}'^{p,l}_t + \mathcal{Z}^{p,l}_t, \quad (10)$$

where $\mathbf{W}^{p,l}_r$ and $\mathbf{W}^{p,l}_s$ are weight matrices related to the re-injection of 2D pose and depth information, respectively. With this approach, further PB at different pyramids and levels are able to refine predictions, considering different sets of features at different resolutions.

3.3 Human Action Recognition

Another important advantage in our method is its ability to integrate high level pose information with low level visual features in a multi-task framework. This characteristic allows sharing the single frame processing pipeline for both pose estimation and visual features extraction. Additionally, visual features are trained using both action sequences and still images captured "in-the-wild", which have been proven as a very efficient way to learn robust visual representations. As shown in Fig. 4, the action prediction part takes as input two different sources of information: *pose features* and *appearance features*. Additionally, similarly to the pose prediction part, action features from previous pyramids ($\mathcal{Y}^{p-1,l}$) and levels ($\mathcal{Y}^{p,l\mp1}$) are also aggregated in each prediction.

3.3.1 Pose Features

In order to explore the rich information encoded with body joint positions, we convert a sequence of T poses with N_j joints each into an image-like representation. Similar representations were previously used in [64], [66]. We choose to encode the temporal dimension as the vertical axis, the joints as the horizontal axis, and the coordinates of each point ((x, y) for 2D, (x, y, z) for 3D) as the channels. With this approach, we can use classical 2D convolutions to extract patterns directly from the temporal sequence of body joints. The predicted coordinates of each body joints are pondered by their confidence scores, thus points that are not present in the image (and consequently cannot be correctly predicted) have less influence on action recognition. A graphical representation of pose features is presented in Fig. 5a.

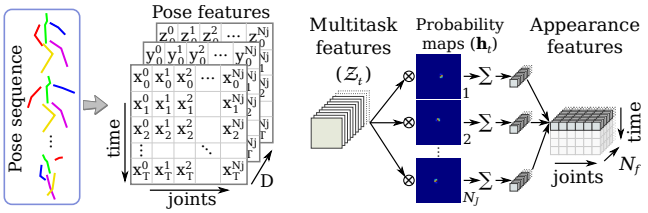

Fig. 5: Extraction of (a) pose and (b) appearance features.

3.3.2 Appearance Features

In addition to the pose information, visual cues are very important to action recognition, since they bring contextual information. In our method, localized visual information is encoded as *appearance features*, which are extracted in a similar process to the one of pose features, with the difference that the first relies on local visual information instead of joint coordinates. In order to extract localized appearance features, we multiply each channel from the tensor of multi-task features $\mathcal{Z}^{p,l}_t \in \mathbb{R}^{H_f \times W_f \times N_f}$ by each channel from the probability maps $\mathbf{h}_t \in \mathbb{R}^{H_f \times W_f \times N_j}$ (outer product of N_f and N_j), which is learned as a byproduct of the pose estimation process. Then, the spatial dimensions are collapsed by a sum, resulting in the appearance features for time t of size $\mathbb{R}^{N_j \times N_f}$. For a sequence of frames, we concatenate each appearance feature map for $t = \{1, 2, \ldots, T\}$ resulting in the video clip appearance features $\mathcal{V} \in \mathbb{R}^{T \times N_j \times N_f}$. To clarify this process, a graphical representation is shown in Fig. 5b.

We argue that our multi-task framework has two benefits for the appearance based part: First, it is computationally very

efficient since most part of the computations are shared. Second, the extracted visual features are more robust since they are trained simultaneously for different but related tasks and on different datasets.

3.3.3 Action Features Aggregation and Re-injection

Some actions are hard to be distinguished from others only by the high level pose representation. For example, the actions *drink water* and *make a phone call* are very similar if we take into account only the body joints, but are easily separated if we have the visual information corresponding to the objects cup and phone. On the other hand, other actions are not directly related to visual information but with body movements, like *salute* and *touch chest*, and in this case the pose information can provide complementary information. In our method, we combine visual cues and body movements by aggregating pose and appearance features. This aggregation is a straightforward process, since both feature types have the same spacial dimensions.

Similarly to the single frame features re-injection mechanism discussed in section 3.2.4, our approach also allows action features re-injection, as detailed in the action prediction part in Fig. 4. We demonstrate in the experiments that this technique also improves action recognition results with no additional parameters.

3.3.4 Decoupled Action Poses

Since the multi-task architecture is trained simultaneously on pose estimation and on action recognition, we may have an effect of competing gradients from poses and actions, specially in the predicted poses, which are used as the output for the first task and as the input for the second task. To mitigate that influence, late in the training process, we propose to decouple estimated poses (used to compute pose scores) from action poses (used by the action recognition part) as illustrated in Fig. 6.

Fig. 6: Decoupled poses for action prediction. The weight matrix $\mathbf{W}'_h$ is initialized with a copy of $\mathbf{W}_h$ after the main training process. The same is done to depth maps ($\mathbf{W}_d$ and $\mathbf{d}$).

Specifically, we first train the network on pose estimation for about one half of the full training iterations, then we replicate only the last layers that project the multi-task feature map $\mathcal{Z}$ to heat maps and depth maps (parameters $\mathbf{W}_h$ and $\mathbf{W}_d$), resulting in a "copy" of probability maps $\mathbf{h}'$ and depth maps $\mathbf{d}'$. Note that this replica corresponds to a simple 1×1 convolution from the feature space to the number of joints, which is almost insignificant in terms of parameters and computations. The "copy" of this layer is a new convolutional layer with its weights $\mathbf{W}'$ initialized with $\mathbf{W}$. Finally, for the remaining training, the action recognition part propagates its loss through the replica poses. This process allows the original pose predictions to stay specialized on the first task, while the replicated poses absorb partially the action gradients and are optimized accordingly to the action recognition task. Despite the replicated poses not being directly supervised in the final training stage (which corresponds to a few more epochs), we show in our experiments that they still remain coherent with supervised estimated poses.

