

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/323410987>

# 2D/3D Pose Estimation and Action Recognition using Multitask Deep Learning

Article · February 2018

CITATIONS

56

READS

1,493

3 authors, including:



[Diogo Carbonera Luvizon](#)

Université de Cergy-Pontoise

16 PUBLICATIONS 461 CITATIONS

[SEE PROFILE](#)



[David Picard](#)

École des Ponts ParisTech

84 PUBLICATIONS 1,097 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Visual representation using tensors [View project](#)



Reassembly from pairwise comparison [View project](#)

# 2D/3D Pose Estimation and Action Recognition using Multitask Deep Learning

Diogo C. Luvizon<sup>1</sup>, David Picard<sup>1,2</sup>, Hedi Tabia<sup>1</sup>

<sup>1</sup>ETIS UMR 8051, Paris Seine University, ENSEA, CNRS, F-95000, Cergy, France

<sup>2</sup>Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6, F-75005 Paris, France

{diogo.luvizon, picard, hedi.tabia}@ensea.fr

## Abstract

Action recognition and human pose estimation are closely related but both problems are generally handled as distinct tasks in the literature. In this work, we propose a multitask framework for jointly 2D and 3D pose estimation from still images and human action recognition from video sequences. We show that a single architecture can be used to solve the two problems in an efficient way and still achieves state-of-the-art results. Additionally, we demonstrate that optimization from end-to-end leads to significantly higher accuracy than separated learning. The proposed architecture can be trained with data from different categories simultaneously in a seamlessly way. The reported results on four datasets (MPII, Human3.6M, Penn Action and NTU) demonstrate the effectiveness of our method on the targeted tasks.

## 1. Introduction

Human action recognition and pose estimation have received an important attention in the last years, not only because of their many applications, such as video surveillance and human-computer interfaces, but also because they are still challenging tasks. Pose estimation and action recognition are usually handled as distinct problems [14] or the last is used as a prior for the first [57, 22]. Despite the fact that pose is of extreme relevance for action recognition, to the best of our knowledge, there is no method in the literature that solves both problems in a joint way to the benefit of action recognition. In that direction, our work proposes unique end-to-end trainable multitask framework to handle 2D and 3D human pose estimation and action recognition jointly, as presented in Figure 1.

One of the major advantages of deep learning is its capability to perform end-to-end optimization. As suggested by Kokkinos [24], this is all the more true for multitask problems, where related tasks can benefit from one another. Recent methods based on deep convolutional neural networks (CNNs) have achieved impressive results on both 2D and

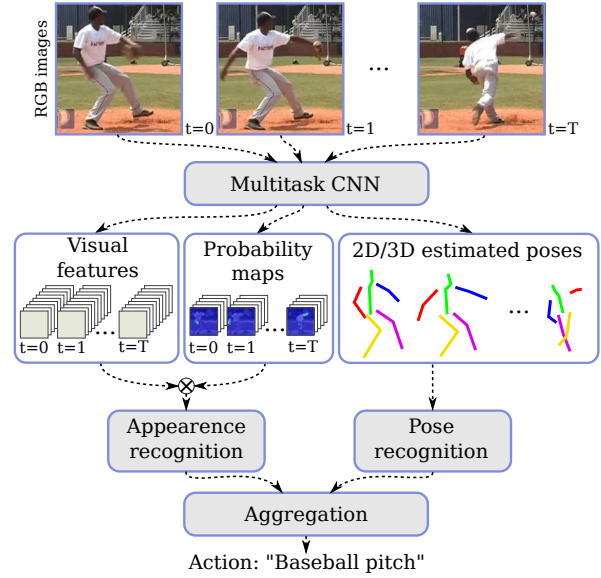


Figure 1. The proposed multitask approach for pose estimation and action recognition. Our method provides 2D/3D pose estimation from single images or frame sequences. Pose and visual information are used to predict actions in a unified framework.

3D pose estimation tasks thanks to the rise of new architectures and the availability of large amounts of data [33, 35]. Similarly, action recognition has recently been improved by using deep neural networks relying on human pose [3]. We believe both tasks have not yet been stitched together to perform a beneficial joint optimization because most pose estimation methods perform heat map prediction. These 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 [28, 58] for joint 2D and 3D pose estimation. This allows us to stack action recognition on top of pose estimation, resulting in a multitask framework trainable from end-to-end. We present our contributions as follows: **First**, the proposed pose estimation method achieves

state-of-the-art results on 3D pose estimation and the most accurate results among regression methods for 2D pose estimation. **Second**, the proposed pose estimation method is based on still images, so it benefits from images “in the wild” for both 2D and 3D predictions. This has been proven a very efficient way to learn 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 pose and visual information. Despite that, we reached state-of-the-art results on both 2D and 3D scenarios, even when compared with methods using ground-truth poses. **Fourth**, the pose estimation method can be trained with multiple types of datasets simultaneously, which makes it able to generalize 3D predictions from 2D annotated data.

The rest of this paper is organized as follows. In section 2 we present a review of the related work. The proposed framework is presented in sections 3 and 4, respectively for the regression method for pose estimation and human action recognition. Our extensive experiments are shown in section 5, followed by our conclusions in section 6.

## 2. Related work

In this section, we present some of the most relevant methods to our work, which are divided into *human pose estimation* and *action recognition*. Since an extensive literature review is prohibitive here due to the limited size of the paper, we encourage the readers to refer to the surveys in [43, 19] for respectively pose estimation and action recognition.

### 2.1. Human pose estimation

**2D pose estimation.** The problem of human pose estimation has been intensively studied in the last years, from Pictorial Structures [2, 17, 37] to more recent CNN approaches [34, 25, 38, 20, 41, 54, 5, 51, 52, 36]. From the literature, we can see that there are two distinct families of methods for pose estimation: detection based and regression based methods. Detection based methods handle pose estimation as a heat map prediction problem, where each pixel in a heat map represents the detection score of a corresponding joint [7, 18]. Exploring the concepts of stacked architectures, residual connections, and multiscale processing, Newell *et al.* [33] proposed the Stacked Hourglass Network, which improved scores on 2D pose estimation challenges significantly. Since then, methods in the state of the art are proposing complex variations of the Stacked Hourglass architecture. For example, Chu *et al.* [16] proposed an attention model based on conditional random field (CRF) and Yang *et al.* [56] replaced the residual unit by a Pyramid Residual Module (PRM). Generative Adversarial Networks (GANs) have been used to improve the capacity of learning

structural information [13] as well as to refine the heat maps by learning more plausible predictions [15],

However, detection approaches do not provide joint coordinates directly. To recover the pose in  $(x, y)$  coordinates, the *argmax* function is usually applied as a post-processing step. On the other hand, regression based approaches use a nonlinear function that maps the input directly to the desired output, which can be the joint coordinates. Following this paradigm, Toshev and Szegedy [52] proposed a holistic solution based on cascade regression for body part detection and Carreira *et al.* [9] proposed the Iterative Error Feedback. The limitation of regression methods is that the regression function is frequently sub-optimal. In order to tackle this weakness, the Soft-argmax function [28] has been proposed to convert heat maps directly to joint coordinates and consequently allow detection methods to be transformed into regression methods. The main advantage of regression methods over detection ones is that they often are fully differentiable. This means that the output of the pose estimation can be used in further processing and the whole system can be fine-tuned.