4 Experiments

In this section, we present quantitative and qualitative results by evaluating the proposed method on two different tasks and on two different modalities: human pose estimation and human action recognition on 2D and 3D scenarios. Since our method relies on body coordinates, we consider four publicly available datasets mostly composed of full poses, which are detailed as follows.

4.1 Datasets

MPII Human Pose Dataset [67] is a well known 2D human pose dataset composed of about 25K images collected from YouTube videos. 2D poses were manually annotated with up to 16 body joints. **Human3.6M** [41] is a 3D human pose dataset composed by videos with 11 subjects performing 17 different activities, all recorded simultaneously by 4 cameras. High precision 3D poses were captured by a MoCap system, from which 17 body joints are used for evaluation. **Penn Action** [68] is a 2D dataset for action recognition composed by 2,326 videos with sports people performing 15 different actions. Human poses were manually annotated with up to 13 body joints. **NTU RGB+D** [69] is a large scale 3D action recognition dataset composed by 56K videos in Full HD with 60 actions performed by 40 different actors and recorded by 3 cameras in 17 different configurations. Each color video has an associated depth map video and 3D Kinect poses.

4.1.1 Evaluation Metrics

On 2D pose estimation, we evaluate our method on the MPII validation set composed of 3K images, using the probability of correct keypoints measure with respect to the head size (PCKh) [67]. On 3D pose estimation, we evaluate our method on Human3.6M by measuring the mean per joint position error (MPJPE) after alignment of the root joint. We follow the most common evaluation protocol [37], [39], [44], [45], [47] by taking five subjects for training (S1, S5, S6, S7, S8) and evaluating on two subjects (S9, S11) on one every 64 frames. We use ground truth person bounding boxes for a fair comparison with previous methods on single person pose estimation. We report results using a single cropped bounding box per sample.

On action recognition, we report results using the percentage of correct action classification score. We use the proposed evaluation protocol for Penn Action [49], splitting the data as 50/50 for training/testing, and the more realistic cross-subject scenario for NTU, on which 20 subjects are used for training, and the remaining are used for testing. Our method is evaluated on *single-clip* and/or *multi-clip*. In the first case, we crop a single clip with T frames in the middle of the video. In the second case, we crop multiple video clips temporally spaced of $T/2$ frames one from another, and the final predicted action is the average decision among all clips from one video.

In our experiments, we consider two scenarios: A) 2D pose estimation and action recognition, on which we use respectively MPII and Penn Action datasets, and B) 3D pose estimation and action recognition, using MPII, Human3.6M, and NTU datasets.

4.2 Implementation and Training Details

4.2.1 Function Loss

For the pose estimation task, we train the network using the elastic net loss [70] function on predicted poses:

$$\mathcal{L}_{\mathbf{p}} = \frac{1}{N_j} \sum_{j=1}^{N_j} \left(\|\hat{\mathbf{p}}^j - \mathbf{p}^j\|_1 + \|\hat{\mathbf{p}}^j - \mathbf{p}^j\|_2^2 \right), \qquad (11)$$

where $\hat{\mathbf{p}}^j$ and $\mathbf{p}^j$ are respectively the estimated and the ground truth positions of the jth body joint. The same loss is used for both 2D and 3D cases, but only available values ((x, y) for 2D and (x, y, z) for 3D) are taken into account for backpropagation, depending on the dataset. We use poses in the camera coordinate system, with (x, y) laying on the image plane and z corresponding to the depth distance, normalized in the interval $[0, 1]$, where the top-left image corner corresponds to $(0, 0)$, and the bottom-right image corner corresponds to $(1, 1)$. For depth normalization, the root joint is assumed to have $z = 0.5$, and a range of 2 meters is used to represent the remaining joints. If a given body joint falls outside the cropped bounding box on training, we set the ground truth confidence flag $\mathbf{c}^j$ to zero, otherwise we set it to one. The ground truth confidence information is used to supervise predicted joint confidence scores $\hat{\mathbf{c}}$ with the binary cross entropy loss. Despite giving an additional information, the supervision on confidence scores has negligible influence on the precision of estimated poses. For the action recognition part, we use categorical cross entropy loss on predicted actions.

4.2.2 Network Architecture

Since the pose estimation part is the most computationally expensive, we chose to use separable convolutions with kernel size equals to 5×5 for single frame layers and standard convolutions with kernel size equals to 3×3 for video clip processing layers (action recognition layers). We performed experiments with the network architecture using 4 levels and up to 8 pyramids ($L = 4$ and $P = 8$). No further significant improvement was noticed on pose estimation by using more than 8 pyramids. On action recognition, this limit was observed at 4 pyramids. For that reason, when using the full model with 8 pyramids, the action recognition part starts only at the $5th$ pyramid, reducing the computational load. In our experiments, we used normalized RGB images of size $256 \times 256 \times 3$ as input, which are reduced to a feature map of size $32 \times 32 \times 288$ by the entry flow network, corresponding to level $l = 1$. At each level, the spatial resolution is reduced by a factor of 2 and the size of features is arithmetically increased by 96. For action recognition, we used $N_v = 160$ and $N_v = 192$ features for Penn Action and NTU, respectively.

4.2.3 Multi-task Training

For all the experiments, we first initialize the network by training pose estimation only, for about 32k iterations with mini batches of 32 images (equivalent to 40 epochs on MPII). Then, all the weights related to pose estimation are fixed and only the action recognition part is trained for 2 and 50 epochs, respectively for Penn Action and NTU datasets. Finally, the full network is trained in a multi-task scenario, simultaneously for pose estimation and action recognition, until the validation scores plateau. Training the network on pose estimation for a few epochs provides a good general initialization and a better convergence of the action recognition part. The intermediate training stage of action recognition has two objectives: first, it is useful to allow a good initialization of the action part, since it is built on top of the pre-initialized pose estimator; and second, it is about 3 times faster than performing multi-task training directly while resulting in similar scores. This process is specially useful for NTU, due to the large amount of training data. The training procedure takes about one day for the pose estimation initialization, then two/three days for the remaining process for Penn Action/NTU, using a desktop GeForce GTX 1080Ti GPU.

For initialization on pose estimation, the network was optimized with RMSprop and initial learning rate of 0.001. For action and multi-task training, we use RMSprop for Penn Action with learning rate reduced by a factor of 0.1 after 15 and 25 epochs, and, for NTU, a vanilla SGD with Nesterov momentum of 0.9 and initial learning rate of 0.01, reduced by a factor of 0.1 after 50 and 55 epochs. We weight the loss on body joint confidence scores and action estimations by a factor of 0.01, since the gradients from the cross entropy loss are much stronger than the gradients from the elastic net loss on pose estimation. This parameter was empirically chosen and we did not observe a significant variation in the results with slightly different values (e.g., with 0.02). Each iteration is performed on 4 batches of 8 frames, composed of random images for pose estimation and video clips for action. We train the model by alternating one batch containing pose estimation samples only and another batch containing action samples only. This strategy resulted in slightly better results compared to batches composed of mixed pose and action samples. We augment training data by performing random rotations from $-40°$ to $+40°$, scaling from 0.7 to 1.3, video temporal subsampling by a factor from 3 to 10, random horizontal flipping, and random color shifting. On evaluation, we also subsampled Penn Action/NTU videos by a factor of 6/8, respectively.

4.3 Evaluation on 3D Pose Estimation

Our results compared to previous approaches are shown in Table 1. Our multi-task method achieves the state-of-the-art average prediction error of 48.6 millimeters on Human3.6M for 3D pose estimation, improving our previous work [9] by 4.6 mm. Considering only the pose estimation task, our average error is 49.5 mm, 0.9 mm higher than the multi-tasking result, which shows the benefit of multi-task training for 3D pose estimation. For the activity "Sit down", which is the most challenging case, we improve previous methods (*e.g.* Yang *et al.* [47]) by 21 mm. The generalization of our method is demonstrated by qualitative results of 3D pose estimation for all datasets in Fig. 10. Note that a single model and a single training procedure was used to produce all the images and scores, including 3D pose estimation and 3D action recognition, as discussed in the following.

4.4 Evaluation on Action Recognition

For action recognition, we evaluate our method considering both 2D and 3D scenarios. For the first, a single model was trained using MPII for single frames (pose estimation) and Penn Action for video clips. In the second scenario, we use Human3.6M for 3D pose supervision, MPII for data augmentation, and NTU video clips for action. Similarly, a single model was trained for all the reported 3D pose and action results.

For 2D, the pose estimation was trained using mixed data from MPII (80%) and Penn Action (20%), using 16 body joints. Results are shown in Table 2. We reached the state-of-the-art action classification score of 98.7% on Penn Action, improving our previous work [9] by 1.3%. Our method outperformed all previous methods, including the ones using ground truth (manually annotated) poses.

For 3D, we trained our multi-task network using mixed data from Human3.6M (50%), MPII (37.5%) and NTU (12.5%) for pose estimation and NTU video clips for action recognition. Our results compared to previous methods are presented in Table 3. Our approach reached 89.9% of correctly classified actions on

TABLE 1: Comparison with previous work on Human3.6M evaluated using the mean per joint position error (MPJPE, in millimeters) metric on reconstructed poses.

Methods	Direction	Discuss	Eat	Greet	Phone	Posing	Purchase	Sitting
Pavlakos *et al.* [45]	67.4	71.9	66.7	69.1	71.9	65.0	68.3	83.7
Mehta *et al.* [39]⋆	52.5	63.8	55.4	62.3	71.8	52.6	72.2	86.2
Martinez *et al.* [37]	51.8	56.2	58.1	59.0	69.5	55.2	58.1	74.0
Sun *et al.* [44]†	52.8	54.8	54.2	54.3	61.8	53.1	53.6	71.7
Yang *et al.* [47]†	51.5	58.9	50.4	57.0	62.1	49.8	52.7	69.2
Sun *et al.* [46]†	–	–	–	–	–	–	–	–
3D heat maps (ours [9], only H36M)	61.7	63.5	56.1	60.1	60.0	57.6	64.6	75.1
3D heat maps (ours [9])†	49.2	51.6	47.6	50.5	51.8	48.5	51.7	61.5
Ours (single-task)†	**43.7**	**48.8**	**45.6**	**46.2**	**49.3**	**43.5**	**46.0**	**56.8**
Ours (multi-task)†	**43.2**	**48.6**	**44.1**	**45.9**	**48.2**	**43.5**	**45.5**	**57.1**
Methods	Sit Down	Smoke	Photo	Wait	Walk	Walk Dog	Walk Pair	Average
Pavlakos *et al.* [45]	96.5	71.4	76.9	65.8	59.1	74.9	63.2	71.9
Mehta *et al.* [39]⋆	120.0	66.0	79.8	63.9	48.9	76.8	53.7	68.6
Martinez *et al.* [37]	94.6	62.3	78.4	59.1	49.5	65.1	52.4	62.9
Sun *et al.* [44] †	86.7	61.5	67.2	53.4	47.1	61.6	53.4	59.1
Yang *et al.* [47]†	85.2	57.4	65.4	58.4	60.1	43.6	47.7	58.6
Sun *et al.* [46]†	–	–	–	–	–	–	–	49.6
3D heat maps (ours [9], only H36M)	95.4	63.4	73.3	57.0	48.2	66.8	55.1	**63.8**
3D heat maps (ours [9])†	70.9	53.7	60.3	48.9	44.4	57.9	48.9	53.2
Ours (single-task)†	**67.8**	**50.5**	**57.9**	**43.4**	**40.5**	**53.2**	**45.6**	**49.5**
Ours (multi-task)†	**64.2**	**50.6**	**53.8**	**44.2**	**40.0**	**51.1**	**44.0**	**48.6**

⋆ Method not using ground-truth bounding boxes.
† Methods using extra 2D data for training.