**3D pose estimation.** Recently, deep architectures have been used to learn precise 3D representations from RGB images [60, 50, 30, 49, 31, 39], thanks to the availability of high quality data [21], and are now able to surpass depth-sensors [32]. Chen and Ramanan [11] divided the problem of 3D pose estimation into two parts. First, they handle the 2D pose estimation considering the camera coordinates and second, the estimated poses are matched to 3D representations by means of a nonparametric shape model. A bone representation of the human pose was proposed to reduce the data variance [47], however, such a structural transformation might effect negatively tasks that depend on the extremities of the human body, since the error is accumulated as we go away from the root joint. Pavlakos *et al.* [35] proposed the volumetric stacked hourglass architecture. However, the method suffers from the significant increase in the number of parameters and in the required memory to store all the gradients. In our approach, we also propose an intermediate volumetric representation for 3D poses, but we use a much lower resolution than in [35] and still are able to increase significantly the state-of-the-art results, since our method is based on a continuous regression function.

### 2.2. Action recognition

**2D action recognition.** Action recognition from videos is considered a difficult problem because it involves high level abstraction, and furthermore the temporal dimension is not easily handled. Previous approaches have explored classical methods for features extraction [55, 23], where the key idea is to use body joint locations to select visual features in space and time. 3D convolutions have been stated recently as the option that gives the highest classification

scores [8, 10, 53], but they involve high number of parameters, require an elevated amount of memory for training, and cannot efficiently benefit from the abundant still images for training. Action recognition is improved by attention models that focus on body parts [4] and two-stream networks can be used to merge both RGB images and the costly optical flow maps [14].

Most 2D action recognition methods use the body joint information only to extract localized visual features, as an attention mechanism. The few methods that directly explore the body joints do not generate it, therefore they are limited to datasets that provide skeletal data. 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 estimated body joints.

**3D action recognition.** Differently from video based action recognition, 3D action recognition is mostly based on skeleton data as the primary information [29, 40]. With recently available 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 use of infrared projectors, these depth sensors are limited to indoor environments. Moreover, they have a low range precision and are not robust to occlusions, frequently resulting in noisy skeletons.

To cope with noisy skeletons, Spatio-Temporal LSTM networks have been widely used by applying a gating mechanism [26] to learn the reliability of skeleton sequences or by using attention mechanisms [27, 46]. In addition to the skeleton data, multimodal approaches can also benefit from the visual cues [45]. In that direction, Baradel *et al.* [3] proposed the Pose-conditioned Spatio-Temporal attention mechanism by using the skeleton sequences for both spatial and temporal attention mechanisms, while action classification is based on pose and appearance features extracted from patches on the hands.

Since our architecture predicts high precision 3D skeleton from the input 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.

### 3. Human pose estimation

Our approach for human pose estimation is a regression method, similarly to [28, 47, 9]. We extended the Soft-argmax function to handle 2D and 3D pose regression in a unified way. The details of our approach are explained as follows.

#### 3.1. Regression-based approach

The human pose regression problem is defined by the input RGB image  $\mathbf{I} \in \mathbb{R}^{W \times H \times 3}$ , the output estimated pose  $\hat{\mathbf{p}} \in \mathbb{R}^{N_J \times D}$  with  $N_J$  body joints of dimension  $D$ , and a regression function  $f_r$ , as given by the following equation:

$$\hat{\mathbf{p}} = f_r(\mathbf{I}, \theta_r), \quad (1)$$

where  $\theta_r$  is a set of trainable parameters of function  $f_r$ . The objective is to optimize the parameters  $\theta_r$  in order to minimize the error between the estimated pose  $\hat{\mathbf{p}}$  and the ground truth pose  $\mathbf{p}$ . In order to implement this function, we use a deep CNN. As the pose estimation is the first part of our multitask approach, the function  $f_r$  has to be differentiable in order to allow end-to-end optimization. This is made possible by the Soft-argmax, which is a differentiable alternative to the *argmax* function and can be used to convert heat maps to  $(x, y)$  joint coordinates proposed in [28].

##### 3.1.1 Network architecture

The network architecture has its entry flow based on Inception-V4 [48] that is used to provide basic features extraction. Then, similarly to what is found in [28],  $K$  prediction blocks are used to refine estimations, from which we use the last prediction  $\mathbf{p}'_K$  as our estimated pose  $\hat{\mathbf{p}}$ . Each prediction block is composed of eight residual depth-wise convolutions separated into three different resolutions. As a byproduct, we also have access to low-level visual features and to the intermediate joint probability maps that are indirectly learned thanks to the Soft-argmax layer. In our method for action recognition, both visual features and joint probability maps are used to produce appearance features, as detailed in section 4.2. A graphical representation of the pose regression network is shown in Figure 2.

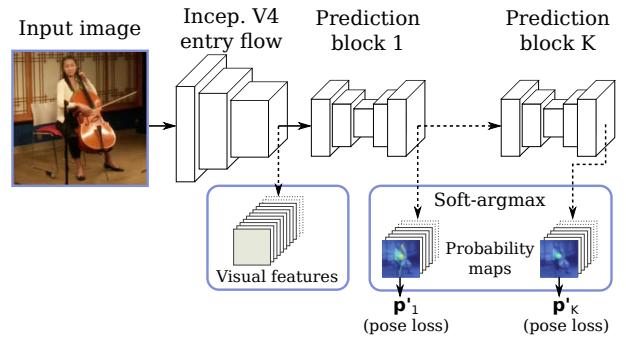


Figure 2. Human pose regression approach from a single RGB frame. The input image is fed through a CNN composed by one entry flow and  $K$  prediction blocks. Predictions are refined at each prediction block.