TABLE 2: Results for action recognition on Penn Action. Results are given as the percentage of correctly classified actions. Our method uses extra 2D pose data from MPII for training.

Methods	RGB	Optical Flow	Annot. poses	Estimated poses	Acc.
Nie *et al.* [49]	✓	-	-	✓	85.5
Iqbal *et al.* [3]	-	-	-	✓	79.0
	✓	✓	-	✓	92.9
Cao *et al.* [51]	✓	-	✓	-	98.1
	✓	-	-	✓	95.3
Du *et al.* [54]⋆	✓	✓	-	✓	97.4
Liu *et al.* [57]†	✓	-	✓	-	98.2
	✓	-	-	✓	91.4
Our previous work [9]	✓	-	✓	-	98.6
	✓	-	-	✓	97.4
Ours (single-clip)	✓	-	-	✓	**98.2**
Ours (multi-clip)	✓	-	-	✓	**98.7**

⋆ Including UCF101 data; † using add. deep features.

TABLE 3: Comparison results on NTU cross-subject for 3D action recognition. Results are given as the percentage of correctly classified actions. Our method uses extra pose data from MPII and H36M for training.

Methods	RGB	Kinect poses	Estimated poses	Acc. cross subject
Shahroudy *et al.* [69]	-	✓	-	62.9
Liu *et al.* [60]	-	✓	-	69.2
Song *et al.* [62]	-	✓	-	73.4
Liu *et al.* [61]	-	✓	-	74.4
Shahroudy *et al.* [63]	✓	✓	-	74.9
Liu *et al.* [57]	✓	-	✓	78.8
	-	✓	-	77.1
Baradel *et al.* [64]	✓	⋆	-	75.6
	✓	✓	-	84.8
Baradel *et al.* [71]	-	-	-	86.6
Our previous work [9]	✓	-	✓	85.5
Ours	✓	-	✓	**89.9**

⋆ Ground truth poses used on test to select visual features.

NTU, which is a strong result considering the hard task of classifying among 60 different actions in the cross-subject split. Our method improves previous results by at least 3.3% and our previous work by 4.4%, which shows the effectiveness of the proposed approach.

4.5 Ablation Study

4.5.1 Network Design

We performed several experiments on the proposed network architecture in order to identify its best arrangement for solving both tasks with the best performance *vs* computational cost trade-off. In Table 4, we show the results on 2D pose estimation and on action recognition considering different network layouts. For example, in the first line, a single PB is used at pyramid 1 and level 2. In the second line, a pair of full downscaling and upscaling pyramids are used, but with supervision only at the last PB. This results in 97.5% of accuracy on action recognition and 84.2% on PCKh for pose estimation. An equivalent network is used in the third line, but then with supervision on all PB blocks, which brings an improvement of 0.9% on pose and 0.6% on action, with the same number of parameters. Note that the networks from the second and third lines are exactly the same, but in the first case, only the last PB is supervised, while in the latter all PB receive supervision. Finally, the last line shows results with the full network, reaching 88.3% on MPII and 98.2% on Penn Action (single-clip), with a single multi-task model.

4.5.2 Pose and Appearance Features

The proposed method benefits from both pose and appearance features, which are complementary to the action recognition task.

TABLE 4: The influence of the network architecture on pose estimation and on action recognition, evaluated respectively on MPII validation set (PCKh@0.5, single-crop) and on Penn Action (classification accuracy, single-clip). Single-PB are indexed by pyramid p and level l, and P and L represent the total number of pyramids and levels on Multi-PB scheme.

Network	Param.	No. PB	PCKh	Action acc.
Single-PB ($p = 1, l = 2$)	2M	1	74.3	97.2
Single-PB ($p = 2, l = 1$)	10M	1	84.2	97.5
Multi-PB ($P = 2, L = 4$)	10M	6	85.1	98.1
Multi-PB ($P = 8, L = 4$)	**26M**	**24**	**88.3**	**98.2**

Additionally, the confidence score $\hat{\mathbf{c}}$ is also complementary to pose itself and leads to marginal action recognition gains if used to weight pose predictions. Similar results are achieved if confidence scores are concatenated to poses. In Table 5, we present results on pose estimation and on action recognition for different features extraction strategies. Considering pose features or appearance features alone, the results on Penn Action are respectively 97.4% and 97.9%, respectively 0.7% and 0.2% lower than combined features. We also show in the last row the influence of decoupled action poses, resulting in a small gain of 0.1% on action scores and 0.3% on pose estimation, which shows that decoupling action poses brings additional improvements, specially for pose estimation. When not considering decoupled poses, note that the best score on pose estimation happens when poses are not directly used for action, which also supports the evidence of competing losses.

TABLE 5: Results with pose and appearance features alone, combined pose and appearance features, and decoupled poses. Experiments with a Multi-PB network with $P = 2$ and $L = 4$.

Action features	MPII val. PCKh	PennAction Acc.
Pose features only	84.9	97.7
Appearance features only	85.2	97.9
Combined	85.1	98.1
Combined + decoupled poses	**85.4**	**98.2**

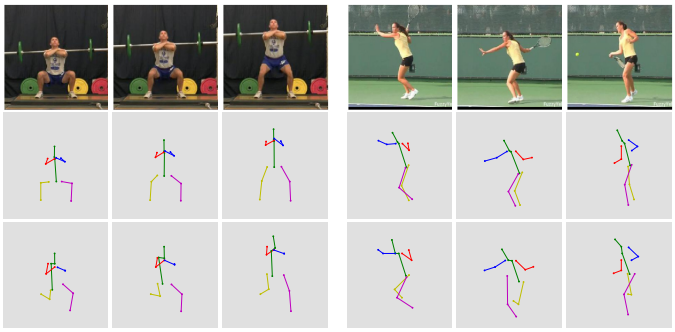

Fig. 7: Two sequences of RGB images (top), predicted supervised poses (middle), and decoupled action poses (bottom).

Additionally, we can observe that decoupled action poses remain coherent with supervised poses, as shown in Fig. 7, which suggests that the initial pose supervision is a good initialization overall. Nonetheless, in some cases, decoupled probability maps can drift to regions in the image more relevant for action recognition, as illustrated in Fig. 8. For example, feet heat maps can drift to objects in the hands, since the last is more informative with respect to the performed action.

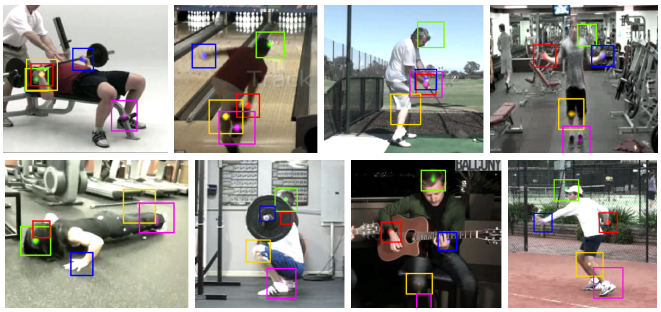

Fig. 8: Drift of decoupled probability maps from their original positions (head, hands and feet) used as an attention mechanism for appearance features extraction. Bounding boxes are drawn here only to highlight the regions with high responses. Each color corresponds to a specific body part (see Fig. 7).

4.5.3 *Single-task vs. multi-task*

In this part we compare the results on human action recognition considering single-task and multi-task training protocols. In Table 6, in the first row, are shown results on PennAction and NTU datasets considering training with action supervision only, *i.e.*, with the full network architecture (including pose estimation layers) but without pose supervision. In the second row we show the results when using the manually annotated poses from PennAction for pose supervision. We did not use NTU (Kinect) poses for supervision since they are very noisy. From this, we can notice an improvement of almost 10% on PennAction, only by adding pose supervision. When mixing with MPII data, it further increases 0.8%. On NTU, multi-tasking improves a significant 1.9%. We believe that the improvement of multi-tasking on PennAction is much more evident because this is a small dataset, therefore it is difficult to learn good representations for complex actions without explicit pose information. On the contrary, NTU is a large scale dataset, more suitable for learning approaches. As a consequence, the gap between single and multi-task on NTU is smaller, but still relevant.

TABLE 6: Results comparing the effect of single and multi-task training for action recognition.

Training protocol	PennAction Acc.	NTU Acc.
Single-task (action only)	87.5	88.0
Multi-task (same dataset)	97.4	–
Multi-task (+MPII +H36M for 3D)	98.2	89.9

4.5.4 *Inference Speed*

Once the network is trained, it can be easily cut to perform faster inferences. For instance, the full model with 8 pyramids can be cut at the 4th or 2nd pyramids, which generally degrades the performance, but allows faster predictions. To show the trade-off between precision and speed, we cut the trained multi-task model at different prediction blocks and estimate the throughput in frames per second (FPS), evaluating pose estimation precision and action recognition classification accuracy. We consider mini batches with 16 images for pose estimation and single video clips of 8 frames for action. The results are shown in Fig. 9. For both 2D and 3D scenarios, the best predictions are at more than 90 FPS. For the 3D scenario, pose estimation on Human3.6M can be performed at more than 180 FPS and still reach a competitive result of 57.3 millimeters error, while for action recognition on

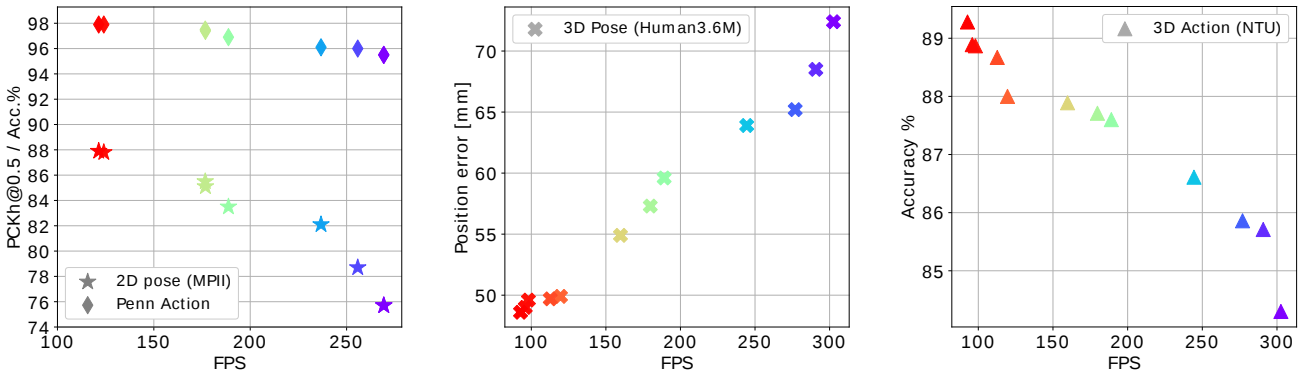

Fig. 9: Inference speed of the proposed method considering 2D (a) and 3D (b,c) scenarios. A single multi-task model was trained for each scenario. The trained models were cut *a posteriori* for inference analysis. Markers with gradient colors from purple to red represent respectively network inferences from faster to slower.