### 3.1.2 The Soft-argmax layer

An intuitive graphical explanation of the Soft-argmax layer is shown in Figure 3. Given an input signal, the main idea is to consider that the argument of the maximum  $\mu$  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. The normalized exponential function (Softmax) is used, since it alleviates the undesirable influences of values below the maximum and increases the “pointiness” of the resulting distribution. For a 2D heat map as input, the normalized signal can be interpreted as the *probability map* of a joint being at position  $(x, y)$ , and the expected value for the joint position is given by the expectation on the normalized signal:

$$\Psi(\mathbf{x}) = \left( \sum_{c=0}^{W_{\mathbf{x}}} \sum_{l=0}^{H_{\mathbf{x}}} \frac{c}{W_{\mathbf{x}}} \Phi(\mathbf{x})_{l,c}, \sum_{c=0}^{W_{\mathbf{x}}} \sum_{l=0}^{H_{\mathbf{x}}} \frac{l}{H_{\mathbf{x}}} \Phi(\mathbf{x})_{l,c} \right)^T, \quad (2)$$

where  $\mathbf{x}$  is the input heat map with dimension  $W_{\mathbf{x}} \times H_{\mathbf{x}}$  and  $\Phi$  is the Softmax normalization function.

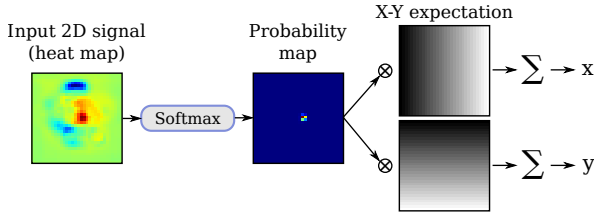


Figure 3. Graphical representation of the Soft-argmax operation for 2D input signals (heat maps). The outputs are the coordinates  $x$  and  $y$  that approximates the maximum in the input signal.

### 3.1.3 Joint visibility

The probability of a certain joint being visible in the image is computed by the Sigmoid function on the maximum value in the corresponding input heat map. Considering a pose layout with  $N_J$  joints, the joint visibility vector is represented by  $\mathbf{v} \in \mathbb{R}^{N_J \times 1}$ . Remark that the *visibility* information is unrelated to the joint *probability map*, since the latter always sums to one.

## 3.2. Unified 2D/3D pose estimation

We extended the 2D pose regression to 3D scenarios by expanding 2D heat maps to volumetric representations. We define  $N_d$  stacked 2D heat maps, corresponding to the depth resolution. The prediction in  $(x, y)$  coordinates is performed by applying the Soft-argmax operation on the averaged heat maps, and the  $z$  component is regressed by apply-

ing a one-dimensional Soft-argmax on the volumetric representation averaged in both  $x$  and  $y$  dimensions, as depicted in Figure 4. The advantage of splitting the pose prediction into two parts,  $(x, y)$  and  $z$ , is that we maintain the 2D heat maps as a byproduct, which is useful for extracting appearance features, as explained in section 4.2.

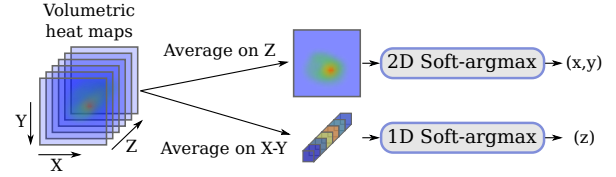


Figure 4. Unified 2D/3D pose estimation by using volumetric heat maps.

With the proposed unified approach, we can train the network with mixed 2D and 3D data. For the first case, only the gradients corresponding to  $(x, y)$  are backpropagated. As a result, the network can be jointly trained with high precise 3D data from motion capture systems and very challenging still images collected in outdoor environments, which are usually manually annotated.

## 4. Human action recognition

One of the most important advantages in our proposed method is the ability to integrate high level pose information with low level visual features in a multitask framework. This characteristic allows to share the network entry flow for both pose estimation and visual features extraction. Additionally, the 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 on Figure 1, the proposed action recognition approach is divided into two parts, one based on a sequence of body joints coordinates, which we call *pose-based recognition*, and the other based on a sequence of visual features, which we call *appearance-based recognition*. The result of each part is combined to estimate the final action label. In this section, we give a detailed explanation about each recognition branch, as well as how we extend single frame pose estimation to extract temporal information from a sequence of frames.

### 4.1. Pose-based recognition

In order to explore the high level information encoded with body joint positions, we convert a sequence of  $T$  poses with  $N_J$  joints each into an image-like representation. 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 a temporal sequence of body joints. Since the pose estimation method is based on still images, we use a time distributed abstraction to process a video clip, which is a straightforward technique to handle both single images and video sequences.

We propose a fully convolutional neural network to extract features from input poses and to produce *action heat maps* as shown on Figure 5. The idea is that for actions depending only on few body joints, such as *shaking hands*, fully-connected layers will require zeroing non-related joints, which is a very difficult learning problem. On the contrary, 2D convolutions enforce this sparse structure without manually choosing joints and are thus easier to learn. Furthermore, different joints have very different coordinates variations and a filter matching, *e.g.*, hand patterns will not respond to feet patterns equally. Such patterns are then combined in subsequent layers in order to produce more discriminative activations until we obtain action maps with a depth equals to the number of actions.

To produce the output probability of each action for a video clip, a pooling operation on the action maps has to be performed. In order to be more sensitive to the strongest responses for each action, we use the *max plus min* pooling followed by a Softmax activation. Additionally, inspired by the human pose regression method, we refine predictions by using a stacked architecture with intermediate supervision in  $K$  prediction blocks. The action heat maps from each prediction block are then re-injected into the next action recognition block.

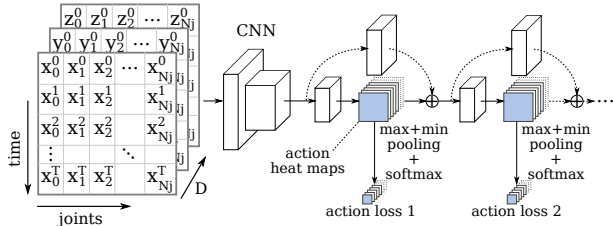


Figure 5. Representation of the architecture for action recognition from a sequence of  $T$  frames of  $N_J$  body joints. The  $z$  coordinates are used for 3D action recognition only. The same architecture is used for appearance-based recognition, except that the input are the appearance features instead of body joints.

## 4.2. Appearance-based recognition

The appearance based part is similar to the pose based part, with the difference that it relies on local appearance features instead of joint coordinates. In order to extract localized appearance features, we multiply the tensor of visual features  $F_t \in \mathbb{R}^{W_f \times H_f \times N_f}$  obtained at the end of the global entry flow by the probability maps  $M_t \in \mathbb{R}^{W_f \times H_f \times N_J}$  obtained at the end of the pose estimation part, where  $W_f \times H_f$  is the size of the feature maps,  $N_f$  is the number of features, and  $N_J$  is the number of

joints. Instead of multiplying each value individually as in the Kronecker product, we multiply each channel, resulting in a tensor of size  $\mathbb{R}^{W_f \times H_f \times N_J \times N_f}$ . 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 features for  $t = \{0, 1, \dots, T\}$  resulting in the video clip appearance features  $V \in \mathbb{R}^{T \times N_J \times N_f}$ . To clarify the above appearance features extraction process, a graphical representation is shown on Figure 6.

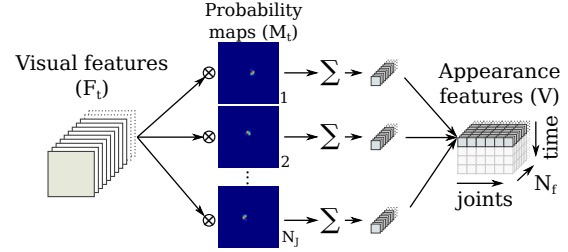


Figure 6. Appearance features extraction from low level visual features and body parts probability maps for a single frame. For a sequence of  $T$  frames, the appearance features are stacked vertically producing a tensor where each line corresponds to one input frame.

The appearance features are fed into an action recognition network similar to the pose-based action recognition block presented on Figure 5 with visual features replacing the coordinates of the body joints.

We argue that our multitask framework has two benefits for the appearance based part: First, it is very computationally efficient since most part of the computations are shared. Second, the extracted visual features are more robust since they are trained simultaneously for different tasks and on different datasets.

## 4.3. Action aggregation

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 that case the pose information can provide complementary information.

In order to explore the contribution from both pose and appearance models, we combine the respective predictions using a fully-connected layer with Softmax activation, which gives the final prediction of our model.

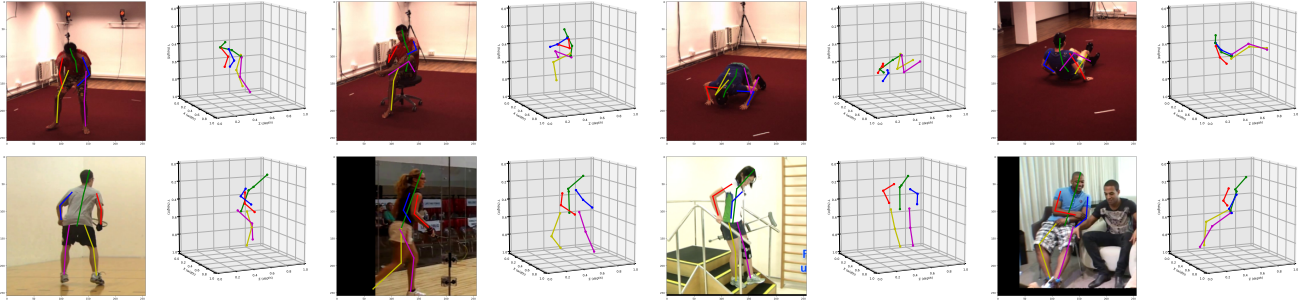


Figure 7. Predicted 3D poses from Human3.6M (top row) and MPII (bottom row) datasets.

## 5. Experiments

In this section we present the experimental evaluation of our method in four different categories using four challenging datasets. We show the robustness and the flexibility of our proposed multitask approach. The four categories are divided into two problems: human pose estimation and action recognition. For both cases, we evaluate our approach on 2D and 3D scenarios.

### 5.1. Datasets

We evaluate our method on four different datasets: on MPII [1] and on Human3.6M [21] for respectively 2D and 3D pose estimation, and on Penn Action [59] and NTU RGB+D [44] for 2D and 3D action recognition, respectively. The characteristics of each dataset are given as follows.

**MPII Human Pose Dataset.** The MPII dataset for single person pose estimation is composed of about 25K images of which 15K are training samples, 3K are validation samples and 7K are testing samples (which labels are withheld by the authors). The images are taken from YouTube videos covering 410 different human activities and the poses are manually annotated with up to 16 body joints.

**Human3.6M.** The Human3.6M [21] dataset is composed by videos with 11 subjects performing 17 different activities and 4 cameras with different points of view, resulting in more than 3M frames. For each person, the dataset provides 32 body joints, from which only 17 are used to compute scores.

**Penn Action .** The Penn Action dataset [59] is composed by 2,326 videos in the wild with 15 different actions, among those “baseball pitch”, “bench press”, “strum guitar”, etc. The challenge on this dataset is that several body parts are missing in many actions and the image scales are very disparate from one sample to another.

**NTU RGB+D.** The NTU dataset is so far the biggest and a very challenging datasets for 3D action recognition. It is composed of more than 56K videos in Full HD of 60

actions performed by 40 different actors and recorded by 3 cameras in 17 different positioning setups, which results in more than 4M video frames.

### 5.2. Implementation details

For the pose estimation task, we train the network using the elastic net loss function on predicted poses as defined in the equation below:

$$L_p = \frac{1}{N_J} \sum_{n=1}^{N_J} (\|\hat{\mathbf{p}}_n - \mathbf{p}_n\|_1 + \|\hat{\mathbf{p}}_n - \mathbf{p}_n\|_2^2), \quad (3)$$

where  $\hat{\mathbf{p}}_n$  and  $\mathbf{p}_n$  are respectively the estimated and the ground truth positions of the  $n^{th}$  joint. For training, we crop bounding boxes centered on the target person by using the ground truth annotations or the persons location, when applicable. For the pose estimation task, on both MPII single person and Human3.6M datasets it is allowed to use the given persons location on evaluation. If a given body joint falls outside the cropped bounding box on training, we set the ground truth visibility flag to zero, otherwise we set it to one. The ground truth visibility information is used to supervise the predicted joint visibility vector  $\mathbf{v}$  with the binary cross entropy loss. When evaluating the pose estimation task we show the results for *single-crop* and *multi-crop*. In the first case, one centered image is used for prediction, and on the second case, multiple images are cropped with small displacements and horizontal flips and the final pose is the average prediction.

For the action recognition task, we train the network using the categorical cross entropy loss. On training, we randomly select fixed-size clips with  $T$  frames from a video sample. On test, we report results on *single-clip* or *multi-clip*. In the first case, we crop a single clip in the middle of the video. For the second case, we crop multiple clips temporally spaced of  $T/2$  frames from each other. The final scores on multi-clip is computed by the average result on all clips from one video. To estimate the bounding box on test, we do an initial pose prediction using the full images from the first, middle, and last frames of a clip. Finally, we select the maximum bounding box that encloses

Table 1. Comparison with previous work on Human3.6M evaluated on the averaged joint error (in millimeters) on reconstructed poses.