NTU, at the same speed, we still obtain state of the art results with 87.7% of correctly classified actions, or even comparable results with recent approaches at more than 240 FPS. Finally, we show our results for both 2D and 3D scenarios compared to previous methods in Table 7, considering different inference speed. Note that our method is the only to perform both pose and action estimation in a single prediction, while achieving state-of-the-art results at a very high speed.

TABLE 7: Results on all tasks with the proposed multi-task model compared to recent approaches using RGB images and/or estimated poses on MPII PCKh validation set (higher is better), Human3.6M MPJPE (lower is better), Penn Action and NTU RGB+D action classification accuracy (higher is better).

Methods	MPII PCKh	H36M MPJPE	PennAction *half/half*	NTU RGB+D Cross-sub.
Pavlakos *et al.* [45]	-	71.9	-	-
Mehta *et al.* [39]	-	68.6	-	-
Martinez *et al.* [37]	-	62.9	-	-
Sun *et al.* [44]	-	59.1	-	-
Yang *et al.* [47]	**88.6**	58.6	-	-
Sun *et al.* [46]	87.3	49.6	-	-
Nie *et al.* [49]	-	-	85.5	-
Iqbal *et al.* [3]	-	-	92.9	-
Cao *et al.* [51]	-	-	95.3	-
Du *et al.* [54]	-	-	97.4	-
Shahroudy *et al.* [63]	-	-	-	74.9
Baradel *et al.* [71]	-	-	-	86.6
Ours [9] @ 85 fps	-	53.2	97.4	85.5
Ours 2D @ 240 fps	85.5	-	**97.5**	-
Ours 2D @ 120 fps	88.3	-	**98.7**	-
Ours 3D @ 240 fps	80.7	63.9	-	86.6
Ours 3D @ 180 fps	83.8	57.3	-	**87.7**
Ours 3D @ 90 fps	87.0	**48.6**	-	**89.9**

5 Conclusion

In this work, we presented a new approach for human pose estimation and action recognition using multi-task deep learning. The proposed method for 3D pose provides highly precise estimations with low resolution feature maps and departs from requiring the expensive volumetric heat maps by predicting specialized depth maps per body joints. The proposed CNN architecture, along with the pose regression method, allows multi-scale pose and action supervision and re-injection, resulting in a highly efficient densely supervised approach. Our method can be trained with mixed 2D and 3D data, benefiting from precise indoor 3D data, as well as "in-the-wild" images manually annotated with 2D poses. This has demonstrated significant improvements for 3D pose estimation. The proposed method can also be trained with single frames and video clips simultaneously and in a seamless way.

More importantly, we show that the hard problem of multi-tasking human poses and action recognition can be handled by a carefully designed architecture, resulting in a better solution for each task than learning them separately. In addition, we show that joint learning human poses results in consistent improvement of action recognition. Finally, with a single training procedure, our multi-task model can be cut at different levels for pose and action predictions, resulting in a highly scalable approach.

Acknowledgments

This work was partially supported by the Brazilian National Council for Scientific and Technological Development (CNPq) – Grant 233342/2014-1.

References

[1] G. Chéron, I. Laptev, and C. Schmid, "P-CNN: Pose-based CNN Features for Action Recognition," in *ICCV*, 2015.

[2] A. Yao, J. Gall, and L. Van Gool, "Coupled action recognition and pose estimation from multiple views," *International Journal of Computer Vision*, vol. 100, no. 1, pp. 16–37, Oct 2012.

[3] U. Iqbal, M. Garbade, and J. Gall, "Pose for action - action for pose," *FG-2017*, 2017.

[4] I. Kokkinos, "Ubernet: Training a 'universal' convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] M. Zolfaghari, G. L. Oliveira, N. Sedaghat, and T. Brox, "Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[6] V. Choutas, P. Weinzaepfel, J. Revaud, and C. Schmid, "Potion: Pose motion representation for action recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[7] D. C. Luvizon, H. Tabia, and D. Picard, "Human pose regression by combining indirect part detection and contextual information," *Computers and Graphics*, vol. 85, pp. 15 – 22, 2019.