Methods	Direction	Discuss	Eat	Greet	Phone	Posing	Purchase	Sitting
Pavlakos <i>et al.</i> [35]	67.4	71.9	66.7	69.1	71.9	65.0	68.3	83.7
Mehta <i>et al.</i> [31]	52.5	63.8	55.4	62.3	71.8	52.6	72.2	86.2
Martinez <i>et al.</i> [30]	51.8	56.2	58.1	59.0	69.5	55.2	58.1	74.0
Sun <i>et al.</i> [47]	52.8	54.8	54.2	54.3	61.8	53.1	53.6	71.7
<b>Ours (single-crop)</b>	<b>51.5</b>	<b>53.4</b>	<b>49.0</b>	<b>52.5</b>	<b>53.9</b>	<b>50.3</b>	54.4	<b>63.6</b>
<b>Ours (multi-crop + h.flip)</b>	<b>49.2</b>	<b>51.6</b>	<b>47.6</b>	<b>50.5</b>	<b>51.8</b>	<b>48.5</b>	<b>51.7</b>	<b>61.5</b>
Methods	Sit Down	Smoke	Photo	Wait	Walk	Walk Dog	Walk Pair	Average
Pavlakos <i>et al.</i> [35]	96.5	71.4	76.9	65.8	59.1	74.9	63.2	71.9
Mehta <i>et al.</i> [31]	120.0	66.0	79.8	63.9	48.9	76.8	53.7	68.6
Martinez <i>et al.</i> [30]	94.6	62.3	78.4	59.1	49.5	65.1	52.4	62.9
Sun <i>et al.</i> [47]	86.7	61.5	67.2	53.4	47.1	61.6	53.4	59.1
<b>Ours (single-crop)</b>	<b>73.5</b>	<b>55.3</b>	<b>61.9</b>	<b>50.1</b>	<b>46.0</b>	<b>60.2</b>	<b>51.0</b>	<b>55.1</b>
<b>Ours (multi-crop + h.flip)</b>	<b>70.9</b>	<b>53.7</b>	<b>60.3</b>	<b>48.9</b>	<b>44.4</b>	<b>57.9</b>	<b>48.9</b>	<b>53.2</b>

all the initially predicted poses. Detailed information about the network layers and implementation are given in the supplemental material.

### 5.3. Evaluation on pose estimation

**2D pose estimation.** We perform quantitative evaluations of the 2D pose estimation using the probability of correct keypoints measure with respect to the head size (PCKh), as shown in Table 2. We are able to recover the results of [28] which is consistent with the similarity between this method and the 2D pose estimation part of our method. From the results we can see that the regression method based on Soft-argmax achieves results very close to the state of the art, specially when considered the accumulated precision given by the area under the curve (AUC), and by far the most accurate approach among fully differentiable methods.

**3D pose estimation.** On Human3.6M, we evaluate the proposed 3D pose regression method by measuring the

mean per joint position error (MPJPE), which is the most challenging and the most common metric for this dataset. We followed the common evaluation protocol [47, 35, 31, 11] by taking five subjects for training (S1, S5, S6, S7, S8) and evaluating on two subjects (S9, S11) on one every 64 frames. For training, we use the data equally balanced as 50%/50% from MPII and Human3.6M. For the multi-crop predictions we use five cropped regions and their corresponding flipped images. Our results compared to the previous approaches are presented in Table 1 and show that our approach is able to outperform the state of the art by a fair margin. Qualitative results from our method are shown in Figure 7, for both Human3.6M and MPII datasets, which also demonstrate the capability of our method to generalize 3D pose predictions from data with only 2D annotated poses.

### 5.4. Evaluation on action recognition

**2D action recognition.** We evaluate our action recognition approach on 2D scenario on the Penn Action dataset. For training the pose estimation part, we use mixed data from MPII (75%) and Penn Action (25%), using 16 body joints. The action recognition part was trained using video clips composed of  $T = 16$  frames. We reached state of the art classification score among methods using RGB and estimated poses. We also evaluated our method without considering the influence of estimated poses by using the manually annotated body joints and are also able to improve over the state of the art. Results are shown in Table 3.

**3D action recognition.** Since skeletal data from NTU is frequently noisy, we train the pose estimation part with only 10% of data from NTU, 45% from MPII, and 45% from Human3.6M, using 20 body joints and video clips of  $T = 20$  frames. Our method improves the state of the art on NTU significantly using only RGB frames and 3D predicted poses, as reported in Table 4. If we consider only RGB frames as input, our method improves over [3] by 9.9%. To the best of our knowledge, all the previous methods use provided poses given by Kinect-v2, which are known to be

Table 2. Comparison results on MPII for single person 2D pose estimation using the PCKh measure with respect to 0.2 and 0.5 of the head size. For older results, please refer to the MPII Leader Board at <http://human-pose.mpi-inf.mpg.de>.

Methods	Year	PCKh @0.2	AUC @0.2	PCKh @0.5	AUC @0.5
Detection methods					
Recurrent VGG [6]	2016	61.6	28.2	88.1	58.8
DeeperCut [20]	2016	64.0	31.7	88.5	60.8
Pose Machines [54]	2016	64.8	33.0	88.5	61.4
Heatmap regression [7]	2016	61.8	28.5	89.7	59.6
Stacked Hourglass [33]	2016	66.5	33.4	90.9	62.9
Fractal NN [34]	2017	—	—	91.2	63.6
Multi-Context Att. [16]	2017	67.8	<b>34.1</b>	91.5	63.8
Self Adversarial [15]	2017	<b>68.0</b>	34.0	91.8	63.9
Adversarial PoseNet[12]	2017	—	—	91.9	61.6
Pyramid Res. Module[56]	2017	—	—	<b>92.0</b>	<b>64.2</b>
Regression methods					
LCR-Net [42]	2017	—	—	74.2	—
Iter. Error Feedback [9]	2016	46.8	20.6	81.3	49.1
Compositional Reg.[47]	2017	—	—	86.4	—
<b>2D Soft-argmax</b>		<b>67.7</b>	<b>34.9</b>	<b>91.2</b>	<b>63.9</b>



Table 3. Comparison results on Penn Action for 2D action recognition. Results given as the percentage of correctly classified actions.