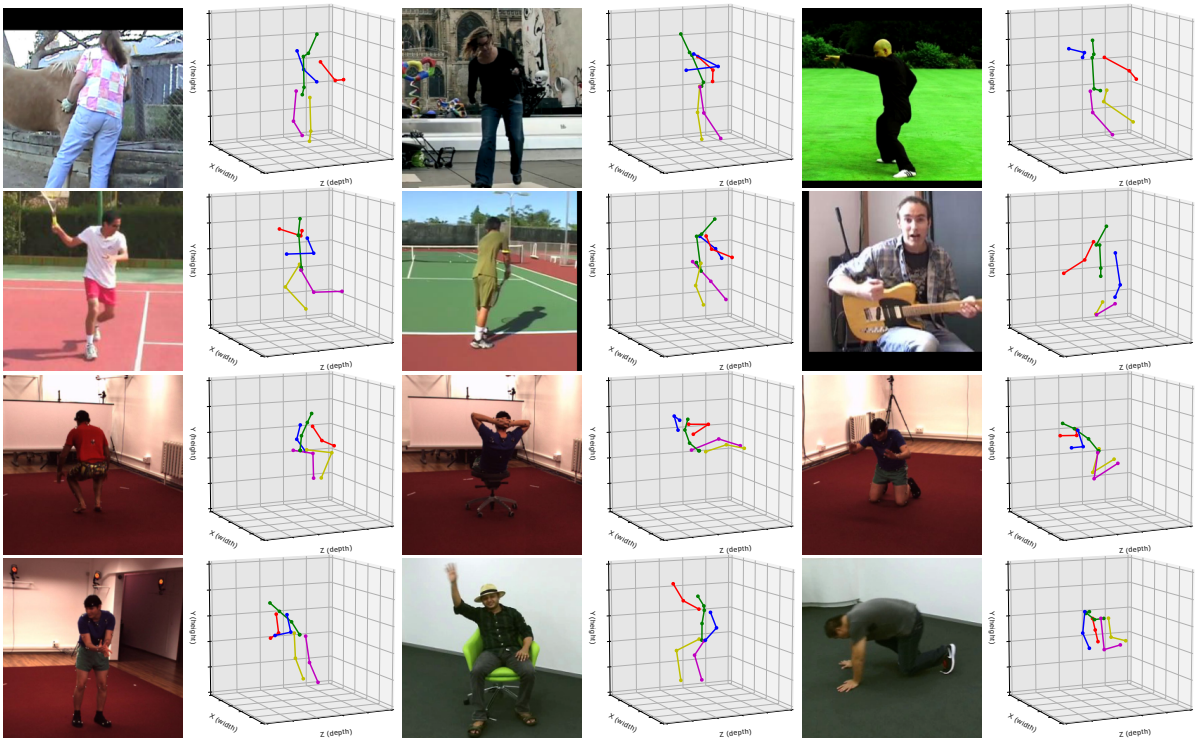

Fig. 10: Predicted 3D poses from RGB images for both 2D and 3D datasets.

[8] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned Invariant Feature Transform," *European Conference on Computer Vision (ECCV)*, 2016.

[9] D. C. 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.

[10] N. Sarafianos, B. Boteanu, B. Ionescu, and I. A. Kakadiaris, "3d human pose estimation: A review of the literature and analysis of covariates," *Computer Vision and Image Understanding*, vol. 152, no. Supplement C, pp. 1 – 20, 2016.

[11] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey," *Image and Vision Computing*, vol. 60, no. Supplement C, pp. 4 – 21, 2017, regularization Techniques for High-Dimensional Data Analysis.

[12] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 1014–1021.

[13] M. Dantone, J. Gall, C. Leistner, and L. V. Gool, "Human Pose Estimation Using Body Parts Dependent Joint Regressors," in *Computer Vision and Pattern Recognition (CVPR)*, June 2013, pp. 3041–3048.

[14] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, "Poselet Conditioned Pictorial Structures," in *Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 588–595.

[15] G. Ning, Z. Zhang, and Z. He, "Knowledge-guided deep fractal neural networks for human pose estimation," *IEEE Transactions on Multimedia*, vol. PP, no. 99, pp. 1–1, 2017.

[16] I. Lifshitz, E. Fetaya, and S. Ullman, *Human Pose Estimation Using Deep Consensus Voting*. Cham: Springer International Publishing, 2016, pp. 246–260.

[17] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele, "DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[18] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, "DeeperCut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model," in *European Conference on Computer Vision (ECCV)*, May 2016.

[19] U. Rafi, I. Kostrikov, J. Gall, and B. Leibe, "An efficient convolutional network for human pose estimation," in *BMVC*, vol. 1, 2016, p. 2.

[20] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[21] V. Belagiannis, C. Rupprecht, G. Carneiro, and N. Navab, "Robust optimization for deep regression," in *International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 2830–2838.

[22] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using Convolutional Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 648–656.

[23] A. Toshev and C. Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks," in *Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1653–1660.

[24] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman, "Deep convolutional neural networks for efficient pose estimation in gesture videos," in *Asian Conference on Computer Vision (ACCV)*, 2014.

[25] A. Bulat and G. Tzimiropoulos, "Human pose estimation via Convolutional Part Heatmap Regression," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 717–732.

[26] G. Gkioxari, A. Toshev, and N. Jaitly, "Chained Predictions Using Convolutional Neural Networks," *European Conference on Computer Vision (ECCV)*, 2016.

[27] A. Newell, K. Yang, and J. Deng, "Stacked Hourglass Networks for Human Pose Estimation," *European Conference on Computer Vision (ECCV)*, pp. 483–499, 2016.

[28] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang, "Multi-context attention for human pose estimation," *arXiv preprint arXiv:1702.07432*, 2017.

[29] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang, "Learning feature pyramids for human pose estimation," in *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[30] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[31] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.

[32] C. Chou, J. Chien, and H. Chen, "Self adversarial training for human pose estimation," *CoRR*, vol. abs/1707.02439, 2017.