Methods	Annot. poses	RGB	Optical Flow	Estimated poses	Acc.
Nie <i>et al.</i> [55]	-	X	-	X	85.5
Iqbal <i>et al.</i> [22]	-	-	-	X	79.0
	-	X	X	X	92.9
Cao <i>et al.</i> [8]	X	X	-	-	98.1
	-	X	-	X	95.3
<b>Ours</b>	X	X	-	-	<b>98.6</b>
	-	X	-	X*	<b>97.4</b>

\* Using mixed data from PennAction and MPII.

Table 4. Comparison results on the NTU for 3D action recognition. Results given as the percentage of correctly classified actions

Methods	Kinect poses	RGB	Estimated poses	Acc. cross subject
Shahroudy <i>et al.</i> [44]	X	-	-	62.9
Liu <i>et al.</i> [26]	X	-	-	69.2
Song <i>et al.</i> [46]	X	-	-	73.4
Liu <i>et al.</i> [27]	X	-	-	74.4
Shahroudy <i>et al.</i> [45]	X	X	-	74.9
	X	-	-	77.1
Baradel <i>et al.</i> [3]	*	X	-	75.6
	X	X	-	84.8
<b>Ours</b>	-	X	-	<b>84.6</b>
	-	X	X	<b>85.5</b>

\* GT poses were used on test to select visual features.

very noisy in some cases. Although we do not use LSTM like other methods, the temporal information is well taken into account using convolution. Our results suggest this approach is sufficient for small video clips as found in NTU.

**Ablation study.** We performed varied experiments on NTU to show the contributions of each component of our methods. As can be seen on Table 5, our estimated poses increase the accuracy by 2.9% over Kinect poses. Moreover, the full optimization also improves by 3.3%, which justify the importance of a fully differentiable approach. And finally, by averaging results from multiple video clips we gain 1.1% more. We also compared the proposed approach of sequential learning followed by fine tuning (Table 3) with joint learning pose and action on PennAction, what result in 97.3%, only 0.1% lower than in the previews case.

The effectiveness of our method relies on three main characteristics: First, the multiple prediction blocks provide a continuous improvement on action accuracy, as can be seen on Figure 8. Second, thanks to our fully differentiable architecture, we can fine tune the model from RGB frames to predicted actions, which brings a significant gain in accuracy. And third, as shown on Figure 9, the proposed approach also benefits from complementary appearance and pose information which lead to better classification accuracy once aggregated.

Table 5. Results of our method on NTU considering different approaches. FT: Fine tuning, MC: Multi-clip.

Experiments	Pose	Appearance (RGB)	Aggregation
Kinect poses	63.3	76.4	78.2
Estimated poses	64.5	80.1	81.1
Est. poses + FT	71.7	83.2	84.4
Est. poses + FT + MC	74.3	84.6	85.5

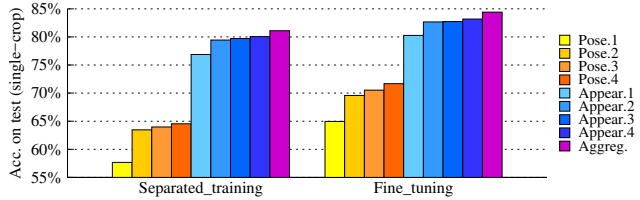


Figure 8. Action recognition accuracy on NTU from pose and appearance models in four prediction blocks, and with aggregated features, for both separated training and full network optimization (fine tuning).

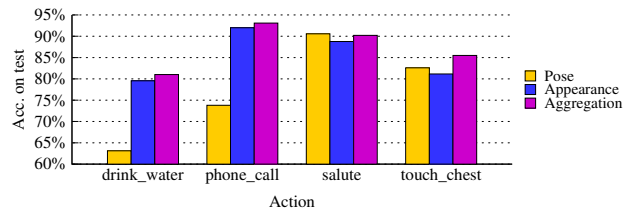


Figure 9. Action recognition accuracy on NTU for different action types from pose, and appearance models and with aggregated results.

## 6. Conclusions

In this paper, we presented a multitask deep architecture to perform 2D and 3D pose estimation jointly with action recognition. Our model first predicts the 2D and 3D location of body joints from the raw RGB frames. These locations are then used to predict the action performed in the video in two different ways: using semantic information by leveraging the temporal evolution of body joint coordinates and using visual information by performing an attention based pooling on human body parts. Heavy sharing of weights and features in our model allows us to solve four different tasks - 2D pose estimation, 3D pose estimation, 2D action recognition, 3D action recognition - with a single model very efficiently compared to dedicated approaches. We performed extensive experiments that show our approach is able to equal or even outperform dedicated approaches on all these tasks.

## 7. Acknowledgements

This work was partially founded by CNPq (Brazil) - Grant 233342/2014-1.

## References

- [1] 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)*, June 2014. [6](#)
- [2] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1014–1021, June 2009. [2](#)
- [3] F. Baradel, C. Wolf, and J. Mille. Pose-conditioned spatio-temporal attention for human action recognition. *arxiv*, 1703.10106, 2017. [1](#), [3](#), [7](#), [8](#)
- [4] 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. [3](#)
- [5] V. Belagiannis, C. Rupprecht, G. Carneiro, and N. Navab. Robust optimization for deep regression. In *International Conference on Computer Vision (ICCV)*, pages 2830–2838, Dec 2015. [2](#)
- [6] V. Belagiannis and A. Zisserman. Recurrent human pose estimation. *CoRR*, abs/1605.02914, 2016. [7](#)
- [7] A. Bulat and G. Tzimiropoulos. Human pose estimation via Convolutional Part Heatmap Regression. In *European Conference on Computer Vision (ECCV)*, pages 717–732, 2016. [2](#), [7](#)
- [8] C. Cao, Y. Zhang, C. Zhang, and H. Lu. Body joint guided 3d deep convolutional descriptors for action recognition. *CoRR*, abs/1704.07160, 2017. [3](#), [8](#)
- [9] 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)*, pages 4733–4742, June 2016. [2](#), [3](#), [7](#)
- [10] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, July 2017. [3](#)
- [11] 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. [2](#), [7](#)
- [12] Y. Chen, C. Shen, X. Wei, L. Liu, and J. Yang. Adversarial posenet: A structure-aware convolutional network for human pose estimation. *CoRR*, abs/1705.00389, 2017. [7](#)
- [13] 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. [2](#)
- [14] G. Ch'eron, I. Laptev, and C. Schmid. P-CNN: Pose-based CNN Features for Action Recognition. In *ICCV*, 2015. [1](#), [3](#)
- [15] C. Chou, J. Chien, and H. Chen. Self adversarial training for human pose estimation. *CoRR*, abs/1707.02439, 2017. [2](#), [7](#)
- [16] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang. Multi-context attention for human pose estimation. *CVPR*, 2017. [2](#), [7](#)
- [17] 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)*, pages 3041–3048, June 2013. [2](#)
- [18] G. Gkioxari, A. Toshev, and N. Jaitly. Chained Predictions Using Convolutional Neural Networks. *European Conference on Computer Vision (ECCV)*, 2016. [2](#)
- [19] S. Herath, M. Harandi, and F. Porikli. Going deeper into action recognition: A survey. *Image and Vision Computing*, 60(Supplement C):4 – 21, 2017. Regularization Techniques for High-Dimensional Data Analysis. [2](#)
- [20] 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. [2](#), [7](#)
- [21] 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*, 36(7):1325–1339, jul 2014. [2](#), [6](#)
- [22] U. Iqbal, M. Garbade, and J. Gall. Pose for action - action for pose. *FG-2017*, 2017. [1](#), [8](#)
- [23] 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. [2](#)
- [24] 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. [1](#)
- [25] I. Lifshitz, E. Fetaya, and S. Ullman. *Human Pose Estimation Using Deep Consensus Voting*, pages 246–260. Springer International Publishing, Cham, 2016. [2](#)
- [26] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *ECCV*, pages 816–833, Cham, 2016. [3](#), [8](#)
- [27] 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)*, July 2017. [3](#), [8](#)
- [28] D. C. Luvizon, H. Tabia, and D. Picard. Human pose regression by combining indirect part detection and contextual information. *CoRR*, abs/1710.02322, 2017. [1](#), [2](#), [3](#), [7](#)
- [29] D. C. Luvizon, H. Tabia, and D. Picard. Learning features combination for human action recognition from skeleton sequences. *Pattern Recognition Letters*, 2017. [3](#)
- [30] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, 2017. [2](#), [7](#)
- [31] 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*, abs/1611.09813, 2016. [2](#), [7](#)
- [32] 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*, volume 36, 2017. [2](#)
- [33] A. Newell, K. Yang, and J. Deng. Stacked Hourglass Networks for Human Pose Estimation. *European Conference on Computer Vision (ECCV)*, pages 483–499, 2016. [1](#), [2](#), [7](#)

- [34] G. Ning, Z. Zhang, and Z. He. Knowledge-guided deep fractal neural networks for human pose estimation. *IEEE Transactions on Multimedia*, PP(99):1–1, 2017. 2, 7
- [35] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Coarse-to-fine volumetric prediction for single-image 3D human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2, 7
- [36] 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. 2
- [37] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet Conditioned Pictorial Structures. In *Computer Vision and Pattern Recognition (CVPR)*, pages 588–595, June 2013. 2
- [38] 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. 2
- [39] 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. 2
- [40] L. L. Presti and M. L. Cascia. 3d skeleton-based human action classification: A survey. *Pattern Recognition*, 53:130–147, 2016. 3
- [41] U. Rafi, I. Kostrikov, J. Gall, and B. Leibe. An efficient convolutional network for human pose estimation. In *BMVC*, volume 1, page 2, 2016. 2
- [42] G. Rogez, P. Weinzaepfel, and C. Schmid. LCR-Net: Localization-Classification-Regression for Human Pose. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2017. 7
- [43] 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*, 152(Supplement C):1 – 20, 2016. 2
- [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*, June 2016. 6, 8
- [45] A. Shahroudy, T.-T. Ng, Y. Gong, and G. Wang. Deep multimodal feature analysis for action recognition in rgb+d videos. *TPAMI*, 2017. 3, 8
- [46] 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*, February 2017. 3, 8
- [47] X. Sun, J. Shang, S. Liang, and Y. Wei. Compositional human pose regression. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2, 3, 7
- [48] C. Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016. 3
- [49] B. Tekin, P. Márquez-Neila, M. Salzmann, and P. Fua. Fusing 2d uncertainty and 3d cues for monocular body pose estimation. *CoRR*, abs/1611.05708, 2016. 2
- [50] D. Tome, C. Russell, and L. Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *CVPR*, July 2017. 2
- [51] 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)*, pages 648–656, June 2015. 2
- [52] A. Toshev and C. Szegedy. DeepPose: Human Pose Estimation via Deep Neural Networks. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1653–1660, 2014. 2
- [53] G. Varol, I. Laptev, and C. Schmid. Long-term Temporal Convolutions for Action Recognition. *TPAMI*, 2017. 3
- [54] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 7
- [55] 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. 2, 8
- [56] 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)*, Oct 2017. 2, 7
- [57] A. Yao, J. Gall, and L. Van Gool. Coupled action recognition and pose estimation from multiple views. *International Journal of Computer Vision*, 100(1):16–37, Oct 2012. 1
- [58] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. LIFT: Learned Invariant Feature Transform. *European Conference on Computer Vision (ECCV)*, 2016. 1
- [59] W. Zhang, M. Zhu, and K. G. Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *ICCV*, pages 2248–2255, Dec 2013. 6
- [60] 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*, abs/1701.02354, 2017. 2

## Appendix A: Network architecture

In our implementation of the proposed approach, we divided the network architecture into four parts: the *multitask stem*, the *pose estimation model*, the *pose recognition model*, and the *appearance recognition model*. We use depth-wise separable convolutions as depicted in [Figure 10](#), batch normalization and ReLU activation. The architecture of the multitask stem is detailed in [Figure 11](#). Each pose estimation prediction block is implemented as a multi-resolution CNN, as presented in [Figure 12](#). We use  $N_d = 16$  heat maps for depth predictions. The CNN architecture for action recognition is detailed in [Figure 13](#).

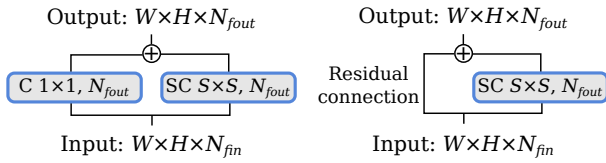


Figure 10. Separable residual module (SR) based on depth-wise separable convolutions (SC) for  $N_{fin} \neq N_{fout}$  (left), and  $N_{fin} = N_{fout}$  (right), where  $N_{fin}$  and  $N_{fout}$  are the input and output features size,  $W \times H$  is the feature map resolution, and  $S \times S$  is the size of the filters, usually  $3 \times 3$  or  $5 \times 5$ . C: Simple 2D convolution.