[33] Y. Chen, C. Shen, X.-S. Wei, L. Liu, and J. Yang, "Adversarial posenet: A structure-aware convolutional network for human pose estimation," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[34] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, "Human pose estimation with iterative error feedback," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4733–4742.
[35] X. Zhou, M. Zhu, G. Pavlakos, S. Leonardos, K. G. Derpanis, and K. Daniilidis, "Monocap: Monocular human motion capture using a CNN coupled with a geometric prior," *CoRR*, vol. abs/1701.02354, 2017.
[36] D. Tome, C. Russell, and L. Agapito, "Lifting from the deep: Convolutional 3d pose estimation from a single image," in *CVPR*, July 2017.
[37] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in *ICCV*, 2017.
[38] B. Tekin, P. Márquez-Neila, M. Salzmann, and P. Fua, "Fusing 2d uncertainty and 3d cues for monocular body pose estimation," *CoRR*, vol. abs/1611.05708, 2016.
[39] D. Mehta, H. Rhodin, D. Casas, O. Sotnychenko, W. Xu, and C. Theobalt, "Monocular 3d human pose estimation using transfer learning and improved CNN supervision," *CoRR*, vol. abs/1611.09813, 2016.
[40] A.-I. Popa, M. Zanfir, and C. Sminchisescu, "Deep multitask architecture for integrated 2d and 3d human sensing," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
[41] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *TPAMI*, vol. 36, no. 7, pp. 1325–1339, jul 2014.
[42] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt, "Vnect: Real-time 3d human pose estimation with a single rgb camera," in *ACM Transactions on Graphics*, vol. 36, 2017.
[43] C.-H. Chen and D. Ramanan, "3d human pose estimation = 2d pose estimation + matching," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
[44] X. Sun, J. Shang, S. Liang, and Y. Wei, "Compositional human pose regression," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
[45] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3D human pose," in *the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
[46] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei, "Integral human pose regression," in *The European Conference on Computer Vision (ECCV)*, September 2018.
[47] W. Yang, W. Ouyang, X. Wang, J. S. J. Ren, H. Li, and X. Wang, "3d human pose estimation in the wild by adversarial learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
[48] U. Iqbal, P. Molchanov, T. Breuel Juergen Gall, and J. Kautz, "Hand pose estimation via latent 2.5d heatmap regression," in *The European Conference on Computer Vision (ECCV)*, September 2018.
[49] B. Xiaohan Nie, C. Xiong, and S.-C. Zhu, "Joint action recognition and pose estimation from video," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
[50] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.
[51] C. Cao, Y. Zhang, C. Zhang, and H. Lu, "Body joint guided 3d deep convolutional descriptors for action recognition," *CoRR*, vol. abs/1704.07160, 2017.
[52] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *CVPR*, July 2017.
[53] G. Varol, I. Laptev, and C. Schmid, "Long-term Temporal Convolutions for Action Recognition," *TPAMI*, 2017.
[54] W. Du, Y. Wang, and Y. Qiao, "Rpan: An end-to-end recurrent pose-attention network for action recognition in videos," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
[55] F. Baradel, C. Wolf, J. Mille, and G. W. Taylor, "Glimpse clouds: Human activity recognition from unstructured feature points," in *Computer Vision and Pattern Recognition (CVPR) (To appear)*, June 2018.
[56] D. Wang, W. Ouyang, W. Li, and D. Xu, "Dividing and aggregating network for multi-view action recognition," in *The European Conference on Computer Vision (ECCV)*, September 2018.
[57] M. Liu and J. Yuan, "Recognizing human actions as the evolution of pose estimation maps," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
[58] D. C. Luvizon, H. Tabia, and D. Picard, "Learning features combination for human action recognition from skeleton sequences," *Pattern Recognition Letters*, 2017.
[59] L. L. Presti and M. L. Cascia, "3d skeleton-based human action classification: A survey," *Pattern Recognition*, vol. 53, pp. 130–147, 2016.
[60] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal lstm with trust gates for 3d human action recognition," in *ECCV*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham, 2016, pp. 816–833.
[61] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global context-aware attention lstm networks for 3d action recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
[62] S. Song, C. Lan, J. Xing, W. Z. (wezeng), and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *AAAI Conference on Artificial Intelligence*, 2017.
[63] A. Shahroudy, T.-T. Ng, Y. Gong, and G. Wang, "Deep multimodal feature analysis for action recognition in rgb+d videos," *TPAMI*, 2017.
[64] F. Baradel, C. Wolf, and J. Mille, "Pose-conditioned spatio-temporal attention for human action recognition," *arxiv*, vol. 1703.10106, 2017.
[65] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
[66] 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.
[67] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D Human Pose Estimation: New Benchmark and State of the Art Analysis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
[68] W. Zhang, M. Zhu, and K. G. Derpanis, "From actemes to action: A strongly-supervised representation for detailed action understanding," in *ICCV*, Dec 2013, pp. 2248–2255.
[69] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+d: A large scale dataset for 3d human activity analysis," in *CVPR*, June 2016.
[70] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society, Series B*, vol. 67, pp. 301–320, 2005.
[71] F. Baradel, C. Wolf, J. Mille, and G. W. Taylor, "Glimpse Clouds: Human Activity Recognition from Unstructured Feature Points," in *Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018.

Diogo Carbonera Luvizon received the B.Sc. degree in Electrical Engineering and the M.Sc. in Image Processing and Graphics from the Federal University of Technology - Paraná (Brazil), in 2015, and the Ph.D. in Computer Science from the Cergy Paris Université (France), in 2019. His main research interests include machine learning and deep learning algorithms for computer vision, humans and 3D scene understanding.

David Picard received the M.Sc. in Electrical Engineering in 2005, the Ph.D. in Image and Signal Processing in 2008 and the Habilitation in Computer Science in 2017. He joined the ETIS laboratory at ENSEA Graduate School (France) in 2010 as an associate professor. Since 2019, he is senior research scientist at École des Ponts ParisTech (France). His research interests include computer vision and machine learning, with a focus on kernel methods, deep learning, and distributed learning.

Hedi Tabia received, in 2008, the M.S. degree in computer science from the INSA of Rouen - Public school of engineers, France. In 2011, he obtained the Ph.D. degree in computer science from the University of Lille. From October 2011 to August 2012, he held a postdoctoral research associate position at the IEF laboratory (University of Paris-sud). During 2012-2019, he was an associate professor at the ENSEA. Since September 2019 he is a professor at Université Paris Saclay.