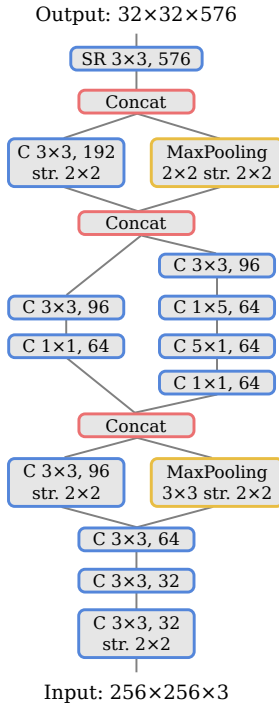


Figure 11. Shared network (entry flow) based on Inception-V4. C: Convolution, SR: Separable residual module.

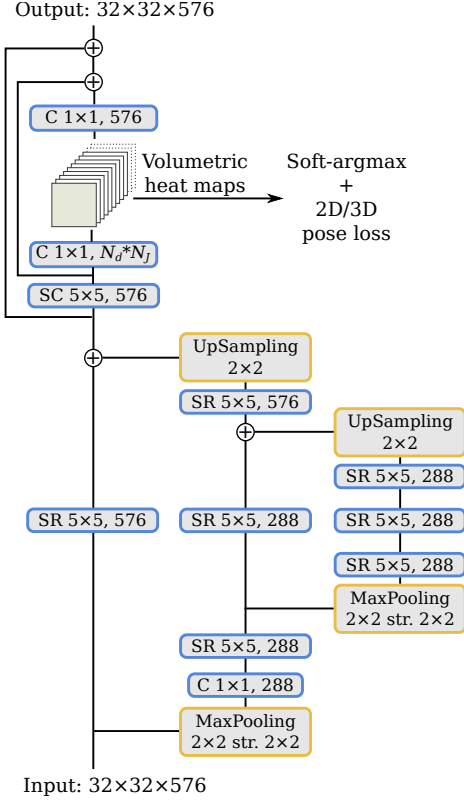


Figure 12. Prediction block for pose estimation, where  $N_d$  is the number of depth heat maps per joint and  $N_J$  is the number of body joints. C: Convolution, SR: Separable residual module.

## Appendix B: Training parameters

In order to merge different datasets, we convert the poses to a common layout, with a fixed number of joints equal to the dataset with more joints. For example, when merging the datasets Human3.6M and MPII, we use all the 17 joints in the first dataset and include one joint on MPII. All the included joints have an invalid value that is not taken into account in the loss function. Additionally, we use an alternated human pose layout, similar to the layout from the Penn Action dataset, which experimentally lead to better scores on action recognition.

We optimize the pose regression part using the RMSprop optimizer with initial learning rate of 0.001, which is reduced by a factor of 0.2 when validation score plateaus, and batches of 24 images. For the action recognition task, we train both pose and appearance models simultaneously using a pre-trained pose estimation model with weights initially frozen. In that case, we use a classical SGD optimizer with Nesterov momentum of 0.98 and initial learning rate of 0.0002, reduced by a factor of 0.2 when validation plateaus, and batches of 2 video clips. When validation accuracy stagnates, we divide the final learning rate by 10 and fine tune the full network for more 5 epochs. When reporting

Table 6. Our results on averaged joint error on reconstructed poses for 3D pose estimation on Human3.6 considering single dataset training (Human3.6M only) and mixed data (Human3.6M + MPII). SC: Single-crop, MC: Multi-crop.

Methods	Direction	Discuss	Eat	Greet	Phone	Posing	Purchase	Sitting
<b>Human3.6 only - SC</b>	64.1	66.3	59.4	61.9	64.4	59.6	66.1	78.4
<b>Human3.6 only - MC</b>	61.7	63.5	56.1	60.1	60.0	57.6	64.6	75.1
<b>Human3.6 + MPII - SC</b>	51.5	53.4	49.0	52.5	53.9	50.3	54.4	63.6
<b>Human3.6 + MPII - MC</b>	49.2	51.6	47.6	50.5	51.8	48.5	51.7	61.5
Methods	Sit Down	Smoke	Photo	Wait	Walk	Walk Dog	Walk Pair	Average
<b>Human3.6 only - SC</b>	102.1	67.4	77.8	59.3	51.5	69.7	60.1	67.3
<b>Human3.6 only - MC</b>	95.4	63.4	73.3	57.0	48.2	66.8	55.1	63.8
<b>Human3.6 + MPII - SC</b>	73.5	55.3	61.9	50.1	46.0	60.2	51.0	55.1
<b>Human3.6 + MPII - MC</b>	70.9	53.7	60.3	48.9	44.4	57.9	48.9	53.2

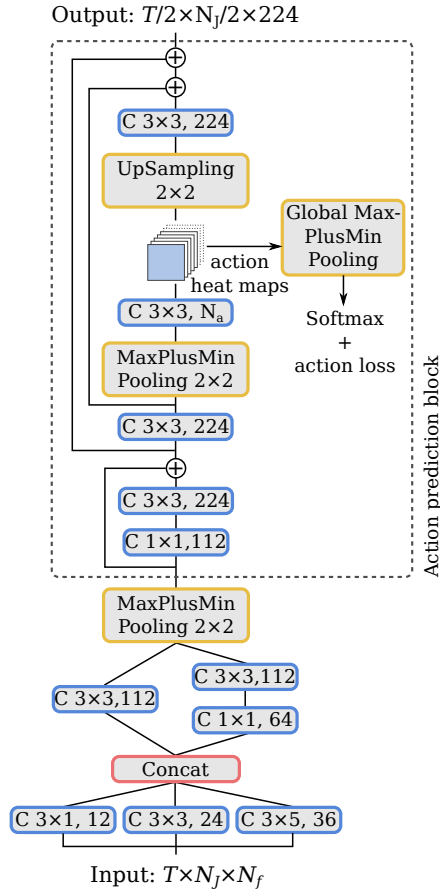


Figure 13. Network architecture for action recognition. The action prediction blocks can be repeated  $K$  times. The same architecture is used for pose and appearance recognition, except that for pose, each convolution uses half the number of features showed here.  $T$  corresponds the number of frames and  $N_a$  is the number of actions.

only pose estimation scores, we use eight prediction blocks ( $K = 8$ ), and for action recognition, we use four prediction blocks ( $K = 4$ ). For all experiments, we use cropped RGB images of size  $256 \times 256$ . We augment the training data by performing random rotations from  $-45^\circ$  to  $+45^\circ$ ,

scaling from 0.7 to 1.3, vertical and horizontal translations respectively from  $-40$  to  $+40$  pixels, video subsampling by a factor from 1 to 3, and random horizontal flipping.

## Appendix C: Additional experiments

In order to show the contribution of multiple datasets in training, we show in Table 6 additional results on 3D pose estimation using Human3.6M only and Human3.6M + MPII datasets for training.